ABSTRACT

Title of Dissertation: NEW EFFICIENT ALGORITHMS FOR NESTED MACHINE LEARNING PROBLEMS

Junyi Li
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Heng Huang
Department of Computer Science

In recent years, machine learning (ML) has achieved remarkable success by training large-scale models on vast datasets. However, building these models involves multiple interdependent tasks-such as data selection, hyperparameter tuning, and model architecture search-that can lead to nested objectives when optimized jointly. These nested objectives arise because each task both influences and depends on the others. This dissertation aims to develop efficient algorithms to tackle these challenging nested problems in machine learning.

In the first part, we formalize nested ML problems as bilevel optimization tasks and presenting efficient algorithms with theoretical guarantees that solve them. Then, in the second part, we extend to the federated/distributed learning context, examining how algorithmic designs must be adapted to meet the challenges of that environment. Finally, in the third part, we cover challenges with hierarchies in the distributed learning setting including data cleaning, network pruning and constrained problems.

NEW EFFICIENT ALGORITHMS FOR NESTED MACHINE LEARNING PROBLEMS

by

Junyi Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Heng Huang, Chair/Advisor
Professor Zhicheng Liu
Professor Ruohan Gao
Professor Ang Li
Professor Shura S. Bhattacharyya, Dean's Representative

To my parents.

Acknowledgments

First and foremost, I would like to extend my deepest gratitude to my advisor, Dr. Heng Huang, for his unwavering support, guidance, and mentorship throughout my PhD journey. His patience, encouragement, and keen insights not only honed my critical thinking skills but also instilled in me a passion for inquiry that will guide my future endeavors. His commitment to helping me grow both personally and professionally has been instrumental to my development, and for that I am truly thankful.

I would also like to thank my committee members, Dr. Zhicheng Liu, Dr. Ruohan Gao, Dr. Ang Li, and Dr. Shura S. Bhattacharyya, for their gracious commitment to serve on my PhD dissertation committee. Moreover, I would also like to thank Dr. Tianyi Zhou for serving on my preliminary exam committee. I am truly grateful for the time and effort they have dedicated to my academic growth.

Thanks to all my collaborators and mentors, including Dr. Zhan Liang, Dr. Feihu Huang, Dr. Hongchang Gao, Dr. Yanfu Zhang and Dr. Shangqian Gao, Dr. Ziyi Chen, Dr. Peiran Yu, Dr. Zhengmian Hu and Dr. Xidong Wu at University of Pittsburgh and University of Maryland; Dr. Charith Peris, Dr. Ninareh Mehrabi, Dr. Palash Goyal, Dr. Kai-Wei Chang, Dr. Aram Galstyan, Dr.Richard Zemel and Dr. Rahul Gupta at Amazon.com. I feel so lucky to work with so many excellent colleagues. To all those mentioned and many others who have supported me along the way, thank you for being part of this incredible journey.

Last but certainly not least, I would like to also extend my deepest gratitude to my parents, Mr. Rongqing Li and Ms. Yonghong Wang, for their unwavering love, guidance, and encouragement. Their commitment to my education and their constant, unconditional support have shaped who I am today. I am forever thankful for the faith they have placed in me and for the countless ways they have helped me grow and succeed.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BN	Batch Normalization
BO/BiO	Bilevel Optimization
CNN	Convolution Neural Networks
DNN	Deep Neural Networks
LLM	Large Language Model
FL	Federated Learning
FLOPs	Floating Point Operations Per Second
FedAvg	Federated Averaging
GRU	Gated Recurrent Unit
HN	Hyper Network
MSE	Mean Squared Error
NLP	Natural Language Processing
NN	Neural Networks
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SVRG	Stochastic Variance Reduced Gradient
STORM	STOchastic Recursive Momentum

Chapter 1: Introduction

Modern machine learning rests on two fundamental pillars: models and data. Over the past few years, remarkable success has been achieved by training large-scale models on vast datasets. However, machine learning often requires navigating multiple data sources and handling various model constraints, which can interleave and lead to nested problems. This dissertation presents efficient algorithms for addressing these nested challenges in both centralized and federated learning environments.

In chapter Chapter 2, we propose a novel Hessian inverse free Fully Single Loop Algorithm (FSLA) for bilevel optimization problems. Classic algorithms for bilevel optimization either admit a double loop structure which is computationally expensive. Recently, several single loop algorithms have been proposed with optimizing the inner and outer variable alternatively. However, these algorithms not yet achieve fully single loop. As they overlook the loop needed to evaluate the hyper-gradient for a given inner and outer state. In order to develop a fully single loop algorithm, we first study the structure of the hyper-gradient and identify a general approximation formulation of hyper-gradient computation that encompasses several previous common approaches, *e.g.* back-propagation through time, conjugate gradient, *etc.* Based on this formulation, we introduce a new state variable to maintain the historical hyper-gradient information. Combining our new formulation with the alternative

update of the inner and outer variables, we propose an efficient fully single loop algorithm. We theoretically show that the error generated by the new state can be bounded and our algorithm converges with the rate of $O(\epsilon^{-2})$.

In chapter Chapter 3, we address a common limitation in bilevel algorithms: presumption of independent sampling. The independent sampling assumption can lead to increased computational costs due to the complicated hyper-gradient formulation of bilevel problems. To address this challenge, we study the example-selection strategy for bilevel optimization and introduce a without-replacement sampling based algorithm which achieves a faster convergence rate compared to its counterparts that rely on independent sampling. Beyond the standard bilevel optimization formulation, we extend our discussion to conditional bilevel optimization and also two special cases: minimax and compositional optimization.

In chapter Chapter 4, we extend the discussion of bilevel optimization to the Federated/Distributed Learning setting. We investigate Federated Bilevel Optimization problems and propose a communication-efficient algorithm, named FedBiOAcc. The algorithm leverages an efficient estimation of the hyper-gradient in the distributed setting and utilizes the momentum-based variance-reduction acceleration. Remarkably, FedBiOAcc achieves a communication complexity $O(\epsilon^{-1})$, a sample complexity $O(\epsilon^{-1.5})$ and the linear speed up with respect to the number of clients. We also analyze a special case of the Federated Bilevel Optimization problems, where lower level problems are locally managed by clients. We prove that FedBiOAcc-Local, a modified version of FedBiOAcc, converges at the same rate for this type of problems.

In Federated Learning (FL), learning and communication have fundamentally divergent requirements for parameter selection, akin to two opposite teams in a tug-of-war game.

To mitigate this discrepancy, we introduce FedSep in chapter Chapter 5, a novel two-layer federated learning framework. FedSep consists of separated communication and learning layers for each client and the two layers are connected through decode/encode operations. In particular, the decoding operation is formulated as a minimization problem. We view FedSep as a federated bilevel optimization problem and propose an efficient algorithm to solve it. Theoretically, we demonstrate that its convergence matches that of the standard FL algorithms. The separation of communication and learning in FedSep offers innovative solutions to various challenging problems in FL, such as Communication-Efficient FL and Heterogeneous-Model FL.

In FL, the data is kept locally by each user. This protects the user privacy, but also makes the server difficult to verify data quality, especially if the data are correctly labeled. Training with corrupted labels is harmful to the federated learning task; however, little attention has been paid to FL in the case of label noise. In chapter Chapter 6, we focus on this problem and propose a learning-based reweighting approach to mitigate the effect of noisy labels in FL. More precisely, we tuned a weight for each training sample such that the learned model has optimal generalization performance over a validation set. More formally, the process can be formulated as a Federated Bilevel Optimization problem. We identify that the high communication cost in hypergradient evaluation is the major bottleneck. So we propose *Comm-FedBiO* to solve the general Federated Bilevel Optimization problems; more specifically, we propose two communication-efficient subroutines to estimate the hypergradient. Convergence analysis of the proposed algorithms is also provided. Finally, we apply the proposed algorithms to solve the noisy label problem.

Neural network pruning, particularly channel pruning, is a widely used technique for

compressing deep learning models to enable their deployment on edge devices with limited resources. Typically, redundant weights or structures are removed to achieve the target resource budget. Although data-driven pruning approaches have proven to be more effective, they cannot be directly applied to federated learning (FL) because of distributed and confidential datasets. In response to this challenge, we design a new network pruning method for FL in chapter Chapter 7. We propose device-wise sub-networks for each device, assuming that the data distribution is similar within each device. These sub-networks are generated through sub-network embeddings and a hypernetwork. To further minimize memory usage and communication costs, we permanently prune the full model to remove weights that are not useful for all devices. During the FL process, we simultaneously train the device-wise sub-networks and the base sub-network to facilitate the pruning process. We then finetune the pruned model with device-wise sub-networks to regain performance. Moreover, we provided the theoretical guarantee of convergence for our method. Our method achieves better performance and resource trade-off than other well-established network pruning baselines, as demonstrated through extensive experiments.

Many crucial FL applications, like disease diagnosis and biomarker identification, often rely on constrained formulations such as Lasso and group Lasso. In chapter Chapter 8, we apply adaptive gradient methods to FL problems with constrains. More specifically, we introduce **FedDA**, a novel adaptive gradient framework for FL. This framework utilizes a restarted dual averaging technique and is compatible with a range of gradient estimation methods and adaptive learning rate schedules. Specifically, an instantiation of our framework FedDA-MVR achieves sample complexity $\tilde{O}(K^{-1}\epsilon^{-1.5})$ and communication complexity $\tilde{O}(K^{-0.25}\epsilon^{-1.25})$ for finding a stationary point ϵ in the constrained setting with K be the

number of clients. We conduct experiments over both constrained and unconstrained tasks to confirm the effectiveness of our approach.

Part I

Solving Nested Problems via Bilevel Optimization

Chapter 2: A Fully Single Loop Algorithm for Bilevel Optimization without Hessian Inverse

2.1 Introduction

In this paper, we study the bilevel optimization problem, which includes two levels of optimization: an outer problem and an inner problem. The outer problem depends on the solution of the inner problem. Many machine learning tasks can be formulated as a bilevel optimization problem, such as hyper-parameter optimization [4], meta learning [5], Stackelberg game model [6], equilibrium model [7], *etc.* However, Bilevel optimization is challenging to solve. The gradient-based algorithm [6] requires a double loop structure. For each inner loop, the inner problem is solved with the given outer state. In the outer loop, the hyper-gradient (the gradient $w.r.t$ the outer variable) is evaluated based on the solution of the inner loop and the outer state is updated with a gradient-based optimizer (such as SGD). This simple double loop algorithm is guaranteed to converge under mild assumptions and works well for small scale problems. But when the inner problem is in large scale, the double loop algorithm is very slow and becomes impractical.

In fact, hyper-gradient evaluation is the major bottleneck of bilevel optimization. The solution of the inner problem is usually implicitly defined over the outer state, and naturally

its gradient $w.r.t$ the outer variable is also implicitly defined. To evaluate hyper-gradient, we need to approximate this implicit gradient via an iterative algorithm (the inner loop). Various algorithms have been proposed for hyper-gradient evaluation [4, 6, 7, 8, 9]. These methods were designed from various perspectives and based on different techniques, thus they look quite different on the first sight. However, we can use a general formulation to incorporate all these methods under one framework. Roughly, the hyper-gradient evaluation can be expressed as a finite sum of terms defined over a sequence of inner states and momentum coefficients, and these states and coefficients are chosen differently in distinct algorithms. Our general formulation provides a new perspective to help understand the properties of these classic methods. In particular, we derive a sufficient condition such that the general formulation converges to the exact hyper-gradient.

One benefit of our general formulation is to inspire new algorithms for bilevel optimization. Based on our general formulation, we propose a new fully single loop algorithm named as 'FSLA'. Compared to previous single loop algorithms [10, 11, 12, 13, 14], our FSLA does not require any inner loop. In literature [12, 13], the existing methods either reuse the last hyper-iteration's inner solution as a warm start of the current iteration, or just alternatively update the inner and outer variables with carefully designed learning rate schedule [15]. They also utilize the variance-reduction techniques to control the variance [13]. However, these methods focus on solving the inner problem and pay little attention to the hyper-gradient evaluation process which also requires a loop. In our new algorithm, we introduce a new state v_k to keep track of the historical hyper-gradient information. During each iteration, we perform one step update. We study the bias caused by v_k and theoretically show that the convergence of our new algorithm is with rate $O(\epsilon^{-2})$.

The main contributions of this paper can be summarized as follows:

1. We propose a general formulation to unify different existing hyper-gradient approximation methods under the same framework, and identify the sufficient condition on which it converges to the exact hyper-gradient;

2. We propose a new fully single loop algorithm for bilevel optimization. The new algorithm avoids the need of time-consuming Hessian inverse by introducing a new state to track the historical hyper-gradient information;

3. We prove that our new algorithm has fast $\mathrm{O}(\epsilon^{-2})$ convergence rate for the nonconvex-strongly-convex case, and we validate effectiveness of our algorithm over different bilevel optimization based machine learning tasks.

Organization: The rest of this paper is organized as follows: in Section 2, we briefly review the recent development of Bilevel optimization; in Section 3, we introduce the general formulation of hyper-gradient approximation and the sufficient condition of convergence; in Section 4, we formally propose our new fully single loop algorithm, FSLA; in Section 5, we present the convergence result of our algorithm; in Section 6, we perform experiments to validate our proposed methods; in Section 7, we conclude and summarize the paper.

Notations: We use ∇_x to denote the full gradient *w.r.t.* the variable x, where the subscript is omitted if clear from the context, and ∂_x denotes the partial derivative. Higher order derivatives follow similar rules. $||\cdot||$ represents ℓ_2-norm for vectors and spectral norm for matrices. $[K]$ represents the sequence from 0 to K. $\Pi_{i=m}^{n} A_i = A_m \times \ldots A_n$ if $m \leq n$, and $\Pi_{i=m}^{n} A_i = I$ if $m > n$.

2.2 Related Works

Bilevel optimization dates back to the 1960s when Willoughby [16] proposed a regularization method, and then followed by many research works [8, 17, 18, 19]. In machine learning community, similar ideas in the name of implicit differentiation were also used in Hyper-parameter Optimization for a long time [20, 21, 22, 23]. However, implicit differentiation needs to compute an accurate inner problem solution in each update of the outer variable, which leads to high computational cost for large-scale problems. Thus, researchers turned to solve the inner problem with a fix number of steps, and computed the gradient *w.r.t* the outer variables with the 'back-propagation through time' technique [24, 25, 26, 27, 28]. Domke [24] considered the case when the inner optimizer is gradient-descent, heavy ball, and LBFGS method, and derived their reversing dynamics with the energy models as example; Maclaurin et al. [25] considered momentum-based SGD method, furthermore, Franceschi et al. [26] discussed two modes, a forward mode and a backward mode, and compared the trade-off of these two modes in terms of memory-usage and computation efficiency; Pedregosa [27] studied the influence of inner solution errors to the convergence of bilevel optimization; Shaban et al. [28] proposed to truncate the back propagation path to save computation.

The back-propagation based methods work well in practice, but the number of inner steps usually relies on trial and error, furthermore, it is still expensive to perform multiple inner steps for modern machine learning models with hundreds of millions of parameters. Recently, it witnessed a surge of interest in using implicit differentiation to derive single loop algorithms. Ghadimi and Wang [6] introduced BSA, an accelerated approximate

implicit differentiation method with Neumann Series. Hong et al. [15] proposed TTSA, a single loop algorithm with a two-timescale learning rate schedule. Ji et al. [12] and Ji and Liang [29] presented warm start strategy to reduce the number of inner steps needed at each iteration. Khanduri et al. [13] designed SUSTAIN which applied variance reduction technique [30, 31] over both the inner and the outer variable. Chen et al. [11] proposed STABLE, a single loop algorithm accumulating the Hessian matrix, and achieved the same order of sample complexity as the single-level optimization problems. Yang et al. [32] proposed two algorithms: MRBO and VRBO. The MRBO method uses double variance reduction trick and resembles SUSTAIN, while VRBO is based on SARAH/SPIDER. Huang and Huang [33] proposed BiO-BreD which is also based on the variance reduction technique with a better dependence over the condition number of inner problem. Huang and Huang [14] proposed BiAdam which accelerates Bilevel Optimization with adaptive gradients. Meanwhile, there are also works utilizing other strategies such as penalty methods [34], and also other formulations like the case where the inner problem has non-unique minimizers [35, 36].

The bilevel optimization has been widely applied to various machine learning applications. Hyper-parameter optimization [4, 5, 37] uses bilevel optimization extensively. Besides, the idea of bilevel optimization has also been applied to meta learning [38, 39, 40], neural architecture search [41, 42, 43], adversarial learning [44, 45], deep reinforcement learning [46], sparse learning [47, 48, 49] *etc.* For a more thorough review of these applications, please refer to the Table 2 of the survey paper by Liu et al. [50].

2.3 A General Formulation of Hyper-Gradient Approximation

In general, bilevel optimization has the following form:

$$\min_{\lambda \in \Lambda} \; f(\lambda) \coloneqq F(\lambda, \omega_\lambda) \quad s.t. \; \omega_\lambda = \arg\min_{\omega} G(\lambda, \omega) \tag{2.1}$$

where F, G denote the outer and inner problems, λ, ω denote the outer and inner variables. Under mild assumptions, the hyper-gradient $\nabla_\lambda f$ can be expressed in Proposition 1:

Proposition 1. *If for any $\lambda \in \Lambda$, ω_λ is unique, and $\partial_{\omega^2} G(\lambda, \omega_\lambda)$ is invertible, we have:*

$$\nabla_\lambda f = \partial_\lambda F(\lambda, \omega_\lambda) + \nabla_\lambda \omega_\lambda \times \partial_\omega F(\lambda, \omega_\lambda) \tag{2.2}$$

and $\nabla_\lambda \omega_\lambda = -\partial_{\omega\lambda} G(\lambda, \omega_\lambda) \partial_{\omega^2} G(\lambda, \omega_\lambda)^{-1}$.

The proof of this proposition is a direct application of the implicit function theorem (we include it in Appendix B.2 for completion). Eq. (2.2) is hard to evaluate due to the involved matrix inversion, and we instead evaluate it approximately. Various approximate methods are proposed in the literature. The most well-known ones are: Back Propagation through Time (BP), Neumann series (NS) and conjugate gradient descent (CG). They look quite different on the first sight: BP is derived based on the chain rule, NS and CG are based on different ways of approximating $\partial_{\omega^2} G(\lambda, \omega_\lambda)^{-1}$. However, they share a common structure, as stated in the following lemma. We use $\nabla_\lambda f$ to represent $\nabla_\lambda f(\lambda)$ when the outer state λ is clear from the context. In the remainder of this section: we use $[K]$ to denote the sequence, k refers to k_{th} element of $[K]$, while s refers to s_{th} element of sub-sequence $[k]$.

Lemma 2. *Given a positive integer K, a sequence of inner variable states $\{\omega_k\}$, vectors $\{p_k\}$ and a sequence of coefficients $\{\beta_k\}$ for $k \in [K]$. A general form of approximate hyper-gradient evaluated at state λ is:*

$$\nabla_\lambda f_K = \partial_\lambda F(\lambda, \omega_K) - \sum_{k=0}^{K-1} \beta_k s_k$$

where s_k has two modes:

$$s_k = \begin{cases} \partial_{\omega\lambda} G(\lambda, \omega_k) \prod_{s=k+1}^{K-1} (I - \beta_s \partial_{\omega^2} G(\lambda, \omega_s)) p_K \\ \partial_{\omega\lambda} G(\lambda, \omega_K) \prod_{s=k+1}^{K-1} (I - \beta_s \partial_{\omega^2} G(\lambda, \omega_s)) p_k \end{cases}$$

We call these two modes as backward and forward respectively (in terms of p_k). More specifically, we have:

1. *BP is in backward mode with $\omega_k = \hat{\omega}_k$, $p_k = \partial_\omega F(\lambda, \hat{\omega}_k)$ and $\beta_k = \eta_k$ for $k \in [K]$. $\{\hat{\omega}_k\}$ are an inner variable sequence generated by the gradient descent algorithm and $\{\eta_k\}$ is the corresponding learning rate sequence;*

2. *NS is in backward mode with $\omega_k = \hat{\omega}$, $p_k = \partial_\omega F(\lambda, \hat{\omega})$ and $\beta_k = \beta$ for $k \in [K]$. $\hat{\omega}$ is an inner variable state and β is some constant;*

3. *CG is in the forward mode with $\omega_k = \hat{\omega}$ for $k \in [K]$. $\hat{\omega}$ is an inner variable state, $\{p_k\}$ and $\{\beta_k\}$ is chosen adaptively by the conjugate steps.*

Proof. In the proof, we justify that the three cases mentioned above indeed satisfy the proposed general hyper-gradient computation formulation.

Case (1): Without loss of generality, assume we run a gradient descent optimizer K steps

over the inner problem in the BP method. In other words, it solves:

$$\min_{\lambda\in\Lambda} \; f_K(\lambda) \coloneqq F(\lambda, \hat{\omega}_K) s.t.\ \hat{\omega}_k = \hat{\omega}_{k-1} - \eta_k \nabla G(\lambda, \hat{\omega}_{k-1}), k \in [K]$$

where η_k is the learning rate. By the chain rule, we get the hyper-gradient $\nabla_\lambda f_K$ of this problem:

$$\nabla_\lambda f_K = \partial_\lambda F(\lambda, \hat{\omega}_K) - \sum_{k=0}^{K-1} \left(\eta_k \partial_{\omega\lambda} G(\lambda, \hat{\omega}_k) \times \prod_{s=k+1}^{K-1} (I - \eta_k \partial_{\omega^2} G(\lambda, \hat{\omega}_s)) \right) \partial_\omega F(\lambda, \hat{\omega}_K)$$

It is easy to verify the claim in the lemma;

Case (2): The NS method is based on the hyper-gradient expression in Proposition 1, but it assumes access of an approximate solution $\hat{\omega}$ instead of the optimum ω_λ and then approximates $\partial_{\omega^2} G(\lambda, \hat{\omega})^{-1}$ with the first K terms of the Neumann series. More precisely:

$$\partial_{\omega^2} G(\lambda, \hat{\omega})^{-1} \approx \beta \sum_{k=0}^{K-1} \left(I - \beta \partial_{\omega^2} G(\lambda, \hat{\omega}) \right)^k \tag{2.3}$$

where β is a small constant. Then we replace ω_λ with $\hat{\omega}$ in Eq. (2.2) and combine with Eq. (2.3). We have:

$$\nabla_\lambda f_K = \partial_\lambda F(\lambda, \hat{\omega}) - \sum_{k=0}^{K-1} \left(\beta \partial_{\omega\lambda} G(\lambda, \hat{\omega}) \times \left(I - \beta \partial_{\omega^2} G(\lambda, \hat{\omega}) \right)^k \right) \partial_\omega F(\lambda, \hat{\omega})$$

Substitute k with $K - k - 1$, it is straightforward to verify the claim in the lemma;

Case (3): The CG method uses the fact that $x = \partial_{\omega^2} G(\lambda, \hat{\omega})^{-1} \partial_\omega F(\lambda, \hat{\omega})$ is the minimizer

of the quadratic optimization problem: $\arg\min_v \frac{1}{2}x^T Ax - x^T b$, where $A = \partial_{\omega^2} G(\lambda, \hat{\omega})$ and $b = \partial_\omega F(\lambda, \hat{\omega})$. Solving this quadratic problem with the conjugate gradient descent, we have the following update rule:

$$x_{k+1} = x_k - \alpha_k(p_k) = (I - \alpha_k A)x_k + \alpha_k(b + \gamma_k p_{k-1})$$

Then second equality follows the update rule of linear CG algorithm. α_k, γ_k are the learning rate and p_k is the conjugate directions. Please refer to section 5 by Nocedal and Wright for more details. Then x_K takes the following form:

$$x_K = \sum_{k=0}^{K-1} \alpha_k \prod_{s=k+1}^{K-1} (I - \alpha_s A)(b + \gamma_k p_{k-1}) \tag{2.4}$$

Substitute the values of A and b into Eq. (2.4) and then combine it with Eq. (2.2), where we approximate $\partial_{\omega\lambda} G(\lambda, \omega_\lambda)$ with $\partial_{\omega\lambda} G(\lambda, \hat{\omega})$. It is straightforward to verify the claim in the lemma. This completes the proof. □

The general formulation in the lemma provides a unified view of the BP, NS and CG methods. Firstly, we can verify that using the NS method is equivalent to solving the quadratic problem (defined in case (3)) with (constant learning rate) the gradient descent. Since the CG method usually performs better than gradient descent in solving linear systems, we expect the CG method requires smaller K than the NS method to reach a given estimation error. Then we compare NS with BP, their difference lies in the sequence $\{\omega_k\}$ and $\{\beta_k\}$: NS uses a single state $\hat{\omega}(\beta)$, while BP uses a sequence of states $\{\omega_k\}(\{\beta_k\})$.

It would be interesting to identify sufficient conditions for $\{\beta_k\}$, $\{\omega_k\}$ and $\{p_k\}$ such

that the general formulation converges to the exact hyper-gradient $\nabla_\lambda f$. In fact, we have the following lemma:

Lemma 3. *Suppose we denote* $m_k = \prod_{s=0}^{k}(1-\mu_G\beta_s)$, $e_{\omega,k} = ||\omega_k - \omega_\lambda||$, $e_{p,k} = ||p_k - \partial_\omega F(\lambda,\omega_\lambda)||$ *for* $k \in [K]$. *Then if* $\lim_{K\to\infty} m_K = 0$, $\lim_{k\to\infty} e_{\omega,K} = 0$, $\lim_{K\to\infty} m_K \sum_{k=0}^{K-1} \beta_k e_{\omega,k}/m_k$ *is finite, in addition, for the backward mode:* $\lim_{K\to\infty} e_{p,K} = 0$*; for the forward mode:*

$$\lim_{K\to\infty} m_K \sum_{k=0}^{K-1} \beta_k e_{p,k}/m_k$$

is finite. Then we have $\nabla_\lambda f_K \to \nabla_\lambda f$, *when* $K \to \infty$, *where* $\nabla_\lambda f_K$ *is defined in Lemma 2.*

We defer the proof of Lemma 3 in in Appendix C, we show some basic ideas here. Firstly, it is straightforward to show $\partial_\lambda F(\lambda,\omega_K) \to \partial_\lambda F(\lambda,\omega_\lambda)$ by using the smoothness assumption and the condition $\omega_K \to \omega_\lambda$. For the term $\sum_{k=0}^{K-1}\beta_k s_k$, we take the backward mode as an example (the forward mode follows similar idea). We denote A_K and A^* as:

$$A_K = \sum_{k=0}^{K-1} \beta_k \partial_{\omega\lambda} G(\lambda,\omega_k) \prod_{s=k+1}^{K-1} (I - \beta_s \partial_{\omega^2} G(\lambda,\omega_s)) \tag{2.5}$$

and $A^* = \partial_{\omega\lambda}G(\lambda,\omega_\lambda)\partial_{\omega^2}G(\lambda,\omega_\lambda)^{-1}$. To show $A_K \to A^*$, we use the recursive relation of A_k and A^*:

$$\begin{aligned} A_{k+1} =& A_k\left(I - \beta_k\partial_{\omega^2}G(\lambda,\omega_k)\right) + \beta_k\partial_{\omega\lambda}(\lambda,\omega_k) \\ =& (1-\beta_k)A_k + \beta_k(A_k(I - \partial_{\omega^2}G(\lambda,\omega_k)) + \partial_{\omega\lambda}G(\lambda,\omega_k)) \end{aligned} \tag{2.6}$$

and $A^* = A^*(I - \partial_{\omega^2}G(\lambda,\omega_\lambda)) + \partial_{\omega\lambda}G(\lambda,\omega_\lambda)$. Based on the recursive relation above and

some linear algebra derivation, we get:

$$||A_{k+1} - A^*|| \leq (1 - \mu_G \beta_k)||A_k - A^*|| + C_1 \beta_k e_{\omega,k} \tag{2.7}$$

It is straightforward to verify that the conditions in the lemma guarantee the convergence of Eq. (2.7). As shown in Eq. (2.7), the momentum coefficient β_k represents the trade-off between the progress and the induced error in one iteration: Larger β_k leads to better contraction factor ϵ, but also bigger bias term $e_{\omega,k}$. In fact, we can get meaningful convergence rate of $\nabla_\lambda f_K$ for several special choices of sequences. As shown in Corollary 4 and Corollary 5:

Corollary 4. *Given a positive integer K, inner state $\hat{\omega}_K$ and constant β. Then we set $\omega_k = \hat{\omega}_K$, $p_k = \partial_\omega F(\lambda, \hat{\omega}_K)$ and $\beta_k = \beta$ for $k \in [K]$. If $\exists\ \epsilon \in (0, 1)$, such that $\beta_k \in (\epsilon/\mu_G, 1/\mu_G)$, then we have: $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(e_{\omega,K})$*

Corollary 5. *Given a positive integer K, we have sequence $\{\hat{\omega}_k\}$ and $\{\hat{\beta}_k\}$ for $k \in [K]$. We pick $\omega_k = \hat{\omega}_k$, $p_k = \partial_\omega F(\lambda, \hat{\omega}_K)$ and $\beta_k = \hat{\beta}_k$ for $k \in [K]$. Suppose we have $\beta_k = O(k^{-1})$ and $e_{\omega,k} = O(k^{-0.5})$, then: $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(K^{-0.5})$*

In fact, the sequences chosen in Corollary 4 correspond to that in the NS method, and we show that if we choose β properly, the hyper-gradient estimation converges in the rate of $e_{\omega,K}$. The BP method corresponds to Corollary 5. Suppose we optimize the inner problem with stochastic gradient descent and learning rate $\beta_k = O(1/k)$, we get $e_{\omega,k} = O(1/\sqrt{k})$. This satisfies the condition in the corollary.

To the best of our knowledge, this is first complexity result of the stochastic case.

Grazzi et al. considered the linear convergence case for BP method. We introduce some more examples of the application of Lemma 3 in the Appendix C.

2.4 Fully Single Loop Algorithm (FSLA)

In this section, we introduce a new single loop algorithm for bilevel optimization. In section 2.3, we identify a general formulation of hyper-gradient approximation in Lemma 2: $\nabla_\lambda f_K = \partial_\lambda F(\lambda, \omega_K) - \sum_{k=0}^{K-1} \beta_k s_k$, where s_k has two modes: forward mode and backward mode. For both modes, they can be expressed by a recursive equation. In the backward mode, we write a recursive relation in terms of A_k as defined in Eq. (2.5) and Eq. (2.6). Similarly, we define $v_K = \sum_{k=0}^{K-1} \beta_k \prod_{s=k+1}^{K-1} (I - \beta_s \partial_{\omega^2} G(\lambda, \omega_s))\, p_k$ in the forward mode, and derive a recursive relation as:

$$v_{k+1} = (I - \beta_k \partial_{\omega^2} G(\lambda, \omega_k))\, v_k + \beta_k p_k \tag{2.8}$$

Based on this observation, we can propose a new single loop bilevel optimization algorithm without Hessian Inverse. In previous literature, researchers maintain a inner state ω_k to avoid the inner loop solving ω_λ for each new outer state λ. On top of this, we maintain a new state v_k and evaluate the hyper-gradient as follows:

$$\begin{aligned} v_k &= \beta_k \partial_\omega F(\lambda, \omega_k) + (I - \beta_k \partial_{\omega^2} G(\lambda, \omega_k)) v_{k-1} \\ \nabla_\lambda f_k &= \partial_\lambda F(\lambda, \omega_k) - \partial_{\omega\lambda} G(\lambda, \omega_k) v_k \end{aligned} \tag{2.9}$$

Note we set $p_k = \partial_\omega F(\lambda, \omega_k)$. Then we alternatively update ω_k, v_k and λ_k, and achieve a fully single loop algorithm without Hessian-Inverse. Note that it is also possible to keep track of A_k, but then we need to store a matrix, which cost more storage. More formally, we get the following new alternative update rule:

$$\begin{aligned} \lambda_k &= \lambda_{k-1} - \alpha_{k-1}\nabla_\lambda f_{k-1} \\ \omega_k &= \omega_{k-1} - \tau_k \partial_\omega G(\lambda_k, \omega_{k-1}) \\ v_k &= \beta_k \partial_\omega F(\lambda_k, \omega_k) + (I - \beta_k \partial_{\omega^2} G(\lambda_k, \omega_k))v_{k-1} \\ \nabla_\lambda f_k &= \partial_\lambda F(\lambda_k, \omega_k) - \partial_{\omega\lambda} G(\lambda_k, \omega_k) v_k \end{aligned} \tag{2.10}$$

where τ_k and α_k are learning rates for inner and outer updates. As a comparison, existing single loop algorithms recompute $\nabla_\lambda f$ from scratch for each new λ, while our algorithm performs one step update over v_k. What's more, we can express $\nabla_\lambda f_k$ in Eq. (2.10) as follows:

$$\nabla_\lambda f_K(\lambda_K) = \partial_\lambda F(\lambda_K, \omega_K) - \sum_{k=0}^{K-1} \beta_k \partial_{\omega\lambda} G(\lambda_K, \omega_K) \times \prod_{s=k+1}^{K-1} (I - \beta_s \partial_{\omega^2} G(\lambda_s, \omega_s)) \partial_\omega F(\lambda_k, \omega_k)$$

This almost fits the forward mode of the general formulation in Lemma 2 except that the outer state is a sequence $\{\lambda_k\}$. In Lemma 3, we show that $\omega_K \to \omega_\lambda$ is a sufficient condition of $\nabla_\lambda f_K(\lambda) \to \nabla_\lambda f(\lambda)$. It is reasonable to guess that if $\lambda_K \to \lambda$ and $\omega_K \to \omega_\lambda$, the convergence is also guaranteed. But the analysis is more challenging than Lemma 3 as the three terms λ_k, ω_k and $\nabla_\lambda f_k$ entangle with each other. However, we will show in the next section that $\lambda_K \to \lambda^*$ when $K \to \infty$, where λ^* is the optimal point of the outer

Algorithm 1 Fully Single Loop Bilevel Optimization Algorithm (FSLA)

1: **Input:** Initial state $\lambda_0 \in \Lambda$, $\omega_0 \in R^n$; the number of hyper-iterations K; constants c_τ, c_β, c_η, δ
2: **for** $k \leftarrow 0$ **to** $K-1$ **do**
3: $\quad \alpha_k \leftarrow \delta/\sqrt{k}$, $\lambda_{k+1} \leftarrow \lambda_k - \alpha_k d_k$
4: $\quad \tau_{k+1} \leftarrow c_\tau \alpha_k$, $\beta_{k+1} \leftarrow c_\beta \alpha_k$ and $\eta_{k+1} \leftarrow c_\eta \alpha_k$
5: $\quad$ Sample $\xi_{k+1}(\xi_{k+1,1} - \xi_{k+1,5})$
6: $\quad \omega_{k+1} \leftarrow \omega_k - \tau_{k+1}\partial_\omega G(\lambda_{k+1}, \omega_k; \xi_{k+1,1})$
7: $\quad v_{k+1} \leftarrow \beta_{k+1}\partial_\omega F(\lambda_{k+1}, \omega_k; \xi_{k+1,2}) + (I - \beta_{k+1}\partial_{\omega^2} G(\lambda_{k+1}, \omega_k; \xi_{k+1,3}))v_k$
8: $\quad \nabla f_{k+1}(\xi_{k+1}) \leftarrow \partial_\lambda F(\lambda_{k+1}, \omega_{k+1}; \xi_{k+1,4}) - \partial_{\omega\lambda} G(\lambda_{k+1}, \omega_{k+1}; \xi_{k+1,5})v_{k+1}$
9: $\quad d_{k+1} \leftarrow \nabla f_{k+1}(\xi_{k+1}) + (1-\eta_{k+1})(d_k - \nabla f_k(\xi_{k+1}))$;
10: **end for**

function $f(\lambda)$.

Finally in Algorithm 1, we provide the pseudo code of our fully single loop algorithm. Compared to Eq. (2.10), we assume access of the stochastic estimate of related values. What's more, we also maintain a momentum of the hyper-gradient d_k with variance reduction correction in Line 10. This term is used to control the stochastic noise. The momentum-based variance reduction technique is recently widely used in the single level stochastic optimization, such as STORM [30].

2.5 Theoretical Analysis

In this section, we prove the convergence of our proposed single loop bilevel optimization algorithm. We consider the non-convex-strongly-convex case. We first briefly state some assumptions needed in our theoretical analysis for the convenience of discussion. A formal description of the assumptions is in the Appendix A.

We first bound the one iteration progress in the following lemma, which follows the usage of smoothness assumption.

Lemma 6. *Under Assumptions A, B, C, we have:*

$$\begin{aligned}E[f(\lambda_{k+1})] \leq & f(\lambda_k) - \frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \alpha_k \Gamma_2^2 E[||\omega_k - \omega_{\lambda_k}||^2] \\ & + 4\alpha_k \Gamma_1^2 E[||v_k - v_{\lambda_k}||^2] + \alpha_k E[||\nabla f_k - d_k||^2] - \frac{\alpha_k}{2}(1 - \alpha_k L_f) E[||d_k||^2]\end{aligned}$$

where $v_\lambda = (I - \partial_\omega \Phi(\lambda, \omega_\lambda))^{-1} \partial_\omega F(\lambda, \omega_\lambda)$ *and* $\Gamma_1^2 = C_{G,\omega\lambda}^2$, $\Gamma_2^2 = 2L_{F,\lambda}^2 + 4C_{F,\omega}^2 L_{G,\omega\lambda}^2/\mu_G^2$ *are constants.*

The proof is in Appendix D.2. Lemma 6 shows that there are three kinds of errors at each iteration: estimation error of ω_{λ_k} ($||\omega_{k+1} - \omega_{\lambda_k}||^2$), error of v_{λ_k} ($||v_{k+1} - v_{\lambda_k}||^2$) and error of momentum d_k ($||\nabla f_k - d_k||^2$). We denote them as A_k, B_k and C_k in the remainder of this section. The three errors entangle with each other as shown by Line 7-10 of Algorithm 1, *e.g.* the estimation error to $\omega_{\lambda_{k-1}}$ will contribute to the error of momentum d_k as shown in Line 8. In fact, we can bound them with the following inequality (use A_k as an example): $A_k \leq \beta A_{k-1} + C_1 B_{k-1} + C_2 C_{k-1} + C_3$ with $\beta < 1$ and C_1, C_2, C_3 are terms not relevant to A_k, B_k and C_k. Please check the Appendix D.3 for more details. Specially, the momentum term C_k reduces the variance similarly to that in the single level variance-reduction optimizers, by which we mean:

$$\begin{aligned}E[||d_k - \nabla f_k||^2] \leq & (1 - \eta_k)^2 E[||d_{k-1} - \nabla f_{k-1}||^2] + 2\eta_k^2 \sigma^2 \\ & + 2(1 - \eta_k)^2 \underbrace{E[||\nabla f_k(\xi_k) - \nabla f_{k-1}(\xi_k)||^2]}_{\Pi}\end{aligned}$$

The part of noise proportional to $(1 - \eta_k^2)\sigma^2$ is absorbed in the term Π due to the correction made by $\nabla f_{k-1}(\xi_k)$ in the d_k update rule. However, the key difference is that Π not only relies on $E[|||d_{k-1}|||^2]$ but also A_{k-1} and B_{k-1}.

Finally, to show the the convergence of Algorithm 1, we denote the following potential function:

$$\Phi_k = f(\lambda_k) + D_1 A_k + D_2 B_k + D_3 C_k$$

where D_1, D_2 and D_3 are some constants. Combine Lemma 6 with the inequalities for A_k, B_k and C_k, and we have: $\Phi_{k+1} - \Phi_k \leq -\alpha_k/2||\nabla f(\lambda_k)||^2 + C\alpha_k^2\sigma^2$ where C is some constant. With the above inequality, we can get a convergence rate of $O(1/\sqrt{K})$ $(O(1/\epsilon^2))$ by choosing the learning rate with $O(1/\sqrt{k})$. More formally, we have:

Theorem 7. *With Assumption A, B, C hold, and take* $\beta_k = c_\beta \alpha_k$, $\tau_k = c_\tau \alpha_k$, $\eta_k = c_\eta \alpha_k$, *and* $\alpha_k = \frac{\delta}{\sqrt{k}}$. *We have:*

$$\frac{1}{K}\sum_{k=0}^{K-1} ||\nabla f(\lambda_k)||^2 \leq \frac{2\Phi_0}{\delta\sqrt{K}} + \frac{2\delta\bar{C}\sigma^2}{\sqrt{K}}$$

where $\bar{C}$, δ, c_β, c_τ *and* c_η *are some constants.*

The proof of the theorem is included in Appendix D.4.

2.6 Numerical Experiments

In this section, we perform experiments to empirically verify the effectiveness of our algorithm. We first perform experiments over a quadratic objective with synthetic dataset to validate Lemma 3. Then we perform a common benchmark task in bilevel optimization: data hyper-cleaning. The experiments are run over a machine with Intel Xeon E5-2683

CPU and 4 Nvidia Tesla P40 GPUs.

2.6.1 Synthetic Dataset: Quadratic Objective

In this experiment, we verify Lemma 3 over some synthetic data. We consider the bilevel optimization problem where both outer and inner problems are quadratic. To make it simpler, the outer problem does not depend on the outer variable directly. More precisely, we study:

$$\min_{\lambda\in\Lambda} f(\lambda) \coloneqq ||A_o\omega_\lambda - b_o||^2 \; s.t. \; \omega_\lambda = \arg\min_{\omega} ||A_{i,\lambda}\lambda + A_{i,\omega}\omega - b_i||^2$$

For this bilevel problem, we can solve the exact minimizer of the inner problem and then evaluate the exact hyper-gradient based on the Proposition 1. This makes it easier to compute and compare the approximation error of different methods. In experiments, we pick problem dimension 5 and randomly sample 10000 data points. We construct the dataset as follows: first randomly sample $A_o, A_{i,\lambda}, A_{i,\omega} \in \mathbb{R}^{10^4\times 5}$ and $\lambda, \omega, \omega_\lambda \in \mathbb{R}^5$ from the Uniform distribution, then we construct b_o and b_i by $A_o\omega_\lambda + \sigma_o$ and $A_{i,\lambda}\lambda + A_{i,\omega}\omega + \sigma_i$, where σ_i and σ_o are Gaussian noise with mean zero and variance 0.1. We use this simple task to validate our claim in Lemma 3. More precisely, we fix the outer state λ and estimate the hyper-gradient with different sequence $\{\omega_k\}, \{\beta_k\}, \{p_k\}$ and then compare their estimation errors. The results are shown in Figure 2.1.

We perform two sets of experiments. The first set includes our single loop method FSLA, and NS, BP and CG, which are three cases discussed in Lemma 2. For these methods, we generate the sequence $\{\omega_k\}$ through solving the inner problem with K steps

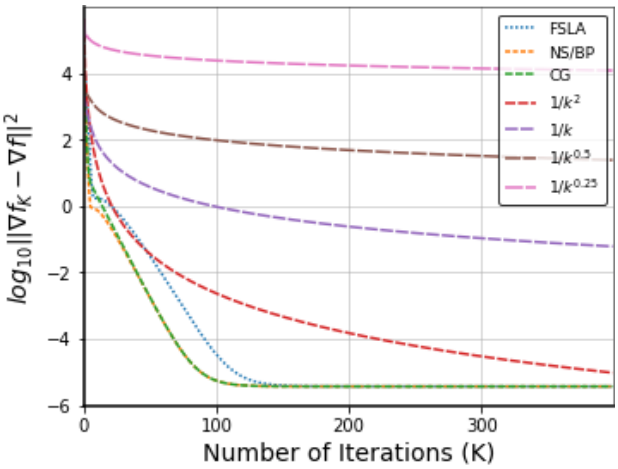


Figure 2.1: The estimation error of hyper-gradient $||\nabla f_K - \nabla f||^2$ for different sequences $\{\omega_k\}$.

of gradient descent and we use learning rate β_k we pick 2×10^{-5}. Note since the second order derivatives of the quadratic objective are constant, BP and NS are the same. As shown by the figure, the hyper-gradient estimation errors of all the four methods converge. Our FSLA takes a bit more number of iterations to converge, however, our algorithm takes less running time. Our method requires $O(1)$ matrix-vector query for every given K, while the other methods requires $O(K)$ queries. In the next set of experiments, we compare with some synthetic $\{\omega_k\}$ sequences which have different convergence rate. More precisely, we use sequence $\{\omega + \tilde{\omega}/(k^\alpha)\}$, where $\tilde{\omega}$ is some random start point, we pick different alpha values (2, 1, 0.5, 0.25). We estimate $\nabla_\lambda f$ according to Corollary 4 (same as the NS), the learning rate is chosen as 2×10^{-5}. Corollary 4 bounds the convergence rate of $\nabla_\lambda f_K$ with the rate of ω_k, which is well verified through the results shown in Figure 2.1.

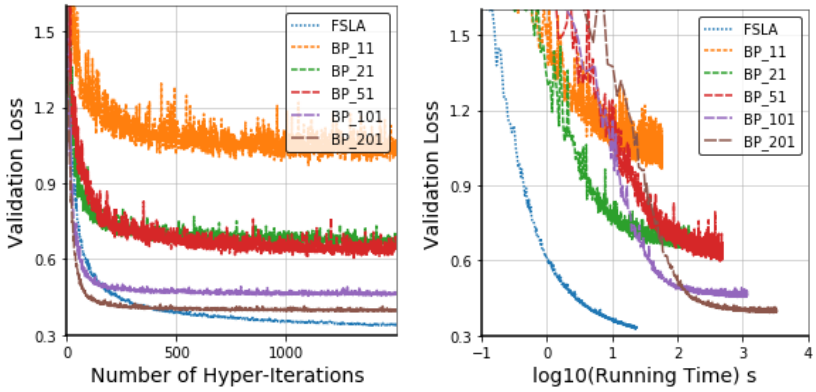


Figure 2.2: FSLA *vs* BP plot of Validation Loss w.r.t Number of hyper-iterations (Left) and $log10$(Running Time) (Right). The perturbation rate γ is 0.8. The post-fix of legend represents the number of inner iterations T.

2.6.2 Hyper Data-cleaning

In this experiment, we demonstrate the efficiency of our FSLA, especially the effect of tracking hyper-gradient history with v_k. More precisely, we compare with three hyper-gradient evaluation methods: BP, NS and CG. For NS and CG methods, we consider both the double loop version and the single loop version. In the single loop version, we update the inner variable with the warm start strategy.

Data cleaning denotes the task of cleaning a noisy dataset. Suppose there is a noisy dataset D_i with N_i samples (the label of some samples are corrupted), the aim of the task is to identify those corrupted data samples. Hyper Data-cleaning approaches this problem by learning a weight per sample. More precisely, we solve the following bilevel optimization problem:

$$\min_{\lambda \in \Lambda} l(\omega_\lambda; D_o) \ s.t. \ \omega_\lambda = \arg\min_{\omega \in \mathbb{R}^d} \frac{1}{N_i} \sum_{j=1}^{N_i} \sigma(\lambda_j) l(\omega, D_{i,j})$$

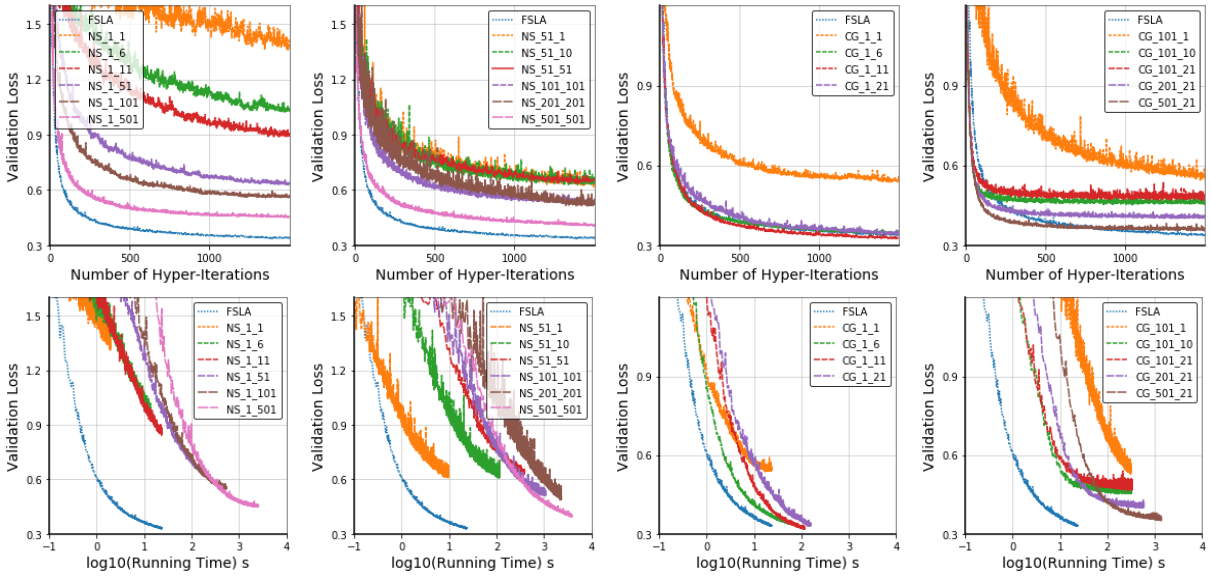


Figure 2.3: FSLA *vs* NS and CG plot of Validation Loss w.r.t Number of hyper-iterations (Top) and $log10$(Running Time) (Bottom). The perturbation rate γ is 0.8. The post-fix of legend represents the number of inner iterations T and approximate steps K.If inner gradient steps equal to 1, we use the warm start trick, otherwise not.

In the inner problem, we minimize a weighted average loss l over the training dataset D_i, with $\sigma(\lambda)$ the sample-wise weight ($\sigma(\cdot)$ is a normalization function and we use $Sigmoid$ in experiments), suppose the minimizer of the inner problem is ω_λ. In in the outer problem, we evaluate ω_λ (a function of λ) over a validation dataset D_o. Then the bilevel problem will find λ such that ω_λ is optimal as evaluated by the validation set.

More specifically, we perform this task over several datasets: MNIST [52], Fashion-MNIST [53] and QMNIST [54]. We construct the datasets as follows: for the training set D_i, we choose 5000 images from the training set, and randomly perturb the label of γ percentage of images. While for the validation set D_o, we randomly select another 5000 images but without any perturbation (all labels are correct). We adopt a 4-layer convolutional neural network in the training. The experimental results are shown in Figure 2.2 and Figure 2.3. For all the methods, we solve the inner problem with stochastic gradient descent, while

for the outer optimizer, all the methods use the variance reduction for fair comparison. In experiments, we vary the number of inner gradient descent steps T and the number of hyper-gradient approximation steps K. The legend in the figures has the form of *method-T-K*. For other hyper-parameters, we perform grid search for each method and choose the best one (hyper-parameters selection is in Appendix E).

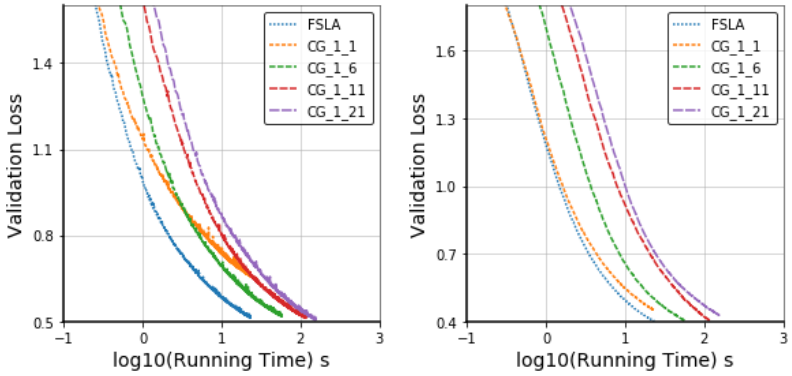


Figure 2.4: FSLA *vs* CG plot of Validation Loss w.r.t $log10$(Running Time). The Left plot shows Fashion-MNIST and the right plot shows the QMNIST. γ is set 0.8.

As shown by the figures, our method converges much faster than the baseline methods. Compared with BP, FSLA surpasses the best BP variant BP_{201} at the 500 iteration, as for the running time, FSLA runs much faster. For NS and CG, $T = 1$ in figures represents using the warm start, where the inner variable is updated from the state of last hyper-iteration. The warm start trick has some kind of acceleration effects. However, for NS, the running time is dominated by the evaluation of Neumann Series. For CG, it converges with around 10 steps, but the extra time is still considerable compared to FSLA. Even with similar computation cost, FSLA still outperforms CG. *CG_1_1* and FSLA both perform one step update per hyper-iteration, but CG converges much slower. This is due to failure of reusing the historical information. Finally, Figure 2.4 includes results for Fashion-MNIST

and QMNIST, where we compare with the best baseline method CG. Our FSLA still outperforms it.

2.7 Proof of Theorems

We first state the full assumptions needed in our analysis:

Assumption 8. *(Outer function) Function F has the following properties:*

a) $F(\lambda, \omega)$ is possibly non-convex, $\partial_\lambda F(\lambda, \omega)$ and $\partial_\omega F(\lambda, \omega)$ are Lipschitz continuous with constant $L_{F,\lambda}$ and $L_{F,\omega}$ respectively

b) $\|\partial_\lambda F(\lambda, \omega)\| \leq C_{F,\lambda}$ and $\|\partial_\omega F(\lambda, \omega)\| \leq C_{F,\omega}$ for some constants $C_{F,\lambda}$ and $C_{F,\omega}$

Assumption 9. *(Inner function) Function G has the following properties:*

a) $G(x, y)$ is continuously twice differentiable, and μ_G-strongly convex w.r.t ω for any given λ

b) $\partial_\omega G(\lambda, \omega)$ is Lipschitz continuous with constant $L_{G,\omega}$, $\|\partial^2_{\omega\lambda} G(\lambda, \omega)\| \leq C_{G,\omega\lambda}$ for some constant $C_{G,\omega\lambda}$

c) $\partial^2_{\omega\lambda} G(\lambda, \omega)$ and $\partial_{\omega^2} G(\lambda, \omega)$ are Lipschitz continuous with constant $L_{G,\omega\lambda}$ and $L_{G,\omega\omega}$ respectively

For convenience, we assume Assumption A and Assumption B also hold for their stochastic version. Next we make the bounded stochastic noise assumption, *i.e.*:

Assumption 10. *(Bounded Variance) We have access to a stochastic unbiased estimate of $\partial^2_{\omega^2} G(\lambda, \omega)$, $\partial^2_{\omega\lambda} G(\lambda, \omega)$, $\partial_\omega G(\lambda, \omega)$, $\partial_\omega F(\lambda, \omega)$, $\partial_\lambda F(\lambda, \omega)$ with their variance bounded by σ^2:*

a) $E[\partial_\lambda F(\lambda, \omega; \xi)] = \partial_\lambda F(\lambda, \omega)$ *and* $var(\partial_\lambda F(\lambda, \omega; \xi)) \leq \sigma^2$

b) $E[\partial_\omega F(\lambda, \omega; \xi)] = \partial_\omega F(\lambda, \omega)$ *and* $var(\partial_\omega F(\lambda, \omega; \xi)) \leq \sigma^2$

c) $E[\partial_\omega G(\lambda, \omega; \xi)] = \partial_\omega G(\lambda, \omega)$ *and* $var(\partial_\omega F(\lambda, \omega; \xi)) \leq \sigma^2$

d) $E[\partial^2_{\omega^2} G(\lambda, \omega; \xi)] = \partial^2_{\omega^2} G(\lambda, \omega)$ *and* $var(\partial^2_{\omega^2} G(\lambda, \omega; \xi)) \leq \sigma^2$

e) $E[\partial^2_{\omega\lambda} G(\lambda, \omega; \xi)] = \partial^2_{\omega\lambda} G(\lambda, \omega)$ *and* $var(\partial^2_{\omega\lambda} G(\lambda, \omega; \xi)) \leq \sigma^2$

2.7.1 Propositions

Proposition 11. *(relaxed triangle inequality) Let* $\{x_k\}, k \in K$ *be* K *vectors. Then the following are true:*

1. $||x_i + x_j||^2 \leq (1 + a)||x_i||^2 + (1 + \frac{1}{a})||x_j||^2$ *for any* $a > 0$*, and*
2. $||\sum_{k=1}^K x_k||^2 \leq K \sum_{k=1}^K ||x_k||^2$

Proposition 12. *(separating mean and variance) Let* $\{\Xi_k\}, k \in K$ *be* K *random variables. Suppose that* $E(\Xi_i) = \xi_i$ *and* $Var(\Xi_i) \leq \sigma^2$*, then:*

$$E[||\sum_{i=1}^K \Xi_k||^2] \leq ||\sum_{k=1}^K \xi_i|| + K^2\sigma^2$$

Please refer to Lemma 3 and Lemma 4 in [55] for the proof of the two propositions.

We first restate Proposition 1 as follows:

Proposition 13. *If for any* $\lambda \in \Lambda$*,* ω_λ *is unique, and* $\partial_{\omega^2} G(\lambda, \omega_\lambda)$ *is invertible, we have:*

$$\nabla_\lambda \omega_\lambda = -\partial_{\omega\lambda} G(\lambda, \omega_\lambda) \partial_{\omega^2} G(\lambda, \omega_\lambda))^{-1}$$

and the hyper-gradient w.r.t λ *has the following form:*

$$\nabla_\lambda f = \partial_\lambda F(\lambda, \omega_\lambda) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda)\partial_{\omega^2} G(\lambda, \omega_\lambda))^{-1}\partial_\omega F(\lambda, \omega_\lambda)$$

This is a standard result in Bilevel optimization, we omit the proof here. Please refer to Lemma 2.1 in [6] for more details.

2.7.2 Proof for Lemma 3 and its Application

We first restate Lemma 3 as follows:

Lemma 14. *(Lemma 3 of main text) Suppose we denote* $m_k = \prod_{s=0}^{k}(1 - \mu_G\beta_s)$, $e_{\omega,k} = ||\omega_k - \omega_\lambda||$, $e_{p,k} = ||p_k - \partial_\omega F(\lambda, \omega_\lambda)||$ *for* $k \in [K]$. *Then if* $\lim_{K\to\infty} m_K = 0$, $\lim_{k\to\infty} e_{\omega,K} = 0$, $\lim_{K\to\infty} m_K \sum_{k=0}^{K-1} \beta_k e_{\omega,k}/m_k$ *is finite, in addition, for the backward mode:* $\lim_{K\to\infty} e_{p,K} = 0$*; for the forward mode:* $\lim_{K\to\infty} m_K \sum_{k=0}^{K-1} \beta_k e_{p,k}/m_k$ *is finite. Then we have* $\nabla_\lambda f_K \to \nabla_\lambda f$, *when* $K \to \infty$, *where* $\nabla_\lambda f_K$ *is defined in Lemma 2 of the main text.*

Proof. Firstly, we prove for the backward mode case. We define:

$$A_K = \sum_{k=0}^{K-1} \beta_k \partial_{\omega\lambda} G(\lambda, \omega_k) \prod_{s=k+1}^{K-1} (I - \beta_s \partial_{\omega^2} G(\lambda, \omega_s))$$

and $A^* = \partial_{\omega\lambda} G(\lambda, \omega_\lambda)\partial_{\omega^2} G(\lambda, \omega_\lambda)^{-1}$. Then a recursive equation satisfied by A_k is:

$$\begin{aligned} A_{k+1} =& A_k \left(I - \beta_k \partial_{\omega^2} G(\lambda, \omega_k)\right) + \beta_k \partial_{\omega\lambda}(\lambda, \omega_k) \\ =& (1 - \beta_k)A_k + \beta_k(A_k(I - \partial_{\omega^2} G(\lambda, \omega_k)) + \partial_{\omega\lambda} G(\lambda, \omega_k)) \end{aligned} \tag{2.11}$$

then by reorganizing the terms in the definition of A^*, we have:

$$A^* = A^*(I - \partial_{\omega^2} G(\lambda, \omega_\lambda)) + \partial_{\omega\lambda} G(\lambda, \omega_\lambda) \tag{2.12}$$

By combining Eq. (2.11) and Eq. (2.12), we have:

$$\begin{aligned}
&||A_{k+1} - A^*|| \\
\leq &(1 - \beta_k)||A_k - A^*|| + \beta_k||A_k(I - \partial_{\omega^2} G(\lambda, \omega_k)) + \partial_{\omega\lambda} G(\lambda, \omega_k) \\
&- A^*(I - \partial_{\omega^2} G(\lambda, \omega_\lambda)) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda)|| \\
\overset{(i)}{\leq} &(1 - \beta_k)||A_k - A^*|| + \beta_k||A_k(I - \partial_{\omega^2} G(\lambda, \omega_k)) - A^*(I - \partial_{\omega^2} G(\lambda, \omega_\lambda))|| \\
&+ \beta_k||\partial_{\omega\lambda} G(\lambda, \omega_k) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda)|| \\
\overset{(ii)}{\leq} &(1 - \beta_k)||A_k - A^*|| + \beta_k||(I - \partial_{\omega^2} G(\lambda, \omega_k))||||A_k - A^*|| \\
&+ \beta_k||A^*||||\partial_{\omega^2} G(\lambda, \omega_k) - \partial_{\omega^2} G(\lambda, \omega_\lambda)|| + \beta_k||\partial_{\omega\lambda} G(\lambda, \omega_k) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda)|| \\
\overset{(iii)}{\leq} &(1 - \beta_k + \beta_k||(I - \partial_{\omega^2} G(\lambda, \omega_k))||)||A_k - A^*|| + \beta_k(L_{G,\omega\lambda} + C_{G,\omega\lambda} L_{G,\omega^2}/\mu_G) e_{\omega,k} \\
\overset{(iv)}{\leq} &(1 - \mu_G \beta_k)||A_k - A^*|| + C_1 \beta_k e_{\omega,k}
\end{aligned}$$

where (i) uses the triangle inequality; (ii) uses the Cauchy-schwarz inequality, (iii) uses the property $||A^*|| \leq C_{G,\omega\lambda}/\mu_G$ (The property is straightforward, please refer to Lemma 2.2 in [7] for a proof); in (iv) we denote $C_1 = L_{G,\omega\lambda} + C_{G,\omega\lambda} L_{G,\omega^2}/\mu_G$ for ease of writing. In summary:

$$||A_{k+1} - A^*|| \leq (1 - \mu_G \beta_k)||A_k - A^*|| + C_1 \beta_k e_{\omega,k} \tag{2.13}$$

Multiplying both sides by $1/m_k$ and sum, we have:

$$||A_K - A^*|| \leq m_K \left(||A_0 - A^*|| + C_1 \sum_{k=0}^{K-1} \beta_k e_{\omega,k}/m_k \right)$$

Then since $\nabla_\lambda f_K = \partial_\lambda F(\lambda, \omega_K) - A_K p_K$ and $\nabla_\lambda f = \partial_\lambda F(\lambda, \omega_\lambda) - A^* \partial_\omega F(\lambda, \omega_\lambda)$, so we have:

$$\begin{aligned} ||\nabla_\lambda f_K - \nabla_\lambda f|| =& ||\partial_\lambda F(\lambda, \omega_K) - A_K p_K - \partial_\lambda F(\lambda, \omega_\lambda) + A^* \partial_\omega F(\lambda, \omega_\lambda)|| \\ \overset{(i)}{\leq}& ||\partial_\lambda F(\lambda, \omega_K) - \partial_\lambda F(\lambda, \omega_\lambda)|| + ||\partial_\omega F(\lambda, \omega_\lambda)||||A_K - A^*|| + ||A^*|| e_{p,K} \\ \overset{(ii)}{\leq}& L_{F,\lambda} e_{\omega,K} + \frac{C_{G,\omega\lambda}}{\mu_G} e_{p,K} + L_{F,\omega} ||A_K - A^*|| \end{aligned} \tag{2.14}$$

where (i) uses triangle inequality and Cauchy-Schwartz inequality; (ii) uses Assumption 8.a. It is straightforward to verify $||\nabla_\lambda f_K - \nabla_\lambda f|| \to 0$ as $k \to \infty$. if the conditions in the Lemma is satisfied.

Next for the forward mode, we take similar approaches. Suppose we define

$$v_K = \sum_{k=0}^{K-1} \beta_k \prod_{s=k+1}^{K-1} \left(I - \beta_s \partial_{\omega^2} G(\lambda, \omega_s) \right) p_k$$

and $v^* = \partial_{\omega^2} G(\lambda, \omega_\lambda)^{-1} \partial_\omega F(\lambda, \omega_\lambda)$. Then we get the recursive equation for v_k as:

$$v_{k+1} = \left(I - \beta_k \partial_{\omega^2} G(\lambda, \omega_k) \right) v_k + \beta_k p_k = (1 - \beta_k) v_k + \beta_k ((I - \partial_{\omega^2} G(\lambda, \omega_k)) v_k + p_k) \tag{2.15}$$

then by reorganizing the terms in the definition of v^*, we have:

$$v^* = (I - \partial_{\omega^2} G(\lambda, \omega_\lambda))v^* + \partial_\omega F(\lambda, \omega_\lambda) \tag{2.16}$$

By combine Eq. (2.15) and Eq. (2.16), we have:

$$\begin{aligned}
||v_{k+1} - v^*|| \leq & (1-\beta_k)||v_k - v^*|| + \beta_k||(I - \partial_{\omega^2} G(\lambda, \omega_k))v_k + p_k \\
& - (I - \partial_{\omega^2} G(\lambda, \omega_\lambda))v^* - \partial_\omega F(\lambda, \omega_\lambda)|| \\
\overset{(i)}{\leq} & (1-\beta_k)||v_k - v^*|| \\
& + \beta_k||(I - \partial_{\omega^2} G(\lambda, \omega_k))v_k - (I - \partial_{\omega^2} G(\lambda, \omega_\lambda))v^*|| + \beta_k e_{p,k} \\
\overset{(ii)}{\leq} & (1-\beta_k)||v_k - v^*|| + \beta_k||(I - \partial_{\omega^2} G(\lambda, \omega_k))||||v_k - v^*|| \\
& + \beta_k||v^*||||\partial_{\omega^2} G(\lambda, \omega_k) - \partial_{\omega^2} G(\lambda, \omega_\lambda)|| + \beta_k e_{p,k} \\
\overset{(iii)}{\leq} & (1 - \beta_k + \beta_k||(I - \partial_{\omega^2} G(\lambda, \omega_k))||)||v_k - v^*|| + \beta_k C_{F,\lambda} L_{G,\omega^2}/\mu_G e_{\omega,k} + \beta_k e_{p,k} \\
\overset{(iv)}{\leq} & (1 - \mu_G \beta_k)||v_k - v^*|| + C_2 \beta_k e_{\omega,k} + \beta_k e_{p,k}
\end{aligned}$$

where (i) uses the triangle inequality; (ii) uses the Cauchy-schwarz inequality; (iii) uses the property $||v^*|| \leq C_{F,\lambda}/\mu_G$; (iv) denotes $C_2 = C_{F,\lambda} L_{G,\omega^2}/\mu_G$. In summary:

$$||v_{k+1} - v^*|| \leq (1 - \mu_G \beta_k)||v_k - v^*|| + C_2 \beta_k e_{\omega,k} + \beta_k e_{p,k} \tag{2.17}$$

Multiplying both sides by $1/m_k$ and sum, then we have:

$$||v_K - v^*|| \leq m_K \left(||v_0 - v^*|| + C_2 \sum_{k=0}^{K-1} \beta_k e_{\omega,k}/m_k + \sum_{k=0}^{K-1} \beta_k e_{p,k}/m_k \right)$$

Finally, since $\nabla_\lambda f_K = \partial_\lambda F(\lambda, \omega_K) - \partial_{\omega\lambda} G(\lambda, \omega_K) v_K$ and $\nabla_\lambda f = \partial_\lambda F(\lambda, \omega_\lambda) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda) v^*$, so we have:

$$\begin{aligned}
||\nabla_\lambda f_K - \nabla_\lambda f|| =& ||\partial_\lambda F(\lambda, \omega_K) + \partial_{\omega\lambda} G(\lambda, \omega_K) v_K - \partial_\lambda F(\lambda, \omega_\lambda)) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda) v^*|| \\
\overset{(i)}{\leq}& ||\partial_\lambda F(\lambda, \omega_K) - \partial_\lambda F(\lambda, \omega_\lambda)|| + ||\partial_{\omega\lambda} G(\lambda, \omega_\lambda)|| * ||v_K - v^*|| \\
&+ ||v^*|| * ||\partial_{\omega\lambda} G(\lambda, \omega_K) - \partial_{\omega\lambda} G(\lambda, \omega_\lambda)|| \\
\overset{(ii)}{\leq}& \left(L_{F,\lambda} + \frac{C_{F,\lambda} L_{G,\omega\lambda}}{\mu_G} \right) e_{\omega,k} + C_{G,\omega\lambda} ||v_K - v^*||
\end{aligned}$$

where (i) uses triangle inequality and Cauchy-Schwartz inequality and (ii) uses assumption 8.a. It is straightforward to verify $||\nabla_\lambda f_K - \nabla_\lambda f|| \to 0$ as $k \to \infty$ if the conditions in the lemma are satisfied. This completes the proof. □

Next we show some corollaries that use specific sequences:

Corollary 15. *Given a positive integer K, inner state $\hat{\omega}_K$ and constant β. We use backward mode of s_k and set $\omega_k = \hat{\omega}_K$, $p_k = \partial_\omega F(\lambda, \hat{\omega}_K)$ and $\beta_k = \beta$ for $k \in [K]$. If $\exists\ \epsilon \in (0, 1)$, such that $\beta_k \in (\epsilon/\mu_G, 1/\mu_G)$, then we have: $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(e_{\omega,K})$*

Corollary 16. *Given a positive integer K, we have sequence $\{\hat{\omega}_k\}$ and $\{\hat{\beta}_k\}$ for $k \in [K]$. We use backward mode of s_k and pick $\omega_k = \hat{\omega}_k$, $p_k = \partial_\omega F(\lambda, \hat{\omega}_K)$ and $\beta_k = \hat{\beta}_k$ for $k \in [K]$. Suppose we have $\beta_k = O(k^{-1})$ and $e_{\omega,k} = O(k^{-0.5})$, then: $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(K^{-0.5})$*

Corollary 17. *Given a positive integer K, we have sequence $\{\hat{\omega}_k\}$ and $\{\hat{\beta}_k\}$ for $k \in [K]$. We use forward mode of s_k and pick $\omega_k = \hat{\omega}_k$, $p_k = \partial_\omega F(\lambda, \hat{\omega}_k)$ and $\beta_k = \hat{\beta}_k$ for $k \in [K]$. Suppose we have $\beta_k = O(k^{-1})$ and $e_{\omega,k} = O(k^{-0.5})$, then: $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(K^{-0.5})$*

Proof. We first prove Corollary 15. Based on Lemma 14, Take corresponding values from

Corollary 15, we have:

$$
\begin{aligned}
||A_K - A^*|| &\leq m_K \left(||A_0 - A^*|| + C_1 \sum_{k=0}^{K-1} \beta_k e_{\omega,k}/m_k \right) \\
&\leq (1-\epsilon)^{K+1}||A_0 - A^*|| + \frac{C_1 e_{\omega,K}}{\mu_G} \sum_{k=0}^{K-1} (1-\epsilon)^{K-k} \\
&\leq (1-\epsilon)^{K+1}||A_0 - A^*|| + \frac{C_1(1-\epsilon)(1-(1-\epsilon)^K)}{\mu_G \epsilon} e_{\omega,K} \\
&\leq (1-\epsilon)^{K+1}||A_0 - A^*|| + \frac{C_1(1-\epsilon)}{\mu_G \epsilon} e_{\omega,K} = O(e_{\omega,K})
\end{aligned}
$$

The last equality holds if $e_{\omega,k}$ converges slower than linear rate. Then we combine with Eq. (2.14), and notice that $e_{p,K} = ||\partial_\omega F(\lambda, \hat{\omega}_K) - \partial_\omega F(\lambda, \omega_\lambda)|| \leq L_{F,\omega} e_{\omega,K}$. It is straightforward to verify $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(e_{\omega,K})$. This completes the proof for Corollary 15.

Next we use induction to prove Corollary 16. By the condition in the corollary, we have $\beta_k = \frac{\theta_1}{k+1}$ and $e_{\omega,k} = \frac{\theta_2}{(k+1)^{0.5}}$ for some constants $\theta_1 > 1/2\mu_G$ and θ_2. Now we pick $\theta_3 = max(||A_0 - A^*||, \frac{C_1\theta_1\theta_2}{\mu_G\theta_1 - 0.5})$ with θ_3 some constant. Then the base of induction satisfies naturally, next suppose we have $||A_k - A^*|| \leq \frac{\theta_3}{(k+1)^{0.5}}$. By Eq. (2.13):

$$
\begin{aligned}
||A_{k+1} - A^*|| \leq & (1 - \frac{\mu_G\theta_1}{k+1}) \frac{\theta_3}{(k+1)^{0.5}} + \frac{C_1\theta_1\theta_2}{(k+1)^{1.5}} \leq (1 - \frac{\mu_G\theta_1}{k+1}) \frac{\theta_3}{(k+1)^{0.5}} + \frac{C_1\theta_1\theta_2}{(k+1)^{1.5}} \\
\leq & (1 - \frac{\mu_G\theta_1}{k+1}) \frac{\theta_3}{(k+1)^{0.5}} + \frac{\theta_3(\mu_G\theta_1 - 0.5)}{(k+1)^{1.5}} \leq \frac{\theta_3}{(k+1)^{0.5}} + \frac{-0.5\theta_3}{(k+1)^{1.5}} \\
\leq & \frac{\theta_3}{(k+1)^{0.5}} \left(1 - \frac{1}{2(k+1)}\right) \leq \frac{\theta_3}{(k+1)^{0.5}} \times \frac{(k+1)^{0.5}}{(k+2)^{0.5}} \leq \frac{\theta_3}{(k+2)^{0.5}}
\end{aligned}
$$

To get the second to last inequality, it is equivalent to get $\left(1 - \frac{1}{2(k+1)}\right)^2 \leq 1 - \frac{1}{k+2}$, which is straightforward to verify. Then we combine with Eq. (2.14), and notice that $e_{p,K} = ||\partial_\omega F(\lambda, \hat{\omega}_K) - \partial_\omega F(\lambda, \omega_\lambda)|| \leq L_{F,\omega} e_{\omega,K}$. It is straightforward to verify $||\nabla_\lambda f_K - \nabla_\lambda f|| = O(e_{\omega,K}) = O(K^{-0.5})$. This finishes the proof for Corollary 16. As for Corollary 17, we follow

similar induction steps as in the proof for Corollary 16, but we instead use the recursive relation in Eq. (2.17) □

It is also possible to consider other sequences, for example, when $e_{\omega,k}$ converges in a slower rate such as $O(K^{0.25})$, but these cases are not as realistic and inspiring as the three corollaries here.

2.7.3 Proof for Theorem 5 and the Lemmas

2.7.3.1 Preliminary Results and Notations

We first restate some definitions: $\omega_{\lambda_k} = \arg\min_{\omega} G(\lambda_k, \omega)$; $v_{\lambda_k} = \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k})^{-1} \partial_\omega F(\lambda_k, \omega_{\lambda_k})$; $\nabla f_k = \partial_\lambda F(\lambda_k, \omega_k) - \partial_{\omega\lambda} G(\lambda_k, \omega_k) v_k$, $\nabla f_k(\xi_k)$ denotes its stochastic estimate as in Algorithm 1. The expectation in the subsequent sections is in terms of the randomness of ϵ_k as defined in the Algorithm 1 of the main text. Next, we show some properties for ω_λ and v_λ in the following proposition:

Proposition 18. *With Assumption 8 and 9 hold , we have:*

$$||\omega_{\lambda_1} - \omega_{\lambda_2}|| \leq C_\omega ||\lambda_1 - \lambda_2||, ||v_{\lambda_1} - v_{\lambda_2}|| \leq C_v ||\lambda_1 - \lambda_2||$$

Where $C_\omega = \frac{C_{G,\omega\lambda}}{\mu_G}$ *and* $C_v = \left(\frac{C_{F,\omega} L_{G,\omega\omega}}{\mu_G^2} + \frac{L_{F,\omega}}{\mu_G} \right)(1 + C_\omega)$. *Furthermore, we have* $||v_\lambda|| \leq \frac{C_{F,\omega}}{\mu_G}$, *and we denote it as* $M = \frac{C_{F,\omega}}{\mu_G}$.

Proof. The Lipschitzness of ω_λ is a rephrase of Lemma 2.2, case b) [6]. Please refer to the paper for the proof. Then based on the definition of $v_\lambda = \partial_{\omega^2} G(\lambda, \omega_\lambda)^{-1} \partial_\omega F(\lambda, \omega_\lambda)$, we get

its bound based on Assumption 8.b and 9.a; as for its Lipschitzness, we have:

$$\begin{aligned}
||v_{\lambda_1} - v_{\lambda_2}|| =& ||\partial_{\omega^2} G(\lambda_1, \omega_{\lambda_1})^{-1} \partial_\omega F(\lambda_1, \omega_{\lambda_1}) - \partial_{\omega^2} G(\lambda_2, \omega_{\lambda_2})^{-1} \partial_\omega F(\lambda_2, \omega_{\lambda_2})|| \\
\leq & ||\partial_\omega F(\lambda_1, \omega_{\lambda_1})|| ||\partial_{\omega^2} G(\lambda_1, \omega_{\lambda_1})^{-1} - \partial_{\omega^2} G(\lambda_2, \omega_{\lambda_2})^{-1}|| \\
& + ||\partial_{\omega^2} G(\lambda_2, \omega_{\lambda_2})^{-1}|| ||\partial_\omega F(\lambda_1, \omega_{\lambda_1}) - \partial_\omega F(\lambda_2, \omega_{\lambda_2})|| \\
\leq & \left(\frac{C_{F,\omega} L_{G,\omega\omega}}{\mu_G^2} + \frac{L_{F,\omega}}{\mu_G} \right) (||\lambda_1 - \lambda_2|| + ||\omega_{\lambda_1} - \omega_{\lambda_2}||) \\
\leq & \left(\frac{C_{F,\omega} L_{G,\omega\omega}}{\mu_G^2} + \frac{L_{F,\omega}}{\mu_G} \right) (1 + C_\omega) ||\lambda_1 - \lambda_2||
\end{aligned}$$

To bound the term $||\partial_{\omega^2} G(\lambda_1, \omega_{\lambda_1})^{-1} - \partial_{\omega^2} G(\lambda_2, \omega_{\lambda_2})^{-1}||$, we use the fact that $||H_2^{-1} - H_1^{-1}|| = ||H_1^{-1}(H_2 - H_1)H_2^{-1}|| \leq ||H_1^{-1}|| ||H_2 - H_1|| ||H_2^{-1}||$. Then it is straightforward to get the bound by Assumption 8 and 9. This completes the proof. □

Next we show two useful bounds for $||\omega_k - \omega_{\lambda_k}||^2$ and $||v_k - v_{\lambda_k}||^2$:

Proposition 19. *With Assumption 8 and 9 hold, we have:*

$$E[||\omega_k - \omega_{\lambda_k}||^2] \leq (1 + \gamma_\omega \alpha_{k-1}) E[||\omega_k - \omega_{\lambda_{k-1}}||^2] + \frac{C_\omega^2 \alpha_{k-1}}{\gamma_\omega} (1 + \gamma_\omega \alpha_{k-1}) E[||d_{k-1}||^2]$$

where γ_ω is some constant.

Proof. Firstly, we have:

$$\begin{aligned}
E[||\omega_k - \omega_{\lambda_k}||^2] \overset{(i)}{\leq} & (1+\gamma_\omega\alpha_{k-1})E[||\omega_k - \omega_{\lambda_{k-1}}||^2] + (1+\frac{1}{\gamma_\omega\alpha_{k-1}})E[||\omega_{\lambda_{k-1}} - \omega_{\lambda_k}||^2] \\
\overset{(ii)}{\leq} & (1+\gamma_\omega\alpha_{k-1})E[||\omega_k - \omega_{\lambda_{k-1}}||^2] + (1+\frac{1}{\gamma_\omega\alpha_{k-1}})C_\omega^2 E[||\lambda_{k-1} - \lambda_k||^2] \\
\leq & (1+\gamma_\omega\alpha_{k-1})E[||\omega_k - \omega_{\lambda_{k-1}}||^2] + (1+\frac{1}{\gamma_\omega\alpha_{k-1}})C_\omega^2\alpha_{k-1}^2 E[||d_{k-1}||^2] \\
\leq & (1+\gamma_\omega\alpha_{k-1})E[||\omega_k - \omega_{\lambda_{k-1}}||^2] + \frac{C_\omega^2\alpha_{k-1}}{\gamma_\omega}(1+\gamma_\omega\alpha_{k-1})E[||d_{k-1}||^2]
\end{aligned}$$

where (i) uses Proposition A.1 and we take α as $\gamma_\omega\alpha_{k-1}$. (ii) is based on Proposition C.1. This completes the proof. □

Proposition 20. *With Assumption 8 and 9 hold, we have:*

$$E[||v_k - v_{\lambda_k}||^2] \leq (1+\gamma_v\alpha_{k-1})E[||v_k - v_{\lambda_{k-1}}||^2] + \frac{C_v^2\alpha_{k-1}}{\gamma_v}(1+\gamma_v\alpha_{k-1})E[||d_{k-1}||^2]$$

where γ_v is some constant.

Proof. Firstly, we have:

$$\begin{aligned}
E[||v_k - v_{\lambda_k}||^2] \overset{(i)}{\leq} & (1+\gamma_v\alpha_{k-1})E[||v_k - v_{\lambda_{k-1}}||^2] + (1+\frac{1}{\gamma_v\alpha_{k-1}})E[||v_{\lambda_{k-1}} - v_{\lambda_k}||^2] \\
\overset{(ii)}{\leq} & (1+\gamma_v\alpha_{k-1})E[||v_k - v_{\lambda_{k-1}}||^2] + (1+\frac{1}{\gamma_v\alpha_{k-1}})C_v^2 E[||\lambda_{k-1} - \lambda_k||^2] \\
\leq & (1+\gamma_v\alpha_{k-1})E[||v_k - v_{\lambda_{k-1}}||^2] + (1+\frac{1}{\gamma_v\alpha_{k-1}})C_v^2\alpha_{k-1}^2 E[||d_{k-1}||^2] \\
\leq & (1+\gamma_v\alpha_{k-1})E[||v_k - v_{\lambda_{k-1}}||^2] + \frac{C_v^2\alpha_{k-1}}{\gamma_v}(1+\gamma_v\alpha_{k-1})E[||d_{k-1}||^2]
\end{aligned} \tag{2.18}$$

where (i) uses Proposition A.1 and we take α as $\gamma_v\alpha_{k-1}$. (ii) is based on Proposition C.1. This completes the proof. □

2.7.3.2 Proof for Iteration Progress Lemma 4

Lemma 21. *(Lemma 4 in the main text) With all the assumptions hold, we have:*

$$E[f(\lambda_{k+1})] \leq f(\lambda_k) - \frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \alpha_k\Gamma_2^2 E[||\omega_k - \omega_{\lambda_k}||^2] + 4\alpha_k\Gamma_1^2 E[||v_k - v_{\lambda_k}||^2] + \alpha_k E[||\nabla f_k - d_k||^2] - \frac{\alpha_k}{2}(1 - \alpha_k L_f)E[||d_k||^2]$$

where $\Gamma_1^2 = C_{G,\omega\lambda}^2$, $\Gamma_2^2 = 2L_{F,\lambda}^2 + 4L_{G,\omega\lambda}^2 M^2$, *and* M *is denoted in Proposition 18.*

Proof. Since $f(\lambda)$ is smooth, we have: $f(\lambda_{k+1}) \leq f(\lambda_k) + \langle \nabla f(\lambda_k), \lambda_{k+1} - \lambda_k \rangle + \frac{L_f}{2}||\lambda_{k+1} - \lambda_k||^2$ (The proof of $f(\lambda)$ is smooth can be found in Lemma 2.2 of [6]). Combine with the update rule $\lambda_{k+1} = \lambda_k - \alpha_k d_k$, we have:

$$\begin{aligned}
E[f(\lambda_{k+1})] \leq & f(\lambda_k) - \alpha_k E[\langle \nabla f(\lambda_k), d_k \rangle] + \frac{L_f \alpha_k^2}{2} E[||d_k||^2] \\
\leq & f(\lambda_k) - \frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 - \frac{\alpha_k}{2}E[||d_k||^2] \\
& + \frac{\alpha_k}{2}E[||\nabla f(\lambda_k) - d_k||^2] + \frac{L_f \alpha_k^2}{2} E[||d_k||^2] \\
\leq & f(\lambda_k) - \frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \frac{\alpha_k}{2}E[||\nabla f(\lambda_k) - d_k||^2] - \frac{\alpha_k}{2}(1 - \alpha_k L_f)E[||d_k||^2] \\
\leq & f(\lambda_k) - \frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \alpha_k \underbrace{E[||\nabla f(\lambda_k) - \nabla f_k||^2}_{\Pi} + \alpha_k E[||\nabla f_k - d_k||^2] \\
& - \frac{\alpha_k}{2}(1 - \alpha_k L_f)E[||d_k||^2]
\end{aligned}$$

In the last inequality, we use the relaxed triangle inequality. Next we bound the term Π.

By the definition of ∇f_k and $\nabla f(\lambda_k)$, we have:

$$
\begin{aligned}
&E[||\nabla f_k - \nabla f(\lambda_k)||^2] \\
=&E[||\partial_\lambda F(\lambda_k, \omega_k) - \partial_\lambda F(\lambda_k, \omega_{\lambda_k}) - (\partial_{\omega\lambda} G(\lambda_k, \omega_k) v_k - \partial_{\omega\lambda} G(\lambda_k, \omega_{\lambda_k}) v_{\lambda_k})\,||^2] \\
\overset{(i)}{\leq}&2E[||\partial_\lambda F(\lambda_k, \omega_k) - \partial_\lambda F(\lambda_k, \omega_{\lambda_k})||^2] + 2E[||\partial_{\omega\lambda} G(\lambda_k, \omega_k) v_k - \partial_{\omega\lambda} G(\lambda_k, \omega_{\lambda_k}) v_{\lambda_k}||^2] \\
\overset{(ii)}{\leq}&2L_{F,\lambda}^2 E[||\omega_k - \omega_{\lambda_k}||^2] + 4E[||\partial_{\omega\lambda} G(\lambda_k, \omega_k)(v_k - v_{\lambda_k})||^2] \\
&+ 4E[||(\partial_{\omega\lambda} G(\lambda_k, \omega_k) - \partial_{\omega\lambda} G(\lambda_k, \omega_{\lambda_k})) v_{\lambda_k}||^2] \\
\overset{(iii)}{\leq}&2L_{F,\lambda}^2 E[||\omega_k - \omega_{\lambda_k}||^2] + 4E[||\partial_{\omega\lambda} G(\lambda_k, \omega_k)||^2 * ||v_k - v_{\lambda_k}||^2] \\
&+ 4E[||\partial_{\omega\lambda} G(\lambda_k, \omega_k) - \partial_{\omega\lambda} G(\lambda_k, \omega_{\lambda_k})||^2 * ||v_{\lambda_k}||^2] \\
\overset{(iv)}{\leq}&(2L_{F,\lambda}^2 + 4M^2 L_{G,\omega\lambda}^2) E[||\omega_k - \omega_{\lambda_k}||^2] + 4C_{G,\omega\lambda}^2 E[||v_k - v_{\lambda_k}||^2] \\
\leq&\Gamma_2^2 E[||\omega_k - \omega_{\lambda_k}||^2] + 4\Gamma_1^2 E[||v_k - v_{\lambda_k}||^2]
\end{aligned}
\tag{2.19}
$$

(i) and (ii) uses the relaxed triangle inequality, (iii) is by the smoothness Assumption 8.a for the first term and the Cauchy-Schwartz inequality for the second and the third term; (iv) uses the Assumption 9.b, 9.c and Proposition 18. Combine Eq. (2.19) and Eq. (2.19), we have:

$$
\begin{aligned}
E[f(\lambda_{k+1})] \leq& f(\lambda_k) - \frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \alpha_k \Gamma_2^2 E[||\omega_k - \omega_{\lambda_k}||^2] \\
&+ 4\alpha_k \Gamma_1^2 E[||v_k - v_{\lambda_k}||^2] + \alpha_k E[||\nabla f_k - d_k||^2] - \frac{\alpha_k}{2}(1 - \alpha_k L_f) E[||d_k||^2]
\end{aligned}
$$

This completes the proof. □

2.7.3.3 Bounds for the Three Types of Errors

As shown by Lemma 21 of Section D.2, we have three types of errors: $||\omega_k - \omega_{\lambda_k}||^2$, $||v_k - v_{\lambda_k}||^2$ and $||\nabla f_k - d_k||^2$. We bound these errors in the subsequent three lemmas.

Lemma 22. *With Assumption 8, 9 and 10 hold, and for $0 < \eta_k < 1$, we have:*

$$\begin{aligned}E[||d_k - \nabla f_k||^2] \leq &(1-\eta_k)E[||d_{k-1} - \nabla f_{k-1}||^2] + C_{d,d}\alpha_{k-1}^2 E[||d_{k-1}||^2] + C_{d,n}\alpha_{k-1}^2\sigma^2 \\ &+ C_{d,\omega}\alpha_{k-1}^2 E[||\omega_{k-1} - \omega_{\lambda_{k-1}}||^2] + C_{d,v}\alpha_{k-1}^2 E[||v_{k-1} - v_{\lambda_{k-1}}||^2]\end{aligned}$$

where $C_{d,d} = 2\Gamma_2^2, C_{d,\omega} = 2(c_\tau^2\Gamma_2^2\Gamma_3^2 + 4c_\beta^2\Gamma_1^2\Gamma_4^2), C_{d,v} = 32c_\beta^2\Gamma_1^2\Gamma_3^2, C_{d,n} = 2(c_\tau^2\Gamma_2^2 + 4c_\beta^2(1 + M^2)\Gamma_1^2 + c_\eta^2(1+M^2))$, c_β, c_τ and c_η are some constants.

Proof. By the definition of d_k, we have:

$$\begin{aligned}&E[||d_k - \nabla f_k||^2] = E[||\nabla f_k(\xi_k) + (1-\eta_k)(d_{k-1} - \nabla f_{k-1}(\xi_k)) - \nabla f_k||^2] \\ =&E[||(1-\eta_k)(d_{k-1} - \nabla f_{k-1}) + \nabla f_k(\xi_k) - \nabla f_k + (1-\eta_k) * (\nabla f_{k-1} - \nabla f_{k-1}(\xi_k))||^2] \\ \overset{(i)}{=}&(1-\eta_k)^2 E[||d_{k-1} - \nabla f_{k-1}||^2] + E[||\eta_k(\nabla f_k(\xi_k) - \nabla f_k) \\ &+ (1-\eta_k)(\nabla f_k(\xi_k) - \nabla f_{k-1}(\xi_k) - (\nabla f_k - \nabla f_{k-1}))||^2] \\ \overset{(ii)}{\leq}&(1-\eta_k)^2 E[||d_{k-1} - \nabla f_{k-1}||^2] + 2\eta_k^2 E[||\nabla f_k(\xi_k) - \nabla f_k||^2] \\ &+ 2(1-\eta_k)^2 E[||\nabla f_k(\xi_k) - \nabla f_{k-1}(\xi_k) - (\nabla f_k - \nabla f_{k-1})||^2] \\ \overset{(iii)}{\leq}&(1-\eta_k)^2 E[||d_{k-1} - \nabla f_{k-1}||^2] + 2(1+M^2)\eta_k^2\sigma^2 \\ &+ 2(1-\eta_k)^2 \underbrace{E[||\nabla f_k(\xi_k) - \nabla f_{k-1}(\xi_k)||^2]}_{\Pi}\end{aligned}$$

(i) is because ξ_k is independent of d_{k-1}, (ii) is because of the relaxed triangle inequality,

and (iii) uses the inequality $E[||X||^2] \geq E[||X - E[X]||^2]$. For the term Π:

$$
\begin{aligned}
&E[||\nabla f_k(\xi_k) - \nabla f_{k-1}(\xi_k)||^2] \\
=&E[||\partial_\lambda F(\lambda_k, \omega_k; \xi_{k,4}) - \partial_{\omega\lambda} G(\lambda_k, \omega_k; \xi_{k,5}) v_k - \partial_\lambda F(\lambda_{k-1}, \omega_{k-1}; \xi_{k,4}) \\
&\quad + \partial_{\omega\lambda} G(\lambda_{k-1}, \omega_{k-1}; \xi_{k,5}) v_{k-1}||^2] \\
\overset{(i)}{\leq}&2E[||\partial_\lambda F(\lambda_k, \omega_k; \xi_{k,4}) - \partial_\lambda F(\lambda_{k-1}, \omega_{k-1}; \xi_{k,4})||^2] \\
&\quad + 2E[||\partial_{\omega\lambda} G(\lambda_k, \omega_k; \xi_{k,5}) v_k - \partial_{\omega\lambda} G(\lambda_{k-1}, \omega_{k-1}; \xi_{k,5}) v_{k-1}||^2] \\
\overset{(ii)}{\leq}&2E[||\partial_\lambda F(\lambda_k, \omega_k; \xi_{k,4}) - \partial_\lambda F(\lambda_{k-1}, \omega_{k-1}; \xi_{k,4})||^2] \\
&\quad + 4M^2 E[||\partial_{\omega\lambda} G(\lambda_k, \omega_k; \xi_{k,5}) - \partial_{\omega\lambda} G(\lambda_{k-1}, \omega_{k-1}; \xi_{k,5})||^2] + 4C_{G,\omega\lambda}^2 E[||v_k - v_{k-1}||^2] \\
\leq&(2L_{F,\lambda}^2 + 4M^2 L_{G,\omega\lambda}^2)(E[||\lambda_k - \lambda_{k-1}||^2] + E[||\omega_k - \omega_{k-1}||^2]) + 4C_{G,\omega\lambda}^2 E[||v_k - v_{k-1}||^2] \\
\leq&\alpha_{k-1}^2 \Gamma_2^2 E[||d_{k-1}||^2] \\
&\quad + \underbrace{\Gamma_2^2 E[||\omega_k - \omega_{k-1}||^2]}_{\Pi_1} + 4\Gamma_1^2 \underbrace{E[||v_k - v_{k-1}||^2]}_{\Pi_2}
\end{aligned}
$$

(i) is by the relaxed triangle inequality, (ii) uses Cauchy-Schwartz inequality, Assumption 9.b and 9.c, we also use $||v_k|| \leq M$, this can be proved by induction. Assume $||v_0|| \leq M$, then by the triangle inequality and Cauchy-Schwartz inequality we have: $||v_{k+1}|| \leq \beta_{k+1}||\partial_\omega F(\lambda_{k+1}, \omega_k; \xi_{k+1,2})|| + ||(I - \beta_{k+1}\partial_{\omega^2} G(\lambda_{k+1}, \omega_k; \xi_{k+1,3}))||||v_k|| \leq \beta_{k+1} C_{F,\omega} + (1 - \beta_{k+1}\mu_G)M = M$.

Next we bound the term Π_1 and Π_2 separately. For the term Π_1, by separating mean and variance, we firstly have:

$$
E[||\omega_k - \omega_{k-1}||^2] = \tau_k^2 E[||\partial_\omega G(\lambda_k, \omega_{k-1}; \xi_{k,1})||^2] \leq \tau_k^2 E[||\partial_\omega G(\lambda_k, \omega_{k-1})||^2] + \tau_k^2 \sigma^2
$$

As for the term $E[||\partial_\omega G(\lambda_k, \omega_k; \xi_{k,1})||^2]$, we have:

$$E[||\partial_\omega G(\lambda_k, \omega_{k-1})||^2] \overset{(i)}{=} E[||\partial_\omega G(\lambda_k, \omega_{k-1}) - \partial_\omega G(\lambda_k, \omega_{\lambda_k})||^2] \overset{(ii)}{\leq} L_{G,\omega}^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2]$$

(i) uses the definition of ω_{λ_k}; (ii) uses Assumption 9.b. Next for the term Π_2, firstly by separating mean and variance, we have:

$$\begin{aligned} E[||v_k - v_{k-1}||^2] =& \beta_k^2 E[||\partial_\omega F(\lambda_k, \omega_{k-1}; \xi_{k,2}) - \partial_{\omega^2} G(\lambda_k, \omega_{k-1}; \xi_{k,3}) * v_{k-1}||^2] \\ \leq& \beta_k^2 E[||\partial_\omega F(\lambda_k, \omega_{k-1}) - \partial_{\omega^2} G(\lambda_k, \omega_{k-1}) * v_{k-1}||^2] + \beta_k^2 (1 + M^2)\sigma^2 \end{aligned}$$

As for the term $E[||\partial_\omega F(\lambda_k, \omega_{k-1}) - \partial_{\omega^2} G(\lambda_k, \omega_{k-1}) * v_{k-1}||^2]$, we have:

$$\begin{aligned} & E[||\partial_\omega F(\lambda_k, \omega_{k-1}) - \partial_{\omega^2} G(\lambda_k, \omega_{k-1}) * v_{k-1}||^2] \\ \overset{(i)}{\leq}& E[||\partial_\omega F(\lambda_k, \omega_{k-1}) - \partial_{\omega^2} G(\lambda_k, \omega_{k-1}) * v_{k-1} - (\partial_\omega F(\lambda_k, \omega_{\lambda_k}) - \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k}) * v_{\lambda_k})||^2] \\ \overset{(ii)}{\leq}& 2E[||\partial_\omega F(\lambda_k, \omega_{k-1}) - \partial_\omega F(\lambda_k, \omega_{\lambda_k})||^2] \\ & + 2E[||\partial_{\omega^2} G(\lambda_k, \omega_{k-1}) * v_{k-1} - \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k}) * v_{\lambda_k}||^2] \\ \overset{(iii)}{\leq}& 2L_{F,\omega}^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] + 4M^2 E[||\partial_{\omega^2} G(\lambda_k, \omega_{k-1}) - \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k})||^2] \\ & + 4L_{G,\omega}^2 E[|v_{k-1} - v_{\lambda_k}||^2] \\ \overset{(iv)}{\leq}& 2(L_{F,\omega}^2 + 2M^2 L_{G,\omega\omega}^2) E[||\omega_{k-1} - \omega_{\lambda_k}||^2] + 4L_{G,\omega}^2 E[|v_{k-1} - v_{\lambda_k}||^2] \\ \leq& \Gamma_4^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] + 4\Gamma_3^2 E[|v_{k-1} - v_{\lambda_k}||^2] \end{aligned}$$

where (i) uses the definition of v_{λ_k}; (ii) uses generalized triangle inequality; (iii) uses Assumption 9.b; (iv) uses the Assumption 9.c. Put everything back into Eq. (2.20), and

re-organize the terms, we get:

$$
\begin{aligned}
&E[||\nabla f_k(\xi_k) - \nabla f_{k-1}(\xi_k)||^2] \\
\leq &\alpha_{k-1}^2\Gamma_2^2 E[||d_{k-1}||^2] + (\tau_k^2\Gamma_2^2\Gamma_3^2 + 4\beta_k^2\Gamma_1^2\Gamma_4^2)E[||\omega_{k-1} - \omega_{\lambda_k}||^2] + 16\beta_k^2\Gamma_1^2\Gamma_3^2 E[|v_{k-1} - v_{\lambda_k}||^2] \\
&+ (\tau_k^2\Gamma_2^2 + 4\beta_k^2(1+M^2)\Gamma_1^2)\sigma^2
\end{aligned}
\tag{2.20}
$$

Finally, we put Eq. (2.20) back into Eq. (2.20) and get:

$$
\begin{aligned}
E[||d_k - \nabla f_k||^2] \leq &(1-\eta_k)^2 E[||d_{k-1} - \nabla f_{k-1}||^2] + 2(1-\eta_k)^2\alpha_{k-1}^2\Gamma_2^2 E[||d_{k-1}||^2] \\
&+ 2(1-\eta_k)^2(\tau_k^2\Gamma_2^2\Gamma_3^2 + 4\beta_k^2\Gamma_1^2\Gamma_4^2)E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&+ 32(1-\eta_k)^2\beta_k^2\Gamma_1^2\Gamma_3^2 E[||v_{k-1} - v_{\lambda_k}||^2] \\
&+ 2(1-\eta_k)^2(\tau_k^2\Gamma_2^2 + 4\beta_k^2(1+M^2)\Gamma_1^2)\sigma^2 + 2(1+M^2)\eta_k^2\sigma^2
\end{aligned}
$$

Suppose we have $\tau_k = c_\tau \alpha_{k-1}$, $\beta_k = c_\beta \alpha_{k-1}$ and $\eta_k = c_\eta \alpha_{k-1}$, then:

$$
\begin{aligned}
&E[||d_k - \nabla f_k||^2] \\
\leq &(1-\eta_k)^2 E[||d_{k-1} - \nabla f_{k-1}||^2] + 2(1-\eta_k)^2 \alpha_{k-1}^2 \Gamma_2^2 E[||d_{k-1}||^2] \\
&+ 2(1-\eta_k)^2 \alpha_{k-1}^2 (c_\tau^2 \Gamma_2^2 \Gamma_3^2 + 4 c_\beta^2 \Gamma_1^2 \Gamma_4^2) E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&\quad + 32(1-\eta_k)^2 \alpha_{k-1}^2 c_\beta^2 \Gamma_1^2 \Gamma_3^2 E[|v_{k-1} - v_{\lambda_k}||^2] \\
&+ 2(1-\eta_k)^2 \alpha_{k-1}^2 (c_\tau^2 \Gamma_2^2 + 4 c_\beta^2 (1+M^2) \Gamma_1^2) \sigma^2 + 2(1+M^2) c_\eta^2 \alpha_{k-1}^2 \sigma^2 \\
\leq &(1-\eta_k) E[||d_{k-1} - \nabla f_{k-1}||^2] + 2(1-\eta_k)^2 \alpha_{k-1}^2 \Gamma_2^2 E[||d_{k-1}||^2] \\
&+ 2\alpha_{k-1}^2 (c_\tau^2 \Gamma_2^2 \Gamma_3^2 + 4 c_\beta^2 \Gamma_1^2 \Gamma_4^2) E[||\omega_{k-1} - \omega_{\lambda_k}||^2] + 32 \alpha_{k-1}^2 c_\beta^2 \Gamma_1^2 \Gamma_3^2 E[|v_{k-1} - v_{\lambda_k}||^2] \\
&+ 2\alpha_{k-1}^2 (c_\tau^2 \Gamma_2^2 + 4 c_\beta^2 (1+M^2) \Gamma_1^2) \sigma^2 + 2(1+M^2) c_\eta^2 \alpha_{k-1}^2 \sigma^2
\end{aligned}
$$

we use the fact that $(1-\eta_k)^2 < (1-\eta_k) < 1$ in the last inequality. Then combine with Proposition 19 and Proposition 20, we get:

$$
\begin{aligned}
&E[||d_k - \nabla f_k||^2] \\
\leq &(1-\eta_k) E[||d_{k-1} - \nabla f_{k-1}||^2] + 2\alpha_{k-1}^2 \Gamma_2^2 E[||d_{k-1}||^2] \\
&\quad + 2\alpha_{k-1}^2 (c_\tau^2 \Gamma_2^2 \Gamma_3^2 + 4 c_\beta^2 \Gamma_1^2 \Gamma_4^2)(1 + \gamma_\omega \alpha_{k-1}) E[||\omega_{k-1} - \omega_{\lambda_{k-1}}||^2] \\
&+ 32 \alpha_{k-1}^2 c_\beta^2 \Gamma_1^2 \Gamma_3^2 (1 + \gamma_v \alpha_{k-1}) E[|v_{k-1} - v_{\lambda_{k-1}}||^2] \\
&\quad + 2\alpha_{k-1}^2 (c_\tau^2 \Gamma_2^2 + 4 c_\beta^2 (1+M^2) \Gamma_1^2 + c_\eta^2 (1+M^2)) \sigma^2 \\
\leq &(1-\eta_k) E[||d_{k-1} - \nabla f_{k-1}||^2] + 2\Gamma_2^2 \alpha_{k-1}^2 E[||d_{k-1}||^2] \\
&\quad + 2(c_\tau^2 \Gamma_2^2 \Gamma_3^2 + 4 c_\beta^2 \Gamma_1^2 \Gamma_4^2) \alpha_{k-1}^2 E[||\omega_{k-1} - \omega_{\lambda_{k-1}}||^2] \\
&+ 32 c_\beta^2 \Gamma_1^2 \Gamma_3^2 \alpha_{k-1}^2 E[|v_{k-1} - v_{\lambda_{k-1}}||^2] + 2\left(c_\tau^2 \Gamma_2^2 + 4 c_\beta^2 (1+M^2) \Gamma_1^2 + c_\eta^2 (1+M^2)\right) \alpha_{k-1}^2 \sigma^2
\end{aligned}
$$

In the first inequality, we omit the higher order terms of α_{k-1} for $E[||d_{k-1}||^2]$; In the second inequality, notice $1+\gamma_\omega\alpha_{k-1}$ and $1+\gamma_v\alpha_{k-1}$ is $O(1)$. Then we get the inequality shown in the lemma. This completes the proof. □

Lemma 23. *With Assumption 8, 9, 10 hold and* $0 < \beta_k < 1/\mu_G$*, we have:*

$$\begin{aligned}E[||v_k - v_{\lambda_k}||^2] \leq &(1+\gamma_v\alpha_{k-1})^3(1-\beta_k\mu_G)E[||v_{k-1} - v_{\lambda_{k-1}}||^2] \\ &+ C_{v,\omega}\alpha_{k-1}E[||\omega_{k-1} - \omega_{\lambda_{k-1}}||^2] + C_{v,d}\alpha_{k-1}E[||d_{k-1}||^2] + C_{v,n}\alpha_{k-1}^2\sigma^2\end{aligned}$$

Where $C_{v,\omega} = c_\beta^2\left(L_{F,\omega}^2 + M^2L_{G,\omega\omega}^2\right)/\gamma_v$*,* $C_{v,d} = C_v^2/\gamma_v$*,* $C_{v,n} = c_\beta^2(1+M^2)$ *,*γ_v*,* c_β *are some constants.*

Proof. Following the definition of v_λ, it is easy to get:

$$v_\lambda = \beta\partial_\omega F(\lambda,\omega_\lambda) + (I - \beta\partial_{\omega^2}G(\lambda,\omega_\lambda))\, v_\lambda$$

. Next by the update rule of v_k and separating mean and variance, we have:

$$\begin{aligned}E[||v_k - v_{\lambda_k}||^2] =&E[||\beta_k\partial_\omega F(\lambda_k,\omega_{k-1};\xi_{k,2}) + (I - \beta_k\partial_{\omega^2}G(\lambda_k,\omega_{k-1};\xi_{k,3})) * v_{k-1} \\ &- (\beta_k\partial_\omega F(\lambda_k,\omega_{\lambda_k}) + (I - \beta_k\partial_{\omega^2}G(\lambda_k,\omega_{\lambda_k})) * v_{\lambda_k})\,||^2] \\ \leq&E[||\beta_k\partial_\omega F(\lambda_k,\omega_{k-1}) + (I - \beta_k\partial_{\omega^2}G(\lambda_k,\omega_{k-1})) * v_{k-1} \\ &- (\beta_k\partial_\omega F(\lambda_k,\omega_{\lambda_k}) + (I - \beta_k\partial_{\omega^2}G(\lambda_k,\omega_{\lambda_k})) * v_{\lambda_k})\,||^2] + (1+M^2)\beta_k^2\sigma^2\end{aligned}$$

For the first term in the above equation, we have:

$$
\begin{aligned}
&E[||\beta_k \partial_\omega F(\lambda_k, \omega_{k-1}) + (I - \beta_k \partial_{\omega^2} G(\lambda_k, \omega_{k-1})) * v_{k-1} \\
&\qquad - (\beta_k \partial_\omega F(\lambda_k, \omega_{\lambda_k}) + (I - \beta_k \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k})) * v_{\lambda_k})\,||^2] \\
\overset{(i)}{\leq} &(1 + \frac{1}{\gamma_v \alpha_{k-1}}) \beta_k^2 E[||\partial_\omega F(\lambda_k, \omega_{k-1}) - \partial_\omega F(\lambda_k, \omega_{\lambda_k})||^2] \\
&+ (1 + \gamma_v \alpha_{k-1}) E[||(I - \beta_k \partial_{\omega^2} G(\lambda_k, \omega_{k-1})) * v_{k-1} - (I - \beta_k \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k})) * v_{\lambda_k}||^2] \\
\overset{(ii)}{\leq} &(1 + \frac{1}{\gamma_v \alpha_{k-1}}) L_{F,\omega}^2 \beta_k^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&\qquad + (1 + \gamma_v \alpha_{k-1})(1 + \frac{1}{\gamma_v \alpha_{k-1}}) M^2 \beta_k^2 E[||\partial_{\omega^2} G(\lambda_k, \omega_{k-1}) - \partial_{\omega^2} G(\lambda_k, \omega_{\lambda_k})||^2] \\
&+ (1 + \gamma_v \alpha_{k-1})^2 E[||(I - \beta_k \partial_{\omega^2} G(\lambda_k, \omega_{k-1})) * (v_{k-1} - v_{\lambda_k})||^2] \\
\overset{(iii)}{\leq} &(1 + \frac{1}{\gamma_v \alpha_{k-1}}) L_{F,\omega}^2 \beta_k^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&\qquad + (1 + \gamma_v \alpha_{k-1})(1 + \frac{1}{\gamma_v \alpha_{k-1}}) M^2 L_{G,\omega\omega}^2 \beta_k^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&+ (1 + \gamma_v \alpha_{k-1})^2 (1 - \beta_k \mu_G)^2 E[||(v_{k-1} - v_{\lambda_k})||^2] \\
\leq &\left((1 + \frac{1}{\gamma_v \alpha_{k-1}}) L_{F,\omega}^2 + (1 + \gamma_v \alpha_{k-1})(1 + \frac{1}{\gamma_v \alpha_{k-1}}) M^2 L_{G,\omega\omega}^2 \right) \beta_k^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&+ (1 + \gamma_v \alpha_{k-1})^2 (1 - \beta_k \mu_G)^2 E[||(v_{k-1} - v_{\lambda_k})||^2]
\end{aligned}
$$

where (i), (ii) uses relaxed triangle inequality case 2; (iii) uses Assumption 9. Finally, combine the above two equations together we have:

$$
\begin{aligned}
&E[||v_k - v_{\lambda_k}||^2] \\
&\leq \left((1 + \frac{1}{\gamma_v \alpha_{k-1}}) L_{F,\omega}^2 + (1 + \gamma_v \alpha_{k-1})(1 + \frac{1}{\gamma_v \alpha_{k-1}}) M^2 L_{G,\omega\omega}^2 \right) \beta_k^2 E[||\omega_{k-1} - \omega_{\lambda_k}||^2] \\
&+ (1 + \gamma_v \alpha_{k-1})^2 (1 - \beta_k \mu_G)^2 E[||(v_{k-1} - v_{\lambda_k})||^2] + (1 + M^2) \beta^2 \sigma^2
\end{aligned}
$$

Then we combine with Proposition 19 and 20 to have:

$$
\begin{aligned}
E[||v_k - v_{\lambda_k}||^2] \leq & (1+\gamma_v\alpha_{k-1})^3(1-\beta_k\mu_G)^2 E[||v_{k-1} - v_{\lambda_{k-1}}||^2] + \beta_k^2(1+M^2)\sigma^2 \\
& + \left((1+\frac{1}{\gamma_v\alpha_{k-1}})L_{F,\omega}^2 + (2+\gamma_v\alpha_{k-1}+\frac{1}{\gamma_v\alpha_{k-1}})M^2L_{G,\omega\omega}^2\right)\times \\
& \quad \beta_k^2(1+\gamma_\omega\alpha_{k-1})E[||\omega_{k-1}-\omega_{\lambda_{k-1}}||^2] \\
& \overset{(ii)}{+} \left((1+\frac{1}{\gamma_v\alpha_{k-1}})L_{F,\omega}^2 + (2+\gamma_v\alpha_{k-1}+\frac{1}{\gamma_v\alpha_{k-1}})M^2L_{G,\omega\omega}^2\right)\times \\
& \quad \beta_k^2\frac{C_\omega^2\gamma_v\alpha_{k-1}}{\gamma_\omega}(1+\gamma_\omega\alpha_{k-1})E[||d_{k-1}||^2] \\
& \overset{(iii)}{+} C_v^2(1+\gamma_v\alpha_{k-1})^2(1+\frac{1}{\gamma_v\alpha_{k-1}})(1-\beta_k\mu_G)^2\gamma_v\alpha_{k-1}^2E[||d_{k-1}||^2]
\end{aligned}
$$

Notice that $(1+\gamma_\omega\alpha_{k-1})$ is O(1), so the term (ii) is $\mathrm{O}(\alpha_k^2)$, while (iii) is $\mathrm{O}(\alpha_k)$, so we can omit the term (ii). Then use the fact that $(1-\beta_k\mu_g)^2 < (1-\beta_k\mu_G) < 1$, we have:

$$
\begin{aligned}
E[||v_k - v_{\lambda_k}||^2] \leq & (1+\gamma_v\alpha_{k-1})^3(1-\beta_k\mu_G) E[||v_{k-1} - v_{\lambda_{k-1}}||^2] + \beta_k^2(1+M^2)\sigma^2 \\
& + \left((1+\frac{1}{\gamma_v\alpha_{k-1}})L_{F,\omega}^2 + (2+\gamma_v\alpha_{k-1}+\frac{1}{\gamma_v\alpha_{k-1}})M^2L_{G,\omega\omega}^2\right)\times \\
& \quad \beta_k^2(1+\gamma_\omega\alpha_{k-1})E[||\omega_{k-1}-\omega_{\lambda_{k-1}}||^2] \\
& + C_v^2(1+\gamma_v\alpha_{k-1})^2(1+\frac{1}{\gamma_v\alpha_{k-1}})\alpha_{k-1}^2E[||d_{k-1}||^2]
\end{aligned}
$$

Finally suppose we have $\tau_k = c_\tau\alpha_{k-1}$, $\beta_k = c_\beta\alpha_{k-1}$ and $\eta_k = c_\eta\alpha_{k-1}$, and omit the higher order terms, we get:

$$
\begin{aligned}
E[||v_k - v_{\lambda_k}||^2] \leq & (1+\gamma_v\alpha_{k-1})^3(1-\beta_k\mu_G) E[||v_{k-1} - v_{\lambda_{k-1}}||^2] + c_\beta^2\alpha_{k-1}^2(1+M^2)\sigma^2 \\
& + \left(L_{F,\omega}^2 + M^2L_{G,\omega\omega}^2\right)c_\beta^2\alpha_{k-1}/\gamma_v E[||\omega_{k-1}-\omega_{\lambda_{k-1}}||^2] \\
& + C_v^2\alpha_{k-1}/\gamma_v E[||d_{k-1}||^2]
\end{aligned}
$$

□

Lemma 24. *With Assumption 8, 9 and 10 hold, and* $0 < \tau_k < min(\frac{\mu_G}{L^2_{G,\omega}}, \frac{1}{\mu_G})$*, we have:*

$$\begin{aligned} E[||\omega_k - \omega_{\lambda_k}||^2] \leq & (1-\mu_G\tau_k)(1+\gamma_\omega\alpha_{k-1})E[||\omega_{k-1} - \omega_{\lambda_{k-1}}||^2] \\ & + C_{\omega,d}\alpha_{k-1}E[||d_{k-1}||^2] + C_{\omega,n}\alpha^2_{k-1}\sigma^2 \end{aligned}$$

Where $C_{\omega,d} = C^2_\omega/\gamma_\omega$*,* $C_{\omega,n} = c^2_\tau$*,* γ_ω *and* c_τ *are some constants.*

Proof. Firstly by the update rule of ω_k, we have:

$$\begin{aligned} & E[||\omega_k - \omega_{\lambda_k}||^2] = E||\omega_k - \tau_k\partial_\omega G(\lambda_k, \omega_k; \xi_{k,1}) - \omega_{\lambda_k}||^2] \\ =& E[||\omega_k - \omega_{\lambda_k}||^2] + \tau^2_k E||\partial_\omega G(\lambda_k, \omega_k; \xi_{k,1})||^2] - 2\tau_k E[\langle \omega_k - \omega_{\lambda_k}, E[\partial_\omega G(\lambda_k, \omega_k; \xi_{k,1})]\rangle] \\ =& E[||\omega_k - \omega_{\lambda_k}||^2] + \tau^2_k E||\partial_\omega G(\lambda_k, \omega_k; \xi_{k,1})||^2] - 2\tau_k E[\langle \omega_k - \omega_{\lambda_k}, \partial_\omega G(\lambda_k, \omega_k)\rangle] \\ \overset{(i)}{\leq} & (1-2\mu_G\tau_k)E[||\omega_k - \omega_{\lambda_k}||^2] + \tau^2_k E[||\partial_\omega G(\lambda_k, \omega_k; \xi_{k,1})||^2] \\ \overset{(ii)}{\leq} & (1-2\mu_G\tau_k)E[||\omega_k - \omega_{\lambda_k}||^2] + \tau^2_k E[||\partial_\omega G(\lambda_k, \omega_k) - \partial_\omega G(\lambda_k, \omega_{\lambda_k})||^2] + \tau^2_k\sigma^2 \\ \overset{(iii)}{\leq} & (1-2\mu_G\tau_k + \tau^2_k L^2_{G,\omega})E[||\omega_k - \omega_{\lambda_k}||^2] + \tau^2_k\sigma^2 \overset{(iv)}{\leq} (1-\mu_G\tau_k)E[||\omega_k - \omega_{\lambda_k}||^2] + \tau^2_k\sigma^2 \end{aligned} \tag{2.21}$$

Where (i) uses the strong convexity of G and the inequality for strongly convex function: $\langle \omega_k - \omega_{\lambda_k}, \partial_\omega G(\lambda_k, \omega_k)\rangle \geq \mu_G||\omega_k - \omega_{\lambda_k}||^2$. (ii) uses separating mean and variance and the definition of ω_{λ_k}; (iii) uses Assumption 9.b; (iv) uses the condition that $\tau_k < \mu_G/L^2_{G,\omega}$.

Finally, we combine Eq. (2.21) with Proposition 19, we get:

$$E[||\omega_k - \omega_{\lambda_k}||^2] \leq (1-\mu_G\tau_k)(1+\gamma_\omega\alpha_{k-1})E[||\omega_{k-1} - \omega_{\lambda_{k-1}}||^2] + \frac{C_\omega^2}{\gamma_\omega}\alpha_{k-1}E[||d_{k-1}||^2] + c_\tau^2\alpha_{k-1}^2\sigma^2$$

□

Where we omit the term $1+\gamma_\omega\alpha_{k-1}$ for $E[||d_{k-1}||^2]$, and use the fact that $\tau_k = c_\tau\alpha_{k-1}$. This completes the proof.

2.7.3.4 Proof for the convergence Theorem 5

Theorem 25. *(Theorem 5 in the main text) With Assumption 8, 9 and 10 hold, and suppose* $\beta_k = c_\beta\alpha_{k-1}$, $\tau_k = c_\tau\alpha_{k-1}$, $\eta_k = c_\eta\alpha_{k-1}$, *and* $\alpha_k = \delta/\sqrt{k+1}$, *then we have:*

$$\frac{1}{K}\sum_{k=0}^{K-1}||\nabla f(\lambda_k)||^2 \leq \frac{2\Phi_0}{\delta\sqrt{K}} + \frac{2\delta\bar{C}\sigma^2\ln(K+1)}{\sqrt{K}}$$

where $\gamma_v = 32\Gamma_1^2C_v^2$, $\gamma_\omega = 8\Gamma_2^2C_w^2$

$c_\tau = (2\gamma_v\Gamma_2^2\gamma_\omega + 2\gamma_v\Gamma_2^2 + 4\Gamma_1^2c_\beta^2\left(L_{F,\omega}^2 + M^2L_{G,\omega\omega}^2\right))/(\gamma_v\Gamma_2^2\mu_G)$,$c_\eta = 4\Gamma_2^2/L_f$, $c_\beta = 2(3\gamma_v + 1)/\mu_G$, $\bar{C} = C_{d,n}/c_\eta + 4\Gamma_1^2C_{v,n} + \Gamma_2^2C_{\omega,n}$, δ *is a constant such that it satisfies Eq. (2.30).*

Proof. We firstly denote the following potential function:

$$\Phi_k = f(\lambda_k) + \Gamma_2^2A_k + 4\Gamma_1^2B_k + C_k/c_\eta$$

where $A_k = ||\omega_k - \omega_{\lambda_k}||^2$, $B_k = ||v_k - v_{\lambda_k}||^2$, $C_k = ||\nabla f_k - d_k||^2$. Next we bound $\Phi_{k+1} - \Phi_k$,

which is composed of four terms, we get a bound for each of them based on Lemma 21-24.

By Lemma 21, we have:

$$\begin{aligned} E[f(\lambda_{k+1}) - f(\lambda_k)] \leq & -\frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \alpha_k \Gamma_2^2 E[A_k] \\ & + 4\alpha_k \Gamma_1^2 E[B_k] + \alpha_k E[C_k] - \frac{\alpha_k}{2}(1 - \alpha L_f) E[||d_k||^2] \end{aligned} \tag{2.22}$$

By Lemma 22, we have:

$$E[C_{k+1} - C_k] \leq -c_\eta \alpha_k E[C_k] + C_{d,d}\alpha_k^2 E[||d_k||^2] + C_{d,\omega}\alpha_k^2 E[A_k] + C_{d,v}\alpha_k^2 E[B_k] + C_{d,n}\alpha_k^2 \sigma^2 \tag{2.23}$$

By Lemma 23, we have:

$$\begin{aligned} E[B_{k+1} - B_k] \leq & ((1 + \gamma_v \alpha_k)^3 (1 - \beta_{k+1}\mu_G) - 1) E[B_k] \\ & + C_{v,\omega}\alpha_k E[A_k] + C_{v,d}\alpha_k E[||d_k||^2] + C_{v,n}\alpha_k^2 \sigma^2 \end{aligned} \tag{2.24}$$

By Lemma 24, we have:

$$E[A_{k+1} - A_k] \leq ((1 - \mu_G \tau_{k+1})(1 + \gamma_\omega \alpha_k) - 1) E[A_k] + C_{\omega,d}\alpha_k E[||d_k||^2] + C_{\omega,n}\alpha_k^2 \sigma^2 \tag{2.25}$$

Sum inequalities Eq. (2.22)- (2.25), we have:

$$\Phi_{k+1} - \Phi_k \leq -\frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 + \left(C_{d,n}/c_\eta + 4\Gamma_1^2 C_{v,n} + \Gamma_2^2 C_{\omega,n}\right)\alpha_k^2 \sigma^2 \tag{2.26}$$

The terms related to $E[||d_k||^2]$, $E[A_k]$ and $E[B_k]$ are eliminated by using the facts shown

in Eq. (2.27)- (2.29). Firstly, for terms related to $E[||d_k||^2]$, we have:

$$\frac{2\Gamma_2^2}{c_\eta}\alpha_k^2 + \frac{4\Gamma_1^2 C_v^2}{\gamma_v}\alpha_k + \frac{\Gamma_2^2 C_\omega^2}{\gamma_\omega}\alpha_k < \frac{\alpha_k}{2}(1-\alpha_k L_f) \tag{2.27}$$

Next for the term related to $E[B_k]$, we have:

$$4\Gamma_1^2((1+\gamma_v\alpha_k)^3(1-\beta_{k+1}\mu_G)-1) + 32c_\beta^2\Gamma_1^2\Gamma_3^2\alpha_k^2/c_\eta + 4\Gamma_1^2\alpha_k < 0 \tag{2.28}$$

Finally, for the term involved with $E[A_k]$, we have:

$$\begin{aligned}\Gamma_2^2((1+\gamma_\omega\alpha_k)(1-\mu_G\tau_{k+1})-1) + 2(c_\tau^2\Gamma_2^2\Gamma_3^2 \\ + 4c_\beta^2\Gamma_1^2\Gamma_4^2)\alpha_k^2/c_\eta + 4\Gamma_1^2 c_\beta^2\left(L_{F,\omega}^2 + M^2 L_{G,\omega\omega}^2\right)\alpha_k/\gamma_v + \Gamma_2^2\alpha_k < 0\end{aligned} \tag{2.29}$$

We can verify the correctness of the inequalities Eq. (2.27)- (2.29) by taking the corresponding values of $\gamma_v = 32\Gamma_1^2 C_v^2$, $\gamma_\omega = 8\Gamma_2^2 C_w^2$, $c_\eta = 4\Gamma_2^2/L_f$, $c_\beta = 2(3\gamma_v+1)/\mu_G$, $c_\tau = (2\gamma_v\Gamma_2^2\gamma_\omega + 2\gamma_v\Gamma_2^2 + 4\Gamma_1^2 c_\beta^2\left(L_{F,\omega}^2 + M^2 L_{G,\omega\omega}^2\right))/(\gamma_v\Gamma_2^2\mu_G)$ as stated in the Theorem and let δ satisfy the following condition:

$$\begin{aligned}\delta < min\Bigg(&\frac{1}{4L_f}, \frac{2\Gamma_2^2\mu_G c_\beta}{12\Gamma_2^2\gamma_v^2 + 8\Gamma_3^2 L_f c_\beta^2}, \frac{c_\eta\Gamma_2^2(1+\gamma_\omega)}{2(c_\tau^2\Gamma_2^2\Gamma_3^2 + 4c_\beta^2\Gamma_1^2\Gamma_4^2) + \Gamma_2^2 c_\eta}, \\ &\frac{1}{c_\eta}, \frac{1}{2(3\gamma_v+1)}, \frac{1}{c_\tau\mu_G}, \frac{\mu_G}{c_\tau L_{G,\omega}^2}\Bigg)\end{aligned}$$

Now we summarize both sides of Eq. (2.26), and rearrange the terms:

$$\sum_{k=0}^{K-1}\frac{\alpha_k}{2}||\nabla f(\lambda_k)||^2 \leq (\Phi_0 - \Phi_K) + \bar{C}\sum_{k=0}^{K-1}\alpha_k^2\sigma^2$$

We denote $\bar{C} = C_{d,n}/c_\eta + 4\Gamma_1^2 C_{v,n} + \Gamma_2^2 C_{\omega,n}$. Suppose we take $\alpha_k = \frac{\delta}{\sqrt{k+1}}$ where δ is chosen *s.t.* Eq. (2.30) is satisfied, we get:

$$\frac{\delta}{\sqrt{K}} \sum_{i=0}^{K-1} ||\nabla f(\lambda_k)||^2 \leq \sum_{i=0}^{K-1} \frac{\delta}{\sqrt{k}} ||\nabla f(\lambda_k)||^2 \leq 2\Phi_0 + 2\bar{C}\delta^2\sigma^2 \ln(K+1)$$

Divide both sides by K, we have:

$$\frac{1}{K} \sum_{i=0}^{K-1} ||\nabla f(\lambda_k)||^2 \leq \frac{2\Phi_0}{\delta\sqrt{K}} + \frac{2\delta\bar{C}\sigma^2 \ln(K+1)}{\sqrt{K}}$$

□

2.8 More Experimental Details

In the Hyper Data-cleaning task, we test over MNIST [52], Fashion-MNIST [53] and QMNIST [54]. The datasets are created as described in the main text. In experiments, we use batch-size 256 and run 2000 hyper-iterations. Other hyper-parameters are chosen as follows: we pick $\delta = 1000$, $c_\tau = 0.001$, $c_\beta = 10^{-4}$ and $c_\eta = 9 * 10^{-4}$ for our FSLA. For BP method, we pick $\delta = 1000$, $c_\tau = 0.001$ and $c_\eta = 9 * 10^{-4}$; For NS method, we choose $\delta = 1000$, $c_\tau = 0.001$, $c_\eta = 9 * 10^{-4}$ and β as 0.1; For CG method, we choose $\delta = 1000$, $c_\tau = 0.001$, and $c_\eta = 9 * 10^{-4}$. Next, we present some ablation studies of the hyper-parameters for FSLA, and the results are shown in Figure 2.5 and Figure 2.6. In each plot, we vary one hyper-parameter and fix other hyper-parameters whose values are taken as just mentioned. We do not include hyper-parameters that diverge. For example, when the batch-size is 32 or when c_β is 5×10^{-4}, the algorithm does not converge.

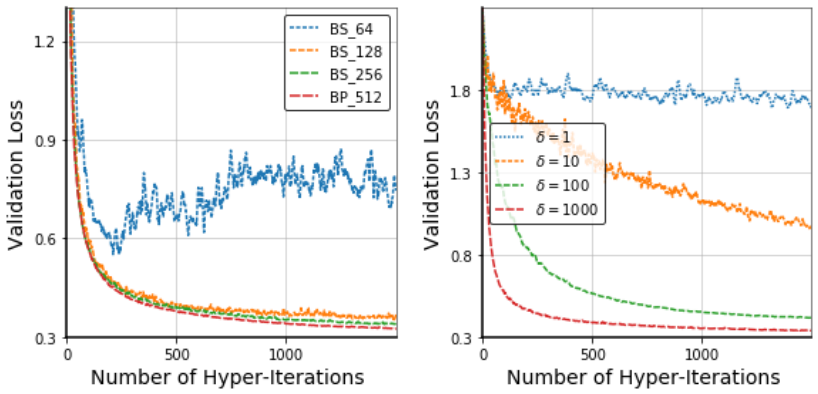


Figure 2.5: Ablation Study for Batch Size (Left) and outer learning rate coefficient δ (Right).

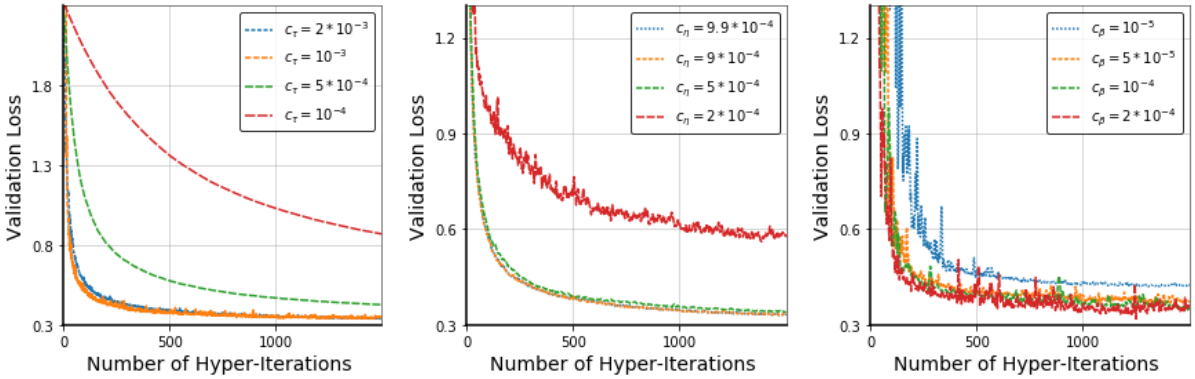


Figure 2.6: Ablation Study for c_τ (Left) and outer c_η (Middle)a and c_β (Right).

Chapter 3: Provably Faster Algorithms for Bilevel Optimization via Without-Replacement Sampling

3.1 Introduction

Bilevel optimization [16, 17] has received a lot of interest recently due to its wide-ranging applicability in machine learning tasks, including hyper-parameter optimization [37], meta-learning [38] and personalized federated learning [56]. A bilevel optimization problem is a type of nested two-level problems as follows:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq f(x, y_x) \text{ s.t. } y_x = \arg\min_{y\in\mathbb{R}^d} g(x, y), \tag{3.1}$$

which includes an outer problem $f(x, y)$ and a inner problem $g(x, y)$. The outer problem $f(x, y)$ relies on the solution y_x of the inner problem $g(x, y)$. Eq. (3.1) can be solved through gradient descent: $x_{t+1} = x_t - \eta\nabla h(x_t)$, with η be the stepsize and $\nabla h(x_t)$ be gradient at the state x_t. Specially, the gradient $\nabla h(x)$ [6] is a function of the inner problem minimizer y_x, first order derivatives $\nabla_x f(x, y)$, $\nabla_y f(x, y)$, $\nabla_y g(x, y)$ and second order derivatives $\nabla_{xy} g(x, y)$, $\nabla_{y^2} g(x, y)$. In particular, y_x can be solved though gradient descent: $y_{t+1} = y_t - \gamma\nabla_y g(x, y_t)$ with γ be the stepsize and $\nabla_y g(x_t, y_t)$ be gradient at the iterate (x_t, y_t). In the standard setup, the outer and inner problems are defined over two datasets $\mathcal{D}_u =$

$\{\xi_i, i \in [m]\}$ and $\mathcal{D}_l = \{\zeta_j, i \in [n]\}$, respectively:

$$f(x,y) = \frac{1}{m}\sum_{i=1}^{m} f(x,y;\xi_i),\ g(x,y) = \frac{1}{n}\sum_{j=1}^{n} g(x,y;\zeta_j),$$

Naturally, we sample from the datasets to estimate the first and second-order derivatives in the hyper-gradient formulation. And to guarantee convergence, previous approaches require the examples be *mutually independent* [12] even at the same state (x, y). For example, two different examples ζ_1 and ζ_2 are sampled to estimate $\nabla_y g(x,y)$ and $\nabla_{y^2} g(x,y)$, respectively. This leads to extra computational cost in practice. Indeed, two backward passes are required to evaluate $\nabla_x f(x,y)$ and $\nabla_y f(x,y)$, while five backward passes are needed to evaluate $\nabla_y g(x,y)$, $\nabla_{y^2} g(x,y)$ and $\nabla_{xy} g(x,y)$ for a given state of (x, y). In contrast, one backward pass are needed to evaluate the former and two backward passes for the latter if not sampling independently for each property. This leads to notable computational cost difference for the current large-scale machine learning models [57]. Therefore, it is beneficial to explore other example-selection strategies beyond independent sampling. More specifically, we aim to answer the following question in this work: *Can we solve the bilevel optimization problem without using independent sampling?*

Although example-selection is under-explored for bilevel optimization, it has been well studied in the single level optimization literature [60, 61, 62]. For single level problems: $\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f(x;\xi_i)$, we iteratively perform the stochastic gradient step $x_{t+1} = x_t - \eta\nabla f(x_t, \xi_t)$, where the sample gradient is used to estimate the full gradient and the example ξ_t is sampled according to some order. More specifically, the sampling schemes are roughly divided into two categories: *with-replacement* sampling and

Table 3.1: **Comparisons of the Bilevel Opt. & Conditional Bilevel Opt. algorithms for finding an ϵ-stationary point**. We include comparison with stochastic gradient descent type of methods and a more comprehensive comparison including other acceleration methods can be found in Table 3.2 of Appendix. An ϵ-stationary point is defined as $\|\nabla h(x)\| \leq \epsilon$. $Gc(f,\epsilon)$ and $Gc(g,\epsilon)$ denote the number of gradient evaluations *w.r.t.* $f(x,y)$ and $g(x,y)$; $JV(g,\epsilon)$ denotes the number of Jacobian-vector products; $HV(g,\epsilon)$ is the number of Hessian-vector products. m and n are the number of data examples for the outer and inner problems, in particular, n is the maximum number of inner problems for conditional bilevel optimization. Our methods have a dependence over example numbers $(max(m,n))^q$, q is a value decided by without-replacement sampling strategy and can have value in $[0,1]$ (A herding-based permutation [1] can let $q=0$).

Setting	Algorithm	$Gc(f,\epsilon)$	$Gc(g,\epsilon)$	$JV(g,\epsilon)$	$HV(g,\epsilon)$
B.O.	BSA [6]	$O(\epsilon^{-4})$	$O(\epsilon^{-6})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	TTSA [15]	$O(\epsilon^{-5})$	$O(\epsilon^{-5})$	$O(\epsilon^{-5})$	$O(\epsilon^{-5})$
	StocBiO [12]	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	SOBA [58]	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	AID/ITD [12]	$O(max(m,n)\epsilon^{-2})$	$O(max(m,n)\epsilon^{-2})$	$O(max(m,n)\epsilon^{-2})$	$O(max(m,n)\epsilon^{-2})$
	WiOR-BO(Ours)	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^q\epsilon^{-3})$
Cond. B.O.	DL-SGD [59]	$O(\epsilon^{-4})$	$O(\epsilon^{-6})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	RT-MLMC [59]	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	WiOR-CBO (Ours)	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^{2q}\epsilon^{-4})$	$O(n(max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^{2q}\epsilon^{-4})$

without-replacement sampling. Traditional stochastic gradient descent typically employs with-replacement sampling to ensure an unbiased estimation of the full gradient in each iteration. In contrast, the without-replacement sampling has biased estimation at each iteration. Despite this, when averaging the gradients of consecutive examples, without-replacement sampling can approximate the full gradient more efficiently [63]. In fact, any permutation-based without-replacement sampling has that the gradient estimation error contracts at rate of $O(K^{-2})$ [63] where K is the number of steps, while stochastic gradient descent only has a rate of $O(K^{-1})$. Inspired by the properties of without-replacement sampling demonstrated in the single-level optimization, we design the algorithm named WiOR-BO (and its variants) which performs without-replacement sampling for bilevel optimization. Naturally, WiOR-BO does not require the independent sampling as in previous methods. In fact, our algorithm enjoys favorable convergence rate. As shown in Table 3.1, WiOR-BO converges faster compared with their counterparts (*e.g.* StocBiO [12]) using in-

dependent sampling. Actually, WiOR-BO gets comparable convergence rate with methods using more complicated variance reduction techniques (*e.g.* MRBO [32]). In Table 3.2, we perform a more comprehensive comparison with other variance reduction techniques and in Section 3.8 of Appendix, we further compare WiOR-BO with these methods.

The **contributions** of our work can be summarized as:

1. We propose **WiOR-BO** which performs without-sampling to solve the bilevel optimization problem and converges to an ϵ stationary point with a rate of $\mathrm{O}(\epsilon^{-3})$. This rate improves over the $\mathrm{O}(\epsilon^{-4})$ rate of its counterparts (*e.g.* stocBiO [12]) using independent sampling.

2. We propose **WiOR-CBO** for the conditional bilevel optimization problems and show that our algorithm converges to an ϵ stationary point with a rate of $\mathrm{O}(\epsilon^{-4})$. This rate improves over the $\mathrm{O}(\epsilon^{-6})$ rate of its counterparts (*e.g.* DL-SGD [59]) using independent-sampling.

3. We customize our algorithms to the special cases of minimax and compositional optimization problems and demonstrate similar favorable convergence rates.

Notations. ∇ (∇_x) denotes full gradient (partial gradient, over variable x), higher order derivatives follow similar rules. $[n]$ represents sequence of integers from 1 to n. $O(\cdot)$ is the big O notation, and we hide logarithmic terms.

3.2 Related Works

Bilevel Optimization has its roots tracing back to the 1960s, beginning with the regularization method proposed by [16], and then followed by many research works [8,

17, 19, 64]. In the field of machine learning, similar implicit differentiation techniques were used to solve Hyper-parameter Optimization [20, 21, 23]. Exact solutions for Bilevel Optimization solve the inner problem for each outer variable but are inefficient. More efficient algorithms solve the inner problem with a fixed number of steps, and use the 'back-propagation through time' technique to compute the hyper-gradient [24, 25, 26, 27, 28]. Recently, single loop algorithms [6, 12, 13, 15, 32, 58, 65, 66] are introduced to perform the outer and inner updates alternatively. Besides, other aspects of the bilevel optimization are also investigated: [59] studied a type of conditional bilevel optimization where the inner problem depends on the example of the outer problem; [67, 68] studied the Hessian-free bilevel optimization and proposed an algorithm with optimal convergence rate; [69] proposed an efficient single loop algorithm for the general non-convex bilevel optimization; [70, 71] studies the bilevel optimization in the federated learning setting.

Sample-selection in stochastic optimization has been extensively studied for the case of single level optimization problems [60, 72, 73, 74]. Beyond the with-replacement sampling adopted by SGD, various without-replacement strategies were also studied in the literature. The random-reshuffling strategy shuffles the data samples at the start of each training epoch, and its property was first studied in [61] via the non-commutative arithmetic-geometric mean conjecture and followed by [73, 75]. [62, 76] studied the shuffling-once strategy, where the data samples are only shuffled at the start of the training once; [77] investigated data echoing where one data sample is repeatedly used. Quasi-Monte Carlo sampling is also applied in stochastic gradient descent [63, 78]. Recently, [1, 63] proposed to use average gradient error to study the convergence of various example order strategies and then proposed a herding-based sampling strategy [79] to proactively minimize the average

gradient error.

3.3 Without-Replacement Sampling in Bilevel Optimization

In this work, we focus on the following finite-sum bilevel optimization problem:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq f(x, y_x) = \frac{1}{m}\sum_{i=1}^{m} f(x, y_x; \xi_i), \text{ s.t. } y_x = \arg\min_{y\in\mathbb{R}^d} g(x, y) = \frac{1}{n}\sum_{j=1}^{n} g(x, y; \zeta_j) \tag{3.2}$$

where both the outer and the inner problems are defined over a finite dataset, and we denote them as $\mathcal{D}_u = \{\xi_i, i \in [m]\}$ and $\mathcal{D}_l = \{\zeta_j, j \in [n]\}$, respectively. Under mild assumptions, the hyper-gradient of $h(x)$ is:

$$\nabla h(x) = \nabla_x f(x, y_x) - \nabla_{xy} g(x, y_x) u_x, \; u_x = \nabla_{y^2} g(x, y_x)^{-1} \nabla_y f(x, y_x) \tag{3.3}$$

Our objective is to minimize $h(x)$ by exploiting Eq. (3.3) with examples from $\mathcal{D}_u$ and $\mathcal{D}_l$ at each step. More specifically, for a given example order π denoting the following example-selection sequence as $\{\xi_t^\pi \in \mathcal{D}_u\}$ and $\{\zeta_t^\pi \in \mathcal{D}_l\}$ for $t \in [T]$, where T denotes the length of the sequence and can be arbitrarily large. We then perform the following update steps iteratively for $t \in [T]$:

$$y_{t+1} = y_t - \gamma_t \nabla_y g(x_t, y_t; \zeta_t^\pi) \tag{3.4a}$$

$$u_{t+1} = u_t - \rho_t(\nabla_{y^2} g(x_t, y_t; \zeta_t^\pi) u_t - \nabla_y f(x_t, y_t; \xi_t^\pi)) \tag{3.4b}$$

$$x_{t+1} = x_t - \eta_t(\nabla_x f(x_t, y_t; \xi_t^\pi) - \nabla_{xy} g(x_t, y_t; \zeta_t^\pi) u_t) \tag{3.4c}$$

Algorithm 2 Without-Replacement Bilevel Optimization (WiOR-BO)

Input: Initial states x_I^0, y_I^0 and u_I^0; learning rates $\{\gamma_i^r, \rho_i^r, \eta_i^r\}, i \in [I], r \in [R]$, $I := \mathrm{lcm}(m, n)$ denotes the least common multiple of m and n.
for epochs $r = 1$ **to** R **do**
  Randomly sample I/m permutations of outer dataset and concatenate them to have $\{\xi_i^\pi, i \in [I]\}$, sample I/n permutations of inner dataset and concatenate them to have $\{\zeta_i^\pi, i \in [I]\}$;
  Set $y_0^r = y_I^{r-1}$, $u_0^r = \mathcal{P}_\iota(u_I^{r-1})$ and $x_0^r = x_I^{r-1}$
  for $i = 0$ **to** $I - 1$ **do**
    $y_{i+1}^r = y_i^r - \gamma_i^r \nabla_y g(x_i^r, y_i^r; \zeta_i^\pi)$; $u_{i+1}^r = u_i^r - \rho_i^r(\nabla_{y^2} g(x_i^r, y_i^r; \zeta_i^\pi)u_i^r - \nabla_y f(x_i^r, y_i^r; \xi_i^\pi))$;
    $x_{i+1}^r = x_i^r - \eta_i^r(\nabla_x f(x_i^r, y_i^r; \xi_i^\pi) - \nabla_{xy} g(x_i^r, y_i^r; \zeta_i^\pi)u_i^r)$;
  end for
end for

where Eq. (3.4) adopts the single-loop [65] update. More specifically, Eq. (3.4a) performs gradient descent to estimate the minimizer y_x; Eq. (3.4b) performs gradient descent to estimate u_x defined in Eq. (3.3); and Eq. (3.4c) perform the gradient descent step given the estimation of y_x and u_x.

Note that at each step t, we use a pair of examples $\{\xi_t^\pi, \zeta_t^\pi\}$ to estimate the involved properties and require only three backward passes, *i.e.* $\nabla f(x_t, y_t; \xi_t^\pi)$, $\nabla g(x_t, y_t; \zeta_t^\pi)$ and $\nabla(\nabla_y g(x_t, y_t; \zeta_t^\pi))$. In contrast, if we independently sample examples for each property, seven backward passes are needed. For large-scale machine learning models where back-propagation is expensive, our method can lead to significant computation saving in practice.

Equation (3.4) is independent of the specific order of samples. It can accommodate both with-replacement and without-replacement sampling strategies. In particular, we consider two most widely-used without-replacement sampling methods: *random-reshuffling* and *shuffle-once*. For random-reshuffling, we set $T = R \times lcm(m, n)$, where $lcm(m, n)$ denotes the least common multiple of m and n and R is a constant. More specifically, we generate $R \times lcm(m, n)/m$ random permutations of examples in $\mathcal{D}_u$ then concatenate them

to get $\{\xi_t^\pi \in \mathcal{D}_u\}$, and we follow similar procedure to generate $\{\zeta_t^\pi \in \mathcal{D}_u\}$. In algorithm 2, we show our **WiOR-BO** algorithm for solving Eq. (3.2) with random-reshuffling sampling. Note that in Line 4 of Algorithm 2, $\mathcal{P}_\iota(\cdot)$ is the projection operation to an ι-size ball and we perform projection to make u_t be bounded during updates. Shuffle-once is another widely used without-replacement sampling strategy, where a single permutation is generated and reused for both $\mathcal{D}_u$ and $\mathcal{D}_l$, and we can modify Line 3 of Algorithm 2 to incorporate shuffle-once. Finally, compared to independent sampling-based algorithm such as stocBiO [12], we require extra space to generate and store the orders of examples, but this cost is negligible compared to memory and computational cost of training large scale models.

3.3.1 Conditional Bilevel Optimization

In Eq. (3.2), we assume the outer dataset $\mathcal{D}_u$ and the inner dataset $\mathcal{D}_l$ are independent with each other. However, it is also common in practice that the inner problem not only relies on the outer variable x, but also the current outer example. This type of problems is known as conditional (contextual) bilevel optimization problems [59] in the literature and we consider the finite setting as follows:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq f(x, y_x) = \frac{1}{m}\sum_{i=1}^{m} f(x, y_x^{(\xi_i)}, \xi_i), \tag{3.5}$$

$$\text{s.t. } y_x^{(\xi_i)} = \arg\min_{y\in\mathbb{R}^d} g^{(\xi_i)}(x, y) = \frac{1}{n_{\xi_i}}\sum_{j=1}^{n_{\xi_i}} g(x, y; \zeta_{\xi_i,j}), \tag{3.6}$$

Note that in Eq. (3.5), the outer problem is defined on a finite dataset $\mathcal{D}_u = \{\xi_i, i \in [m]\}$, and for each example $\xi_i \in \mathcal{D}_u$, we have a inner problem $g^{(\xi_i)}(x, y)$ defined on a dataset

$\mathcal{D}_{l,\xi_i} = \{\zeta_{\xi_i,j}, j \in [n_{\xi_i}]\}$. The hyper-gradient of Eq. (3.5) is:

$$\nabla h(x) = \frac{1}{m}\sum_{i=1}^{m}\left(\nabla_x f(x, y_x^{\xi_i}; \xi_i) - \nabla_{xy} g^{(\xi_i)}(x, y_x^{\xi_i}) u_x^{\xi_i}\right), u_x^{\xi_i} = \nabla_{y^2} g^{(\xi_i)}(x, y_x)^{-1} \nabla_y f(x, y_x; \xi_i), \tag{3.7}$$

The key difference of Eq. (3.7) with Eq. (3.3) is that $y_x^{\xi_i}$ and $u_x^{\xi_i}$ are defined for each example $\xi_i \in \mathcal{D}_u$. To minimize $h(x)$ in Eq. (3.5), we assume a sample order π denoting the following example selection sequence as $\{\xi_t^\pi \in \mathcal{D}_u, t \in [T]\}$, where T denotes the example sequence length and can be arbitrarily large. We then perform the following update steps iteratively:

$$x_{t+1} = x_t - \eta_t(\nabla_x f(x_t, \hat{y}_t; \xi_t^\pi) - \nabla_{xy} g^{\xi_t^\pi}(x_t, \hat{y}_t)\hat{u}_t) \tag{3.8}$$

where $\hat{y}_t$ and $\hat{u}_t$ are estimations of $y_{x_t}^{\xi_t^\pi}$ $u_{x_t}^{\xi_t^\pi}$. We get them by performing the following rules T_l steps over the sample order $\{\zeta_{\xi_t^\pi,t_l}^\pi \in \mathcal{D}_{l,\xi_t^\pi}, t_l \in [T_l]\}$:

$$y_{t_l+1} = y_{t_l} - \gamma_{t_l}\nabla_y g(x_t, y_{t_l}; \zeta_{\xi_t^\pi,t_l}^\pi), u_{t_l+1} = u_{t_l} - \rho_{t_l}(\nabla_{y^2} g(x_t, y_{t_l}, \zeta_{\xi_t^\pi,t_l}^\pi)u_{t_l} - \nabla_y f(x_t, y_{t_l}; \xi_t^\pi)) \tag{3.9}$$

where both the outer variable state x_t and sample ξ_t^π are fixed. We then set $\hat{y}_t = y_{T_l}$ and $\hat{u}_t = u_{T_l}$ in Eq. (3.8). Note that Eq. (3.8) and Eq. (3.9) perform a double-loop update rule [59]. Since the inner problem relies on the example of the outer problem, we compute $y_x^{\xi_i}$ and $u_x^{\xi_i}$ for each new state x and sample ξ_i instead of reusing them as in the single loop update of Eq. (3.4).

Note that an important feature of our algorithm is that we select samples based on

Algorithm 3 Without-Replacement Conditional Bilevel Optimization (WiOR-CBO)

Input: Initial state x^0, learning rates $\{\eta_i^r\}, i \in [m], r \in [R]$.
for epochs $r = 1$ **to** R **do**
  Randomly sample a permutation of outer dataset to have $\{\xi_i^\pi, i \in [m]\}$ and set $x_0^r = x_m^{r-1}$ or x_0 if $r = 1$.
  for $i = 0$ **to** $m - 1$ **do**
    Initial states y^0 and u^0, learning rates $\{\gamma_j^s, \rho_j^s\}, j \in [n_i], s \in [S]$.
    for $s = 1$ **to** S **do**
      Sample a permutation of inner dataset to have $\{\zeta_{\xi_i,j}^\pi, j \in [n_i]\}$, set $y_0^s = y_{n_i}^{s-1}$ and $u_0^s = \mathcal{P}_\iota(u_{n_i}^{s-1})$ or $y_0^s = y_0$ and $u_0^s = u_0$ if $s = 1$.
      for $j = 0$ **to** $n_i - 1$ **do**
        $y_{j+1}^s = y_j^s - \gamma_j^s \nabla_y g(x_i^r, y_j^s; \zeta_{\xi_i,j}^\pi)$;
        $u_{j+1}^s = u_j^s - \rho_j^s(\nabla_{y^2} g(x_i^r, y_j^s; \zeta_{\xi_i,j}^\pi)u_j^s - \nabla_y f(x_i^r, y_j^s; \xi_i^\pi))$;
      end for
    end for
    $x_{i+1}^r = x_i^r - \eta_i^r(\nabla_x f(x_i^r, y_{n_i}^S; \xi_i^\pi) - \nabla_{xy} g^{\xi_i^\pi}(x_i^r, y_{n_i}^S)u_{n_i}^S)$;
  end for
end for

an example order π at each step instead of sampling independently. Similar to the unconditional bilevel optimization, various example orders can be incorporated. For random-reshuffling, we set $T = Rm$ with R be a constant. Then we generate R random permutations of examples in $\mathcal{D}_u$ and concatenate them to get the example order $\{\xi_t^\pi \in \mathcal{D}_u, t \in [T]\}$. Similarly, we concatenate S random permutations of the inner dataset $\mathcal{D}_{l,\xi_t}$ to get $\{\zeta_{\xi_t^\pi,t_l}^\pi \in \mathcal{D}_{l,\xi_t^\pi}, t_l \in [T_l]\}\}$ with a sequence length $T_l = S \times n_{\xi_i}$. We summarize this in Algorithm 3 and denote it as **WiOR-CBO**. We slightly abuse the notation to replace n_{ξ_i} with n_i in the inner loop update for better clarity.

3.3.2 MiniMax and Compositional Optimization Problems

Our discussion of without-replacement sampling for bilevel optimization can be customized for two important nested optimization problems: *compositional optimization* and *minimax optimization* problems, as they can be viewed as a special type of bilevel opti-

mization problem. Suppose we let the outer problem only rely on y, set the inner problem as $g(x,y) = \frac{1}{2}\|y - r(x)\|^2$. Consider the finite case of $f(y)$ and $r(x)$, we have the following compositional optimization problem:

$$\min_{x\in\mathbb{R}^p} h(x) := \frac{1}{m}\sum_{i=1}^{m} f(r(x);\xi_i) = \frac{1}{m}\sum_{i=1}^{m} f(\frac{1}{n}\sum_{j=1}^{n} r(x;\zeta_j);\xi_i) \tag{3.10}$$

Then we can derive the hyper-gradient as: $\nabla h(x) = \nabla_x r(x) u_x$, where $u_x = \nabla_y f(r(x))$, meanwhile, we have $\nabla_y g(x,y) = y - r(x)$. Then we can instantiate Algorithm 2 to get an algorithm for compositional optimization problem using without-replacement sampling (Algorithm 4 of Appendix). Similar to the SCGD algorithm [80], we track the exponential moving average of the inner state y, besides we also track the exponential moving average of inner gradient (the update rule of u in Algorithm 4).

Next, similar to the unconditional case, the conditional compositional optimization problem can be specialized from the conditional bilevel optimization problem Eq. (3.5) to have:

$$\min_{x\in\mathbb{R}^p} h(x) := \frac{1}{m}\sum_{i=1}^{m} f(\frac{1}{n_i}\sum_{j=1}^{n_i} r(x;\zeta_{\xi_i,j});\xi_i) \tag{3.11}$$

and we can instantiate Algorithm 3 in the special case of Eq. (3.11) to get Algorithm 6 (in the Appendix).

The minimax optimization can also be viewed as a special type of bilevel optimization. More specifically, if we have $g(x,y) = -f(x,y)$ in Eq. (3.2), then we get the following

minimax problem:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq \frac{1}{m}\sum_{i=1}^{m}\max_{y\in\mathbb{R}^d} f(x,y;\xi_i) \tag{3.12}$$

In the special case of Eq. (3.12), the hyper-gradient $\nabla h(x)$ in Eq. (3.3) degenerates to $\nabla h(x) = \nabla_x f(x, y_x)$ due to the fact that $\nabla_y f(x, y_x) = -\nabla_y g(x, y_x) = 0$ by the optimality condition. Meanwhile, we have $\nabla_y g(x, y) = -\nabla_y f(x, y)$. As a result, we can simplify Algorithm 2 to get an alternative update algorithm for minimax optimization using without-replacement sampling (Algorithm 5 in the Appendix). Note that due to the linearity of minimax problem *w.r.t.* examples, the conditional case for minimax optimization degenerates to the unconditional case. [81, 82] also considers the without-replacement sampling for the minimax problem, however, they focus on the strongly-convex-strongly-concave and two-sided Polyak-Łojasiewicz settings, in contrast, our WiOR-MiniMax can be applied to nonconvex-strongly-concave setting.

3.4 Theoretical Analysis

In this section, we provide theoretical analysis of our algorithms. We first state some assumptions:

Assumption 26. *Function $f(x, y; \xi), \xi \in \mathcal{D}_u$, is possibly non-convex, L-smooth and has C_f-bounded gradient* w.r.t. *the variable y.*

Assumption 27. *Function $g(x, y; \zeta), \zeta \in \mathcal{D}_l$, is μ-strongly convex* w.r.t *y for any given x, and L-smooth. Furthermore, $\nabla_{xy} g(x, y; \zeta)$ and $\nabla_{y^2} g(x, y; \zeta)$ are Lipschitz continuous with constants L_{xy} and L_{y^2} respectively.*

Assumption 28. *For all examples $\xi_i \in D_u$ and states $(x, y) \in \mathbb{R}^{p+d}$, there exists an upper bound A such that the sample gradient errors satisfy $\|\nabla f(x, y; \xi_i) - \nabla f(x, y)\| \leq A$.*

Assumption 29. *For all examples $\zeta_j \in D_l$ and states $(x, y) \in \mathbb{R}^{p+d}$, there exists an upper bound A such that the sample gradient errors satisfy $\|\nabla g(x, y; \zeta_i) - \nabla g(x, y)\| \leq A$ and $\|\nabla^2 g(x, y; \zeta_i) - \nabla^2 g(x, y)\| \leq A$.*

As stated in The assumption 26 and 27, we consider the *non-convex-strongly-convex* bilevel optimization problems, this class of problems is widely studied in bilevel optimization literature [6, 12].Assumption 28 and 29 guarantee that the example gradients are good estimation of full gradient and is widely used in the analysis of single-level finite-sum optimization problems [76]. Note that as the hyper-gradient (Eq. 3.3) involves second order derivatives, Assumption 29 also requires that the second order example gradients have bounded bias.

We next make the following assumptions about the example selection sequence. We assume $\{\xi_t^\pi \in \mathcal{D}_u, t \in [T]\}$ and $\{\zeta_t^\pi \in \mathcal{D}_l, t \in [T]\}$ satisfy:

Assumption 30. *Given sequences of samples: $\{\xi_t^\pi \in \mathcal{D}_u, t \in [T]\}$ and $\{\zeta_t^\pi \in \mathcal{D}_l, t \in [T]\}$, we assume there exists some constants α, C such that for any step t and any interval $k > 0$, we have:*

$$\Big\|\frac{1}{k}\sum_{\tau=t}^{t+k-1} \nabla f(x_t, y_t; \xi_\tau^\pi) - \nabla f(x_t, y_t)\Big\|^2 \leq k^{-\alpha} C^2$$

and similar bounds hold for ∇g and $\nabla^2 g$ (See Appendix 3.6.1 for the full version of this assumption.)

Assumption 30 measures how well the sample gradients approximate the full gradient over an interval on average. Similar assumptions are used to study the sample order in single level optimization problems [63]. If samples are selected independently at each step, we get $\alpha = 1$ and $C = A$ based on Assumption 28 and 29. For any permutation-based order (*e.g.* random-reshuffling and shuffle-once), we can show the assumption holds with $\alpha = 2$ and $C = \max(m, n) \times A$. Note that the C depends on the number of samples m (n), this dependence can be reduced to $(\max(m, n))^{0.5}$ for sufficiently small learning rates [63, 83]. More recently, [1] proposed a herding-based algorithm which explicitly optimizes the gradient error in Assumption 30 and shows that we can reach $\alpha = 2$ and $C = O(A)$.

Bilevel Optimization. We are ready to show the convergence of Algorithm 2. Firstly, we denote the following potential function $\mathcal{G}_t := \mathcal{G}_t = h(x_t) + \phi_y \|y_t - y_{x_t}\|^2 + \phi_u \|u_t - u_{x_t}\|^2$, where ϕ_y and ϕ_u are constants that relates to the smoothness parameters of $h(x)$. Then we have:

Theorem 31. *Suppose Assumptions 26-29 are satisfied, and Assumption 30 is satisfied with some α and C for the example order we use in Algorithm 2. We choose learning rates $\eta_t = \eta$, $\gamma_t = c_1\eta$, $\rho_t = c_2\eta$ and denote $T = R \times I$ be the total number of steps. Then for any pair of values (E, k) which has $T = E \times k$ and $\eta \leq \frac{1}{k\tilde{L}_0}$, we have:*

$$\frac{1}{E}\sum_{e=0}^{E-1} \|\nabla h(x_{ke})\|^2 \leq \frac{2\Delta}{kE\eta} + 2\tilde{L}^2 k^{-\alpha} C^2$$

where c_1, c_2, $\tilde{L}_0$, $\tilde{L}$ are constants related to the smoothness parameters of $h(x)$, Δ is the initial sub-optimality, R and I are defined in Algorithm 2.

To reach an ϵ stationary point, we choose $\eta = \frac{1}{k\tilde{L}_0}$, and for (E, k), we choose: $E = \frac{4\tilde{L}_0\Delta}{\epsilon^2}$ and $k = \left(\frac{4\tilde{L}^2C^2}{\epsilon^2}\right)^{1/\alpha}$, which means we need to choose the number of epochs $R = \frac{T}{I} = \frac{4\tilde{L}_0(4\tilde{L}^2C^2)^{1/\alpha}\Delta}{I\epsilon^{2+2/\alpha}}$. Specially, if we choose examples independently at each step, we have $C = A$ and $\alpha = 1$, then we need $T = O(\epsilon^{-4})$ steps to reach an ϵ-stationary point. This recovers the rate of stocBiO algorithm [12] and SOBA [58]. Next, if we use some permutation-based without-replacement sampling strategy witch has $C = (\max(m, n))^q \times A$ $(q \in [0, 1])$ and $\alpha = 2$, then we need $T = O((\max(m, n))^q\epsilon^{-3})$. In particular, by adopting a herding-based permutation as in [1], we get $q = 0$, and our WiOR-BO strictly outperforms the independent sampling-based stocBiO and SOBA algorithms.

Conditional Bilevel Optimization. Next, we study the convergence rate of Algorithm 3. Besides Assumptions 28 and 29, we need the bias of sample hyper-gradient be bounded too:

Assumption 32. *For all examples $\xi_i \in D_u$ and states $x \in \mathbb{R}^p$, there exists an upper bound A such that the sample gradient errors satisfy $\|\nabla h(x; \xi_i) - \nabla h(x)\| \leq A$.*

We also augment Assumption 30 to include the case of $\nabla h(x)$ (Seem Appendix 3.6.1 for more details.). Then we are ready to show the convergence rate, we use the potential function $\mathcal{G}_t = h(x_t)$ and have:

Theorem 33. *Suppose Assumptions 26-29 and 32 are satisfied, and Assumption 30 is satisfied with some α and C for the example order we use in Algorithm 3. We choose learning rates $\eta_t = \eta = \frac{1}{8k\tilde{L}}$, $\gamma_t = \gamma = \frac{1}{256kL\kappa}$, $\rho_t = \rho = \frac{1}{512kL\kappa}$ and denote $T = R \times m$ be the total number of outer steps and $T_l = S \times \max(n_i)$ be the maximum inner steps. Then*

for any pair of values (E, k) which has $T = E \times k$ *and* $T_l = E_l \times k$*, we have:*

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{8\Delta_h}{kE\eta} + C_u(1-\frac{k\mu\rho}{2})^{E_l}\Delta_u + C_y(1-\frac{k\mu\gamma}{2})^{E_l}\Delta_y + (C')^2C^2k^{-\alpha}$$

where C_u*,* C_y*,* C'*,* $\tilde{L}$ *are constants related to the smoothness parameters of* $h(x)$*,* Δ_h*,* Δ_u*,* Δ_y *are the initial sub-optimality,* R *and* S *are defined in Algorithm 3.*

To reach an ϵ stationary point, we choose: $E = \frac{128\bar{L}\Delta}{\epsilon^2}$, and , $k = \left(\frac{2(C')^2C^2}{\epsilon^2}\right)^{1/\alpha}$ and $E_l = O(log(\epsilon^{-1}))$. Then the sample complexity of Algorithm 3 is $EE_lk^2 = O(C^{4/\alpha}\epsilon^{-(2+4/\alpha)})$, where we omit the logarithmic term E_l. Specially, if we choose examples independently at each step, then we have sample complexity $O(\epsilon^{-6})$ steps for an ϵ-stationary point. This recovers the rate of DL-SGD algorithm [59]. Next, if we use some permutation-based without-replacement sampling strategy, then we have sample complexity $O((\max(m,n))^{2q}\epsilon^{-4})$, which can match the state-of-art algorithm RT-MLMC [59] if we choose an example order with $q = 0$ [1].

MiniMax and Compositional Optimization. As special cases of Bilevel Optimization, we can also show that Algorithm 4, Algorithm 5 and Algorithm 6 converge under similar rate. Please see Corollary 54 and Corollary 53 for more details.

3.5 Numerical Experiments

In this section, we verify the effectiveness of the proposed algorithm through a synthetic invariant risk minimization task and two real world bilevel tasks: Hyper-Data Cleaning and Hyper-representation Learning. The code is written in Pytorch. Our experiments

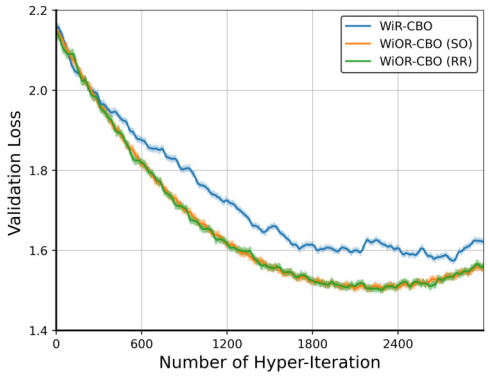


Figure 3.1: Comparison of different sampling strategies for the Invariant Risk-Minimization task.

were conducted on servers equipped with 8 NVIDIA A5000 GPUs. Experiments are averaged over five independent runs, and the standard deviation is indicated by the shaded area in the plots.

3.5.1 Invariant Risk-Minimization

In this section, we consider a synthetic invariant risk minimization task [84]. More specifically, we perform logistic regression over a set of examples $\{(c_i, b_i), i \in [M]\}$ with c be the input and b be the target. However, we does not have access to c but a set of noisy observations $\{c_{j,i}, j \in n_i\}$ for each c_i. To learn the coefficient x between c and b, we optimize the following objective:

$$\min_{x\in\mathbb{R}^p} \frac{1}{m}\sum_{i=1}^{m} \log\Big(1+\exp\Big(-b_i y_x^{(i)}\Big)\Big), \text{ s.t. } y_x^{(i)} = \arg\min_{y\in\mathbb{R}^d} \frac{1}{n_i}\sum_{j=1}^{n_i}\frac{1}{2}\|y - c_{j,i}x\|^2$$

This is a conditional bilevel optimization problem and we can solve it using our WiOR-CBO. We generate synthetic data to test the effects of different sample order (sample generation process is described in Appendix 3.7). More specifically, we compare independent-sampling

(WiR-CBO), shuffle-once (WiOR-CBO (SO)) and random-reshuffling (WiOR-CBO (RR)). As shown in Figure 3.1, the two without-replacement sampling methods outperforms the independent sampling one.

3.5.2 Hyper-Data Cleaning

In the Hyper-Data Cleaning task, we are given noisy training dataset whose labels are corrupted by noise, meanwhile, we have a validation set whose labels are not corrupted. Our target is then using the clean validation samples to identify corrupted samples in the training set. More specifically, we learn optimal weights for training samples such that a model learned over the weighted training set performs well on the validation set. We can formulate this task as a bilevel optimization problem (Eq. (3.2)). A mathematical formulation of the task is included in Appendix 3.7.

Dataset and Baselines. We construct datasets based on MNIST [85]. For the training set, we randomly sample 40000 images from the original training dataset and then randomly perturb a fraction of labels of samples. For the validation set, we randomly select 5000 clean images from the original training dataset. In our experiments, we test our WiOR-BO algorithm (Algorithm 2), with the shuffle-once (WiOR-BO-SO) and random-reshuffling (WiOR-BO-RR) sampling; additionally, we also consider the following non-adaptive bilevel algorithms as baselines: reverse [26], stocBiO [12], BSA [6], AID-CG [7], MRBO [32], VRBO [32] and WiR-BO (the variant of our WiOR-BO using independent sampling, which is similar to the single loop algorithms such as SOBA [58], AmIGO [86] and FLSA [65]). We perform grid search for hyper-parameters and report the best results.

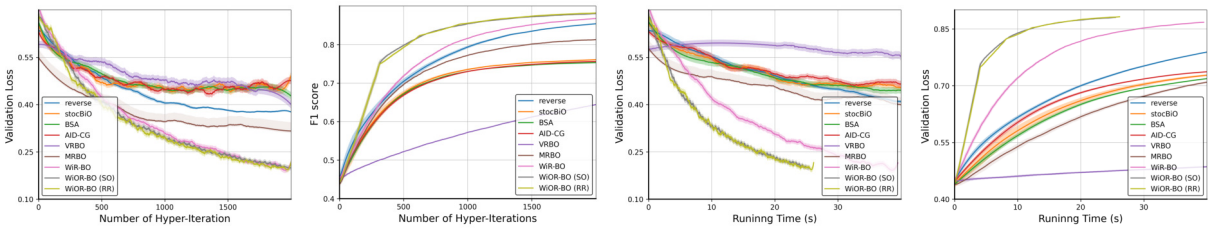


Figure 3.2: **Comparison of different algorithms for the Hyper-data Cleaning task.** The left two plots show validation error/F1 score vs Number of Hyper-Iterations and the right two plots show validation error/F1 score vs Running Time. The fraction of the noisy samples is 0.6.

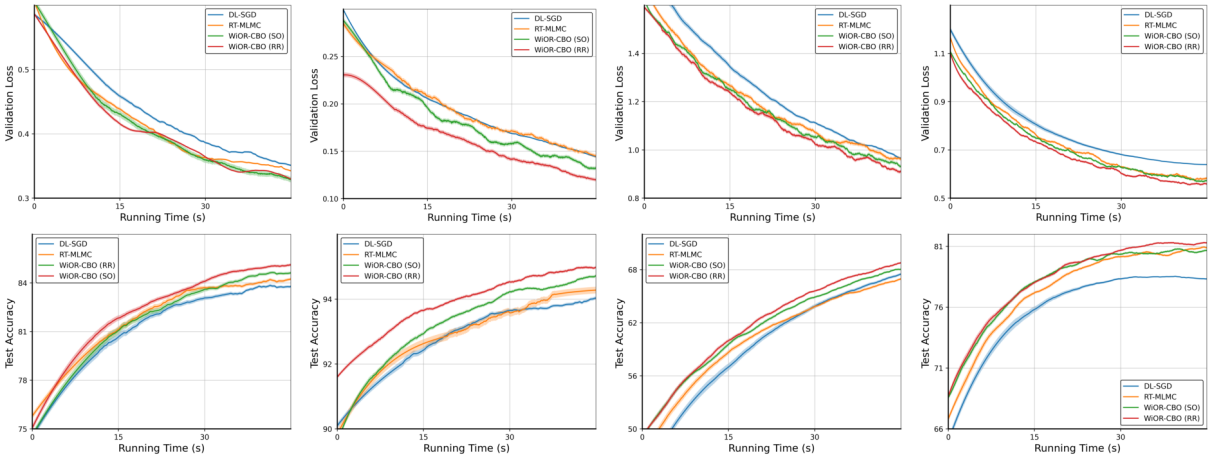


Figure 3.3: **Comparison of different algorithms for the Hyper-Representation Task over the Omniglot Dataset.** From Left to Right: 5-way-1-shot, 5-way-5-shot, 20-way-1-shot, 20-way-5-shot.

We summarize the results in Figure 3.2. In the figure, we compare the validation loss and F1 score *w.r.t.* both Hyper-iterations and running time. As shown in the figure, the two variants of our algorithm WiOR-BO-SO and WiOR-BO-RR get similar performance and they both outperform other baselines. In particular, note that WiR-BO differs with WiOR-BO-SO (RR) only over the example order, but the later converges much faster in terms of running time, this is partially due to less umber of backward passes needed by without-replacement sampling per iteration.

3.5.3 Hyper-Representation Learning

In this section, we consider Hyper-representation learning task. In this task, we learn a hyper-representation of the data such that a linear classifier can be learned quickly with a small number of samples. We consider the Omniglot [87] and MiniImageNet [88] data sets. This task can be viewed as a conditional bilevel optimization problem (Eq. (3)) and a mathematical formulation of the task is included in Appendix 3.7.

Dataset and Baselines. The details of the datasets are included in Appendix 3.7 and we consider N-way-K-shot classification task following [89]. In our experiments, we test our WiOR-CBO (Algorithm 3), using the shuffle-once (WiOR-BO-SO) and random-reshuffling (WiOR-BO-RR) sampling. Besides, we compare with the following baselines: DL-SGD [59] and RT-MLMC [59]. Note that our WiOR-CBO can also use independent sampling, and it has similar performance as DL-SGD. We perform grid search of hyper-parameters for each method and report the best results.

We summarize the experimental results for the Omniglot dataset in Figure 3.3 and we defer the results for MiniImageNet to the Appendix 3.7. As shown in the figure, our WiOR-CBO SO/RR outperforms DL-SGD and is comparable to the state-of-the-art algorithm RT-MLMC.

3.6 Proof for Theorems

In this section, we provide proof for the convergence results in the main text. First, we restate all assumptions needed in our proof below:

Assumption 34 (Assumption 4.1)**.** *The function $f(x, y)$ is possibly non-convex and uniformly L-smooth,* i.e. *for any x_1, $x_2 \in \mathcal{X}$, y_1, $y_2 \in \mathbb{R}^d$, any $\xi \in \mathcal{D}_u$. Denote $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$, then we have:*

$$f(z_1; \xi) \leq f(z_2; \xi) + \langle \nabla f(z_2, \xi), z_1 - z_2 \rangle + \frac{L}{2}||z_1 - z_2||^2.$$

or equivalently: $||\nabla f(z_1; \xi) - \nabla f(z_2; \xi)|| \leq L||z_1 - z_2||$. We also assume for any $x \in \mathcal{X}$ and any $y \in \mathbb{R}^d$, and we denote $z = (x, y)$, then we have $||\nabla_y f(z)|| \leq C_f$.

Assumption 35 (Assumption 4.2)**.** *Function $g(x, y)$ is uniformly μ-strongly convex* w.r.t *y for any given x and uniformly L-smooth,* i.e. *for any y_1, $y_2 \in \mathbb{R}^d$ and $\zeta \in \mathcal{D}_l$, we have:*

$$g(x, y_1; \zeta) \geq g(x, y_2; \zeta) + \langle \nabla_y g(x, y_2; \zeta), y_2 - y_1 \rangle + \frac{\mu}{2}||y_2 - y_1||^2.$$

and we have:

$$g(z_1; \zeta) \leq g(z_2; \zeta) + \langle \nabla g(z_2; \zeta), z_1 - z_2 \rangle + \frac{L}{2}||z_1 - z_2||^2.$$

equivalently: $||\nabla g(z_1; \zeta) - \nabla g(z_2; \zeta)|| \leq L||z_1 - z_2||$. Furthermore, we have $\nabla_{xy} g(x, y)$ and $\nabla_{y^2} g(x, y)$ are Lipschitz continuous with constant L_{xy} and L_{y^2} respectively, i.e. *we have: $||\nabla_{xy} g(z_1; \zeta) - \nabla_{xy} g(z_2; \zeta)|| \leq L_{xy}||z_1 - z_2||$ and $||\nabla_{y^2} g(z_1; \zeta) - \nabla_{y^2} g(z_2; \zeta)|| \leq L_{y^2}||z_1 - z_2||$.*

Assumption 36 (Assumption 4.3)**.** *For all examples $\xi_i \in D_u$ and states $(x, y) \in \mathbb{R}^{p+d}$, there exists an outer bound A such that the sample gradient errors satisfy $\|\nabla f(x, y; \xi_i) - \nabla f(x, y)\| \leq A$.*

Algorithm 4 Without-Replacement Compositional Optimization (WiOR-Comp)

Input: Initial states x_I^0, y_I^0 and u_I^0; learning rates $\{\gamma_i^r, \rho_i^r, \eta_i^r\}, i \in [I], r \in [R]$, $I = \text{lcm}(m, n)$.
for epochs $r = 1$ **to** R **do**
  Generate sample order sequence $\{\xi_i^\pi\}$ and $\{\zeta_i^\pi\}$ for $i \in [I]$ as Line 3 of Algorithm 2;
  Set $y_0^r = y_I^{r-1}$, $u_0^r = \mathcal{P}_\iota(u_I^{r-1})$ and $x_0^r = x_I^{r-1}$
  for $i = 0$ **to** $I - 1$ **do**
    $y_{i+1}^r = (1 - \gamma_i^r)y_i^r + \gamma_i^r r(x_i^r, \zeta_i^\pi)$; $u_{i+1}^r = (1 - \rho_i^r)u_i^r + \rho_i^r \nabla_y f(y_i^r; \xi_i^\pi)$;
    $x_{i+1}^r = x_i^r - \eta_i^r \nabla r(x_i^r; \zeta_i^\pi)u_i^r$;
  end for
end for

Assumption 37 (Assumption 4.4)**.** *For all examples $\zeta_j \in D_l$ and states $(x, y) \in \mathbb{R}^{p+d}$, there exists an outer bound A such that the sample gradient errors satisfy $\|\nabla g(x, y; \zeta_i) - \nabla g(x, y)\| \leq A$ and $\|\nabla^2 g(x, y; \zeta_i) - \nabla^2 g(x, y)\| \leq A$.*

Assumption 38 (Assumption 4.7)**.** *For all examples $\xi_i \in D_u$ and states $x \in \mathbb{R}^p$, there exists an upper bound A such that the sample gradient errors satisfy $\|\nabla h(x; \xi_i) - \nabla h(x)\| \leq A$.*

Assumption 39. *Given two sequences $\{\xi_t^\pi \in \mathcal{D}_u, t \in [T]\}$ and $\{\zeta_t^\pi \in \mathcal{D}_l, t \in [T]\}$, we assume there exists some constants α, C such that for any step t and any $k > 0$, we have: $\left\|\frac{1}{k}\sum_{\tau=t}^{t+k-1} \nabla f(x_t, y_t; \xi_\tau^\pi) - \nabla f(x_t, y_t)\right\|^2 \leq k^{-\alpha}C^2$, $\left\|\frac{1}{k}\sum_{\tau=t}^{t+k-1} \nabla g(x_t, y_t; \zeta_\tau^\pi) - \nabla g(x_t, y_t)\right\|^2 \leq k^{-\alpha}C^2$, $\left\|\frac{1}{k}\sum_{\tau=t}^{t+k-1} \nabla^2 g(x_t, y_t; \zeta_\tau^\pi) - \nabla^2 g(x_t, y_t)\right\|^2 \leq k^{-\alpha}C^2$, $\left\|\frac{1}{k}\sum_{\tau=t}^{t+k-1} \nabla h(x_t; \xi_\tau^\pi) - \nabla h(x_t)\right\|^2 \leq k^{-\alpha}C^2$ (Only for the conditional bilevel optimization case).*

3.6.1 Proof for Bilevel Optimization Problems

Suppose we define:

$$\Phi(x, y) = \nabla_x f(x, y) - \nabla_{xy} g(x, y) \times [\nabla_{y^2} g(x, y)]^{-1} \nabla_y f(x, y)$$

Algorithm 5 Without-Replacement MiniMax Opt. (WiOR-MiniMax)

Input: Initial states x_m^0, y_m^0; learning rates $\{\gamma_i^r, \eta_i^r\}, i \in [m], r \in [R]$.
for epochs $r = 1$ **to** R **do**
  sample a permutation of outer dataset to have $\{\xi_i^\pi, i \in [m]\}$;
  Set $y_0^r = y_m^{r-1}$ and $x_0^r = x_m^{r-1}$
  for $i = 0$ **to** $m - 1$ **do**
    $y_{i+1}^r = y_i^r + \gamma_i^r \nabla_y f(x_i^r, y_i^r; \xi_i^\pi)$;
    $x_{i+1}^r = x_i^r - \eta_i^r \nabla_x f(x_i^r, y_i^r; \xi_i^\pi)$;
  end for
end for

Algorithm 6 Without-Replacement Conditional Compositional Optimization (WiOR-CComp)

Input: Initial state x^0, learning rates $\{\eta_i^r\}, i \in [m], r \in [R]$.
for epochs $r = 1$ **to** R **do**
  Randomly sample a permutation of outer dataset to have $\{\xi_i^\pi, i \in [m]\}$ and set $x_0^r = x_m^{r-1}$ or x_0 if $r = 1$.
  for $i = 0$ **to** $m - 1$ **do**
    Initial states y^0 and u^0, learning rates $\{\gamma_j^s, \rho_j^s\}, j \in [n_i], s \in [S]$.
    for $s = 1$ **to** S **do**
      Sample a permutation of inner dataset to have $\{\zeta_{\xi_i,j}^\pi, i \in [n_i]\}$, set $y_0^s = y_{n_i}^{s-1}$ and $u_0^s = \mathcal{P}_\epsilon(u_{n_i}^{s-1})$ or $y_0^s = y_0$ and $u_0^s = u_0$ if $s = 1$.
      for $j = 0$ **to** $n_i - 1$ **do**
        $y_{j+1}^s = (1 - \gamma_j^s) y_j^s + \gamma_j^s r(x_i^r; \zeta_{\xi_i,j}^\pi)$;
        $u_{j+1}^s = (1 - \rho_j^s) u_j^s + \rho_j^s \nabla f(y_j^s; \xi_i^\pi)$;
      end for
    end for
    $x_{i+1}^r = x_i^r - \eta_i^r \nabla r^{\xi_i^\pi}(x_i^r) u_{n_i}^S$;
  end for
end for

Then we have $\nabla h(x) = \Phi(x, y_x)$. Furthermore, we have the following proposition:

Proposition 40. *Suppose Assumptions 26 and 27 hold, the following statements hold:*

a) y_x *is Lipschitz continuous in* x *with constant* $\rho = \kappa$, *where* $\kappa = \frac{L}{\mu}$ *is the condition number of* $g(x, y)$.

b) $\|\Phi(x_1; y_1) - \Phi(x_2; y_2)\|^2 \leq \hat{L}^2(\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2)$, *where* $\hat{L} = O(\kappa^2)$.

c) $h(x)$ *is smooth in* x *with constant* $\bar{L}$ *i.e., for any given* $x_1, x_2 \in X$, *we have* $\|\nabla h(x_2) -$

$\nabla h(x_1)\| \leq \bar{L}\|x_2 - x_1\|$ *where* $\bar{L} = O(\kappa^3)$.

This is a standard results in bilevel optimization and we omit the proof here.

The main technique we use to analysis the convergence of Algorithm 2 is *aggregated analysis*, in other words, we perform analysis over a block of length k instead of the more common per step analysis in stochastic bilevel optimization. Note that this technique is essential in analyzing without-replacement sampling for single-level optimization problems, and we extend them to the bilevel setting. More specifically, suppose the total number of training steps are $T := R \times I$, where R and I are denoted in Algorithm 2, then we consider a block of training steps $[t, t+k]$ for $t \in [T-k]$. Furthermore, we denote: $\bar{\eta}_t = \sum_{\tau=t}^{t+k-1} \eta_\tau$, and $\bar{\nu}_t = \frac{1}{\bar{\eta}_t}\sum_{\tau=t}^{t+k-1} \eta_\tau \nu_\tau$; $\bar{\gamma}_t = \sum_{\tau=t}^{t+k-1} \gamma_\tau$ and $\bar{w}_t = \frac{1}{\bar{\gamma}_t}\sum_{\tau=t}^{t+k-1} \gamma_\tau w_\tau$; $\bar{\rho}_t = \sum_{\tau=t}^{t+k-1} \rho_\tau$, and $\bar{q}_t = \frac{1}{\bar{\rho}_t}\sum_{\tau=t}^{t+k-1} \rho_\tau q_\tau$, where $\nu_t = \nabla_x f(x_t, y_t; \xi_t) - \nabla_{xy} g(x_t, y_t; ; \zeta_t) u_t$, $w_t = \nabla_y g(x_t, y_t; \zeta_t)$ and $q_t = \nabla_{y^2} g(x_t, y_t; \zeta_t) u_t - \nabla_y f(x_t, y_t; \xi_t)$. In particular, we denote $r_x(u)$ as:

$$r_x(u) = \frac{1}{2} u^T \nabla_{y^2} g(x, y_x) u - \nabla_y f(x, y_x)^T u,$$

by assumption it is L smooth and μ strongly convex. It is straightforward to see that $u_x = \arg\min_{u \in \mathbb{R}^d} r_x(u)$, where u_x is defined in Eq. (3.3). Finally, in the proof, we consider the general case of different orders for all involved example derivatives. In other words, we have $\{\xi_{t,1}\}$, $\{\xi_{t,2}\}$, $\{\zeta_{t,1}\}$, $\{\zeta_{t,2}\}$ and $\{\zeta_{t,3}\}$ for $\nabla_y f(x,y)$, $\nabla_x f(x,y)$,$\nabla_y g(x,y)$, $\nabla_{y^2} g(x,y)$, $\nabla_{xy} g(x,y)$ respectively. The two orders used by Algorithm 2 is a special case of the proof and is practically appealing as they can reuse computation as we discussed in the introduction.

Lemma 41. *For* $\tau > t \geq 0$, *the bias of* $\| \sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \big(\nu_{\bar{\tau}} - \nabla h(x_t)\big)\|^2$, $\Big\| \sum_{\bar{\tau}=t}^{\tau-1} \rho_{\bar{\tau}} \big(\nabla r_{x_t}(u_t) - q_{\bar{\tau}}\big)\Big\|^2$ *and* $\| \sum_{\bar{\tau}=t}^{\tau-1} \gamma_{\bar{\tau}} \big(\nabla_y g(x_{\bar{\tau}}, y_{\bar{\tau}}, \zeta_{\bar{\tau},1}) - \nabla_y g(x_t, y_t)\big)\|^2$ *are bounded by Eq.* (3.16), *Eq.* (3.17)

and Eq. (3.18).

Proof. First for $\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}(\nu_{\bar{\tau}}-\nabla h(x_t))\|^2$, we have:

$$
\begin{aligned}
&\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla h(x_t)-\nu_{\bar{\tau}}\Big)\Big\|^2\\
&=\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_{x_t})-\nabla_{xy}g(x_t,y_{x_t})u_{x_t}-(\nabla_x f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},2})-\nabla_{xy}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},3})u_{\bar{\tau}})\Big)\Big\|^2\\
&\le 2\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_{x_t})-\nabla_x f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},2})\Big)\|^2\\
&\quad+2\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_{xy}g(x_t,y_{x_t})u_{x_t}-\nabla_{xy}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},3})u_{\bar{\tau}}\Big)\|^2\\
&\le 2\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_{x_t})-\nabla_x f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},2})\Big)\|^2\\
&\quad+4\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\nabla_{xy}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},3})\Big(u_{x_t}-u_{\bar{\tau}}\Big)\|^2\\
&\quad+\frac{4C_f^2}{\mu^2}\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_{xy}g(x_t,y_{x_t})-\nabla_{xy}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},3})\Big)\|^2\\
&\le 2\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_{x_t})-\nabla_x f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},2})\Big)\|^2+4L^2\bar{\eta}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\|(u_{x_t}-u_{\bar{\tau}})\|^2\\
&\quad+\frac{4C_f^2}{\mu^2}\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_{xy}g(x_t,y_{x_t})-\nabla_{xy}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},3})\Big)\|^2 \qquad (3.13)
\end{aligned}
$$

For the first and the third terms above, we have:

$$
\begin{aligned}
&\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_{x_t})-\nabla_x f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},2})\Big)\|^2\\
&\le 3L^2(\bar{\eta}_t)^2\|y_{x_t}-y_t\|^2\\
&\quad+3L^2\bar{\eta}_t\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\|z_t-z_{\bar{\tau}}\|^2+3\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_t)-\nabla_x f(x_t,y_t;\xi_{\bar{\tau},2})\Big)\|^2 \qquad (3.14)
\end{aligned}
$$

and

$$
\begin{aligned}
&\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_{xy}g(x_t,y_{x_t})-\nabla_{xy}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},3})\Big)\|^2\\
&\le 3L_{xy}^2(\bar{\eta}_t^\tau)^2\|y_{x_t}-y_t\|^2+3L_{xy}^2\bar{\eta}_t^\tau\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\|z_t-z_{\bar{\tau}}\|^2\\
&\quad+3\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_{xy}g(x_t,y_t)-\nabla_{xy}g(x_t,y_t;\zeta_{\bar{\tau},3})\Big)\|^2
\end{aligned}
\tag{3.15}
$$

Combine Eq (3.14) and (3.15) with Eq. (3.13) and denote $\tilde{L}_1^2=\Big(L^2+\frac{2L_{xy}^2C_f^2}{\mu^2}\Big)$ to have:

$$
\begin{aligned}
&\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nu_{\bar{\tau}}-\nabla h(x_t)\Big)\|^2\\
&\le 6\tilde{L}_1^2(\bar{\eta}_t^\tau)^2\|y_{x_t}-y_t\|^2+6\tilde{L}_1^2\bar{\eta}_t^\tau\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\|z_t-z_{\bar{\tau}}\|^2+8L^2(\bar{\eta}_t^\tau)^2\|u_{x_t}-u_t\|^2\\
&\quad+8L^2\bar{\eta}_t^\tau\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\|u_t-u_{\bar{\tau}}\|^2+6\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_x f(x_t,y_t)-\nabla_x f(x_t,y_t;\xi_{\bar{\tau},2})\Big)\|^2\\
&\quad+\frac{12C_f^2}{\mu^2}\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla_{xy}g(x_t,y_t)-\nabla_{xy}g(x_t,y_t;\zeta_{\bar{\tau},3})\Big)\|^2
\end{aligned}
\tag{3.16}
$$

Next, for the term $\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}(\nabla r_{x_t}(u_t)-q_{\bar{\tau}})\|^2$, we have:

$$
\begin{aligned}
&\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla r_{x_t}(u_t)-q_{\bar{\tau}}\Big)\Big\|^2\\
&\le\Big\|\sum_{\tau=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_{y^2}g(x_{\bar{\tau}},y_{\bar{\tau}},\zeta_{\bar{\tau},2})u_{\bar{\tau}}-\nabla_y f(x_{\bar{\tau}},y_{\bar{\tau}},\xi_{\bar{\tau},1})-\Big(\nabla_{y^2}g(x_t,y_{x_t})u_t-\nabla_y f(x_t,y_{x_t})\Big)\Big\|^2\\
&\le 2\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_y f(x_t,y_{x_t})-\nabla_y f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},1})\Big)\|^2\\
&\quad+2\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_{y^2}g(x_t,y_{x_t})u_t-\nabla_{y^2}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},2})u_{\bar{\tau}}\Big)\|^2\\
&\le 2\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_y f(x_t,y_{x_t})-\nabla_y f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},1})\Big)\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+ 4\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\nabla_{y^2}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\bar{\tau},2})\Big(u_t-u_{\bar{\tau}}\Big)\|^2\\
&+\frac{4C_f^2}{\mu^2}\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_{y^2}g(x_t,y_{x_t})-\nabla_{y^2}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\tau,2})\Big)\|^2\\
\leq\ & 2\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_y f(x_t,y_{x_t})-\nabla_y f(x_{\bar{\tau}},y_{\bar{\tau}};\xi_{\bar{\tau},1})\Big)\|^2+4L^2\bar{\rho}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\|(u_t-u_{\bar{\tau}})\|^2\\
&+\frac{4C_f^2}{\mu^2}\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_{y^2}g(x_t,y_{x_t})-\nabla_{y^2}g(x_{\bar{\tau}},y_{\bar{\tau}};\zeta_{\tau,2})\Big)\|^2
\end{aligned}
$$

The first and the third terms can be bounded similarly as in Eq. (3.14) and Eq. (3.15), and denote $\tilde{L}_2^2=\Big(L^2+\frac{2L_{y^2}^2C_f^2}{\mu^2}\Big)$, then we have:

$$
\begin{aligned}
&\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla r_{x_t}(u_t)-q_{\bar{\tau}}\Big)\Big\|^2\\
&\leq 6\tilde{L}_2^2(\bar{\rho}_t^{\tau})^2\|y_{x_t}-y_t\|^2+6\tilde{L}_2^2\bar{\rho}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\|z_t-z_{\bar{\tau}}\|^2+4L^2\bar{\rho}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\|u_t-u_{\bar{\tau}}\|^2\\
&\quad+6\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_y f(x_t,y_t)-\nabla_y f(x_t,y_t;\xi_{\bar{\tau},1})\Big)\|^2\\
&\quad+\frac{12C_f^2}{\mu^2}\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(\nabla_{y^2}g(x_t,y_t)-\nabla_{y^2}g(x_t,y_t;\zeta_{\bar{\tau},2})\Big)\|^2
\end{aligned}
\tag{3.17}
$$

Finally, for $\|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}(\nabla_y g(x_{\bar{\tau}},y_{\bar{\tau}},\zeta_{\bar{\tau},1})-\nabla_y g(x_t,y_t))\|^2$, we have:

$$
\begin{aligned}
&\|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\Big(\nabla_y g(x_{\bar{\tau}},y_{\bar{\tau}},\zeta_{\bar{\tau},1})-\nabla_y g(x_t,y_t)\Big)\|^2\\
&\leq 2\|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\Big(\nabla_y g(x_{\bar{\tau}},y_{\bar{\tau}},\zeta_{\bar{\tau},1})-\nabla_y g(x_t,y_t,\zeta_{\bar{\tau},1}))\Big)\|^2\\
&\quad+2\|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\Big(\nabla_y g(x_t,y_t,\zeta_{\bar{\tau},1})-\nabla_y g(x_t,y_t)\Big)\|^2\\
&\leq 2L^2\bar{\gamma}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\|z_{\bar{\tau}}-z_t\|^2+2\|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\Big(\nabla_y g(x_t,y_t)-\nabla_y g(x_t,y_t,\zeta_{\bar{\tau},1})\Big)\|^2
\end{aligned}
\tag{3.18}
$$

This completes the proof. □

Lemma 42. *For all $t \geq 0$ and any constant $m > 0$, suppose $\bar{\eta}_t < \frac{1}{2\bar{L}}$, the iterates generated satisfy:*

$$
\begin{aligned}
h(x_{t+k}) \leq h(x_t) - \frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 - \frac{\bar{\eta}_t}{4}\|\bar{\nu}_t\|^2 + 3\tilde{L}_1^2\bar{\eta}_t\|y_{x_t} - y_t\|^2 + 3\tilde{L}_1^2\sum_{\tau=t}^{t+k-1}\eta_\tau\|z_t - z_\tau\|^2 \\
+ 4L^2\bar{\eta}_t\|u_{x_t} - u_t\|^2 + 4L^2\sum_{\tau=t}^{t+k-1}\eta_\tau\|u_t - u_\tau\|^2 \\
+ \frac{3}{\bar{\eta}_t}\|\sum_{\tau=t}^{t+k-1}\eta_\tau\Big(\nabla_x f(x_t, y_t) - \nabla_x f(x_t, y_t; \xi_{\tau,2})\Big)\|^2 \\
+ \frac{6C_f^2}{\bar{\eta}_t\mu^2}\|\sum_{\tau=t}^{t+k-1}\eta_\tau\Big(\nabla_{xy} g(x_t, y_t) - \nabla_{xy} g(x_t, y_t; \zeta_{\tau,3})\Big)\|^2
\end{aligned}
$$

Proof. By the smoothness of f, we have:

$$
\begin{aligned}
h(x_{t+k}) &\leq h(x_t) + \langle\nabla h(x_t), x_{t+k} - x_t\rangle + \frac{\bar{L}}{2}\|x_{t+k} - x_t\|^2 \\
&= h(x_t) - \bar{\eta}_t\langle\nabla h(x_t), \bar{\nu}_t\rangle + \frac{\bar{\eta}_t^2\bar{L}}{2}\|\bar{\nu}_t\|^2 \\
&\overset{(a)}{=} h(x_t) - \frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 + \frac{\bar{\eta}_t}{2}\|\nabla h(x_t) - \bar{\nu}_t\|^2 - \left(\frac{\bar{\eta}_t}{2} - \frac{\bar{\eta}_t^2\bar{L}}{2}\right)\|\bar{\nu}_t\|^2 \\
&\overset{(b)}{\leq} h(x_t) - \frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 + \frac{\bar{\eta}_t}{2}\|\nabla h(x_t) - \bar{\nu}_t\|^2 - \frac{\bar{\eta}_t}{4}\|\bar{\nu}_t\|^2
\end{aligned} \tag{3.19}
$$

where equality (a) uses $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a - b\|^2]$; (b) follows the assumption that $\bar{\eta}_t < 1/2\bar{L}$. Next, for the third term, by using Eq. (3.16), we have:

$$
\begin{aligned}
\frac{\bar{\eta}_t}{2}\|\nabla h(x_t) - \bar{\nu}_t\|^2 \leq 3\tilde{L}_1^2\bar{\eta}_t\|y_{x_t} - y_t\|^2 + 3\tilde{L}_1^2\sum_{\tau=t}^{t+k-1}\eta_\tau\|z_t - z_\tau\|^2 \\
+ 4L^2\bar{\eta}_t\|u_{x_t} - u_t\|^2 + 4L^2\sum_{\tau=t}^{t+k-1}\eta_\tau\|u_t - u_\tau\|^2 \\
+ \frac{3}{\bar{\eta}_t}\|\sum_{\tau=t}^{t+k-1}\eta_\tau\Big(\nabla_x f(x_t, y_t) - \nabla_x f(x_t, y_t; \xi_{\tau,2})\Big)\|^2
\end{aligned}
$$

$$+\frac{6C_f^2}{\bar{\eta}_t\mu^2}\|\sum_{\tau=t}^{t+k-1}\eta_\tau\Big(\nabla_{xy}g(x_t,y_t)-\nabla_{xy}g(x_t,y_t;\zeta_{\tau,3})\Big)\|^2$$

where $\tilde{L}_1^2=\Big(L^2+\frac{2L_{xy}^2C_f^2}{\mu^2}\Big)$. This completes the proof. □

Lemma 43. *When $\bar{\gamma}_t<\frac{1}{4L}$, the error of the inner variable y_t are bounded through Eq. (3.20).*

Proof. then by proposition 52 and choose $\bar{\gamma}_t<\frac{1}{4L}$, we have:

$$\|y_{t+k}-y_{x_t}\|^2\leq(1-\frac{\mu\bar{\gamma}_t}{2})\|y_t-y_{x_t}\|^2-\frac{\bar{\gamma}_t^2}{2}\|\nabla_y g(x_t,y_t)\|^2+\frac{4\bar{\gamma}_t}{\mu}\|\nabla_y g(x_t,y_t)-\bar{w}_t\|^2$$

Furthermore, by the generalized triangle inequality, we have:

$$\begin{aligned}\|y_{t+k}-y_{x_{t+k}}\|^2&\leq(1+\frac{\mu\bar{\gamma}_t}{4})\|y_{t+k}-y_{x_t}\|^2+(1+\frac{4}{\mu\bar{\gamma}_t})\|y_{x_{t+k}}-y_{x_t}\|^2\\&\leq(1+\frac{\mu\bar{\gamma}_t}{4})\|y_{t+k}-y_{x_t}\|^2+\frac{5\kappa^2}{\mu\bar{\gamma}_t}\|x_t-x_{t+k}\|^2\\&\leq(1-\frac{\mu\bar{\gamma}_t}{4})\|y_t-y_{x_t}\|^2-\frac{\bar{\gamma}_t^2}{2}\|\nabla_y g(x_t,y_t)\|^2+\frac{5\kappa^2}{\mu\bar{\gamma}_t}\|x_t-x_{t+k}\|^2\\&\quad+\frac{10L^2}{\mu}\sum_{\tau=t}^{t+k-1}\gamma_\tau\|z_t-z_\tau\|^2\\&\quad+\frac{10}{\mu\bar{\gamma}_t}\|\sum_{\tau=t}^{t+k-1}\gamma_\tau\Big(\nabla_y g(x_t,y_t)-\nabla_y g(x_t,y_t;\zeta_{\tau,1})\Big)\|^2\end{aligned}\tag{3.20}$$

where the second inequality is due to $\bar{\gamma}_t<\frac{1}{4L}<\frac{1}{\mu}$ and uses Eq. (3.18). This completes the proof. □

Lemma 44. *When $\bar{\rho}_t<\frac{1}{4L}$, the error of variable u_t are bounded through Eq. (3.21).*

Proof. suppose we choose $\bar{\rho}_t<\frac{1}{4L}$, then we have:

$$\|u_{t+k}-u_{x_t}\|^2\leq(1-\frac{\mu\bar{\rho}_t}{2})\|u_t-u_{x_t}\|^2-\frac{(\bar{\rho}_t)^2}{2}\|\nabla r_{x_t}(u_t)\|^2+\frac{4\bar{\rho}_t}{\mu}\|\nabla r_{x_t}(u_t)-\bar{q}_t\|^2$$

Furthermore, by the generalized triangle inequality, we have:

$$\begin{aligned}
\|u_{t+k} - u_{x_{t+k}}\|^2 &\leq (1 + \frac{\mu\bar{\rho}_t}{4})\|u_{t+k} - u_{x_t}\|^2 + (1 + \frac{4}{\mu\bar{\rho}_t})\|u_{x_{t+k}} - u_{x_t}\|^2 \\
&\leq (1 + \frac{\mu\bar{\rho}_t}{4})\|u_{t+k} - u_{x_t}\|^2 + \frac{5\hat{L}^2}{\mu\bar{\rho}_t}\|x_t - x_{t+k}\|^2 \\
&\leq (1 - \frac{\mu\bar{\rho}_t}{4})\|u_t - u_{x_t}\|^2 - \frac{\bar{\rho}_t^2}{2}\|\nabla r_{x_t}(u_t)\|^2 + \frac{5\hat{L}^2}{\mu\bar{\rho}_t}\|x_t - x_{t+k}\|^2 \\
&\quad + \frac{30\tilde{L}_2^2\bar{\rho}_t}{\mu}\|y_{x_t} - y_t\|^2 + \frac{30\tilde{L}_2^2}{\mu}\sum_{\tau=t}^{t+k-1}\rho_\tau\|z_t - z_\tau\|^2 \\
&\quad + \frac{20L^2}{\mu}\sum_{\tau=t}^{t+k-1}\rho_\tau\|u_t - u_\tau\|^2 \\
&\quad + \frac{30}{\mu\bar{\rho}_t}\|\sum_{\tau=t}^{t+k-1}\rho_\tau\Big(\nabla_y f(x_t, y_t) - \nabla_y f(x_t, y_t; \xi_{\tau,1})\Big)\|^2 \\
&\quad + \frac{60C_f^2}{\mu^3\bar{\rho}_t}\|\sum_{\tau=t}^{t+k-1}\rho_\tau\Big(\nabla_{y^2} g(x_t, y_t) - \nabla_{y^2} g(x_t, y_t; \zeta_{\tau,2})\Big)\|^2
\end{aligned} \tag{3.21}$$

where in the last inequality, we use the condition that $\bar{\rho}_t < \frac{1}{4L}$ and Eq. (3.17). This completes the proof. □

Lemma 45. *Suppose* $\bar{\eta}_t \leq \min\Big(\frac{1}{96\tilde{L}_1^2}, \frac{1}{32L^2c_1^2}, \frac{1}{128L^2}, \frac{1}{64L^2c_2^2}, \frac{1}{96\tilde{L}_2^2c_2^2}\Big)$, *then the drift of* $z_\tau = (x_\tau, y_\tau)$ *and* u_τ *can be bounded as in Eq.* (3.26) *and Eq.* (3.25).

Proof. We first have:

$$\begin{aligned}
\|x_t - x_\tau\|^2 &= \|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\nu_{\bar{\tau}}\|^2 \leq 2\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nu_{\bar{\tau}} - \nabla h(x_t)\Big)\|^2 + 2(\bar{\eta}_t^\tau)^2\|\nabla h(x_t)\|^2 \\
\|y_t - y_\tau\|^2 &= \|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\omega_{\bar{\tau}}\|^2 \leq 2\|\sum_{\bar{\tau}=t}^{\tau-1}\gamma_{\bar{\tau}}\Big(\nabla_y g(x_{\bar{\tau}}, y_{\bar{\tau}}, \zeta_{\bar{\tau},1}) - \nabla_y g(x_t, y_t)\Big)\|^2 \\
&\quad + 2(\bar{\gamma}_t^\tau)^2\|\nabla_y g(x_t, y_t)\|^2 \\
\|u_t - u_\tau\|^2 &= \|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}q_{\bar{\tau}}\|^2 \leq 2\|\sum_{\bar{\tau}=t}^{\tau-1}\rho_{\bar{\tau}}\Big(q_{\bar{\tau}} - \nabla r_{x_t}(u_t)\Big)\|^2 + 2(\bar{\rho}_t^\tau)^2\|\nabla r_{x_t}(u_t)\|^2
\end{aligned} \tag{3.22}$$

For the term $\|z_t - z_\tau\|^2$, we have $\|z_t - z_\tau\|^2 = \|x_t - x_\tau\|^2 + \|y_t - y_\tau\|^2$. Combine Eq. (3.22) with Eq. (3.16) and (3.18), sum τ over $[t, t+k-1]$ and use Proposition 51 to have:

$$\begin{aligned}\sum_{\tau=t}^{t+k-1} \eta_\tau \|z_t - z_\tau\|^2 \leq & \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(\sum_{\bar{\tau}=t}^{\tau-1} \Big(12\tilde{L}_1^2 \bar{\eta}_t^\tau \eta_{\bar{\tau}} + 4L^2 \bar{\gamma}_t^\tau \gamma_{\bar{\tau}} \Big) \|z_t - z_{\bar{\tau}}\|^2 \Big) \\ & + 2\bar{\alpha}_t \bar{\eta}_t^2 \|\nabla h(x_t)\|^2 + 2\bar{\alpha}_t \bar{\gamma}_t^2 \|\nabla_y g(x_t, y_t)\|^2 \\ & + 12\tilde{L}_1^2 \bar{\alpha}_t (\bar{\eta}_t)^2 \|y_{x_t} - y_t\|^2 + 16L^2 \bar{\alpha}_t (\bar{\eta}_t)^2 \|u_{x_t} - u_t\|^2 \\ & + 16L^2 \bar{\eta}_t \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(\sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \|u_t - u_{\bar{\tau}}\|^2 \Big) \\ & + 12\bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2 + \frac{24C_f^2}{\mu^2} \bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2 + 4\bar{\eta}_t \gamma_t^2 k^{2-\alpha} C^2\end{aligned}$$

By the fact that $\tau \leq t + k$, we have:

$$\begin{aligned}\sum_{\tau=t}^{t+k-1} \eta_\tau \|z_t - z_\tau\|^2 \leq & \sum_{\tau=t}^{t+k-1} \Big(12\tilde{L}_1^2 (\bar{\eta}_t)^2 \eta_\tau + 4L^2 \bar{\eta}_t \bar{\gamma}_t \gamma_\tau \Big) \|z_t - z_\tau\|^2 \\ & + 2\bar{\eta}_t^3 \|\nabla h(x_t)\|^2 + 2\bar{\eta}_t \bar{\gamma}_t^2 \|\nabla_y g(x_t, y_t)\|^2 \\ & + 12\tilde{L}_1^2 (\bar{\eta}_t)^3 \|y_{x_t} - y_t\|^2 + 16L^2 (\bar{\eta}_t)^3 \|u_{x_t} - u_t\|^2 \\ & + 16L^2 (\bar{\eta}_t)^2 \sum_{\tau=t}^{t+k-1} \eta_\tau \|u_t - u_\tau\|^2 \\ & + 12\bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2 + \frac{24C_f^2}{\mu^2} \bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2 + 4\bar{\eta}_t \gamma_t^2 k^{2-\alpha} C^2 \qquad (3.23)\end{aligned}$$

For $\|u_t - u_\tau\|^2$, we combine Eq. (3.22) with Eq. (3.17) and sum τ over $[t, t+k-1]$ and use the fact that $\tau \leq t + k$ to have:

$$\sum_{\tau=t}^{t+k-1} \eta_\tau \|u_t - u_\tau\|^2 \leq 2\bar{\eta}_t (\bar{\rho}_t)^2 \|\nabla r_{x_t}(u_t)\|^2 + 12\tilde{L}_2^2 \bar{\eta}_t (\bar{\rho}_t)^2 \|y_{x_t} - y_t\|^2$$

$$+ 12\tilde{L}_2^2\bar{\eta}_t\bar{\rho}_t\sum_{\tau=t}^{t+k-1}\rho_\tau\|z_t - z_\tau\|^2 + 8L^2\bar{\eta}_t\bar{\rho}_t\sum_{\tau=t}^{t+k-1}\rho_\tau\|u_t - u_\tau\|^2$$
$$+ 12\bar{\eta}_t\rho_t^2k^{2-\alpha}C^2 + \frac{24C_f^2\bar{\eta}_t}{\mu^2}\rho_t^2k^{2-\alpha}C^2 \tag{3.24}$$

Suppose we have $\gamma_\tau = c_1\eta_\tau = \frac{20\kappa\tilde{L}_3}{\mu}\eta_\tau$, $\rho_\tau = c_2\eta_\tau = 40\kappa\hat{L}\eta_\tau$ and set:

$$(\bar{\eta}_t)^2 < \min\left(\frac{1}{96\tilde{L}_1^2}, \frac{1}{32L^2c_1^2}, \frac{1}{128L^2}, \frac{1}{64L^2c_2^2}, \frac{1}{96\tilde{L}_2^2c_2^2}\right)$$

Then we combine Eq. (3.23) and Eq. (3.24) to have:

$$\sum_{\tau=t}^{t+k-1}\eta_\tau\|u_t - u_\tau\|^2 \leq 4c_2^2(\bar{\eta}_t)^3\|\nabla r_{x_t}(u_t)\|^2 + 4\bar{\eta}_t^3\|\nabla h(x_t)\|^2 + 4c_1^2\bar{\eta}_t^3\|\nabla_y g(x_t, y_t)\|^2$$
$$+ \Big(24 + \frac{48C_f^2}{\mu^2} + 8c_1^2 + 24c_2^2 + \frac{48c_2^2C_f^2}{\mu^2}\Big)\bar{\eta}_t\eta_t^2k^{2-\alpha}C^2 \tag{3.25}$$

and

$$\sum_{\tau=t}^{t+k-1}\eta_\tau\|z_t - z_\tau\|^2 \leq 4c_2^2(\bar{\eta}_t)^3\|\nabla r_{x_t}(u_t)\|^2 + 4\bar{\eta}_t^3\|\nabla h(x_t)\|^2 + 4c_1^2\bar{\eta}_t^3\|\nabla_y g(x_t, y_t)\|^2$$
$$+ \Big(24 + \frac{48C_f^2}{\mu^2} + 8c_1^2 + 24c_2^2 + \frac{48c_2^2C_f^2}{\mu^2}\Big)\bar{\eta}_t\eta_t^2k^{2-\alpha}C^2 \tag{3.26}$$

This completes the proof. □

Suppose we define the potential function $\mathcal{G}(t)$:

$$\mathcal{G}_t = h(x_t) + \phi_y\|y_t - y_{x_t}\|^2 + \phi_u\|u_t - u_{x_t}\|^2$$

where $\phi_u = \frac{32L^2\bar{\eta}_t}{\mu\bar{\rho}_t}$, $\phi_y = \frac{8\tilde{L}_3^2\bar{\eta}_t}{\mu\bar{\gamma}_t}$ and $\tilde{L}_3^2 = 960\kappa^2\tilde{L}_2^2 + 3\tilde{L}_1^2$.

Theorem 46. *Suppose Assumptions 26-29 are satisfied, and Assumption 30 is satisfied with some α and C for the example order we use in Algorithm 2. We choose learning rates $\eta_t = \eta$, $\gamma = c_1\eta$, $\rho = c_2\eta$ and denote $T = R \times I$ be the total number of steps. Then for any pair of values (E, k) which has $T = E \times k$ and $\eta \leq \frac{1}{k\tilde{L}_0}$, we have:*

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{2\Delta}{kE\eta} + 2\tilde{L}^2 k^{-\alpha} C^2$$

where c_1, c_2, $\tilde{L}_0$, $\tilde{L}$ are constants related to the smoothness parameters of $h(x)$, Δ is the initial sub-optimality, R and I are defined in Algorithm 2.

Proof. First we combine Lemma 42-Lemma 44, and use the condition that $\gamma_t = \frac{20\kappa\tilde{L}_3}{\mu}\eta_t$, $\rho_t = 40\kappa\hat{L}\eta_t$, and by the assumption about gradient error:

$$\begin{aligned}
\mathcal{G}_{t+k} - \mathcal{G}_t \leq & -\frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 - \frac{4c_1\tilde{L}_3^2(\bar{\eta}_t)^2}{\mu}\|\nabla_y g(x_t, y_t)\|^2 - \frac{16c_2L^2(\bar{\eta}_t)^2}{\mu}\|\nabla r_{x_t}(u_t)\|^2 \\
& - \tilde{L}_3^2\bar{\eta}_t\|y_{x_t} - y_t\|^2 - 4L^2\bar{\eta}_t\|u_{x_t} - u_t\|^2 \\
& + \sum_{\tau=t}^{t+k-1}\Big(960\kappa^2\tilde{L}_2^2\eta_\tau + 3\tilde{L}_1^2\eta_\tau + 80\kappa^2\tilde{L}_3^2\eta_\tau\Big)\|z_t - z_\tau\|^2 \\
& + \sum_{\tau=t}^{t+k-1}\Big(640\kappa^2L^2\eta_\tau + 4L^2\eta_\tau\Big)\|u_t - u_\tau\|^2 \\
& + \Big(3 + \frac{6C_f^2}{\mu^2} + \frac{80\tilde{L}_3^2}{\mu^2} + 960\kappa^2 + \frac{1920\kappa^2C_f^2}{\mu^2}\Big)\frac{\eta_t^2k^{2-\alpha}C^2}{\bar{\eta}_t}
\end{aligned} \tag{3.27}$$

Suppose we denote $\tilde{L}_4^2 = \Big(960\kappa^2\tilde{L}_2^2 + 3\tilde{L}_1^2 + 80\kappa^2\tilde{L}_3^2 + 640\kappa^2L^2 + 4L^2\Big)$, and set:

$$(\bar{\eta}_t)^2 < \min\Big(\frac{1}{16\tilde{L}_4^2}, \frac{\tilde{L}_3^4}{c_1^2\mu^2\tilde{L}_4^4}, \frac{16L^4}{c_2^2\mu^2\tilde{L}_4^4}, \frac{\tilde{L}_3^2}{48\tilde{L}_4^2\tilde{L}_2^2c_2^2}, \frac{\tilde{L}_3^2}{48\tilde{L}_4^2\tilde{L}_1^2}, \frac{1}{8\tilde{L}_4^2}\Big)$$

and we bound the terms $\|u_{x_t} - u_t\|^2$ and $\|z_t - z_\tau\|^2$ through Lemma 45 to have:

$$\mathcal{G}_{t+k} - \mathcal{G}_t \leq -\frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 + \Big(24 + \frac{48C_f^2}{\mu^2} + 8c_1^2 + 24c_2^2 + \frac{48c_2^2C_f^2}{\mu^2}\Big)\tilde{L}_4^2\bar{\eta}_t\eta_t^2k^{2-\alpha}C^2 + \Big(3 + \frac{6C_f^2}{\mu^2} + \frac{80\tilde{L}_3^2}{\mu^2} + 960\kappa^2 + \frac{1920\kappa^2C_f^2}{\mu^2}\Big)\frac{\eta_t^2k^{2-\alpha}C^2}{\bar{\eta}_t}$$

Next suppose we set $\tilde{L}_4\bar{\eta}_t < 1$, then we have:

$$\mathcal{G}_{t+k} - \mathcal{G}_t \leq -\frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 + \Big(24 + \frac{48C_f^2}{\mu^2} + 8c_1^2 + 24c_2^2 + \frac{48c_2^2C_f^2}{\mu^2} + 3 + \frac{6C_f^2}{\mu^2} + \frac{80\tilde{L}_3^2}{\mu^2} + 960\kappa^2 + \frac{1920\kappa^2C_f^2}{\mu^2}\Big)\frac{\eta_t^2k^{2-\alpha}C^2}{\bar{\eta}_t}$$

For ease of notation we denote $\tilde{L}^2 = \Big(24 + \frac{48C_f^2}{\mu^2} + 8c_1^2 + 24c_2^2 + \frac{48c_2^2C_f^2}{\mu^2} + 3 + \frac{6C_f^2}{\mu^2} + \frac{80\tilde{L}_3^2}{\mu^2} + 960\kappa^2 + \frac{1920\kappa^2C_f^2}{\mu^2}\Big)$, then we have:

$$\mathcal{G}_{t+k} - \mathcal{G}_t \leq -\frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 + \frac{\tilde{L}^2\eta_t^2k^{2-\alpha}C^2}{\bar{\eta}_t}$$

Sum the above inequality over E phases to have:

$$\begin{aligned}\sum_{e=0}^{E-1}\frac{\bar{\eta}_{ke}}{2}\|\nabla h(x_{ke})\|^2 &\leq \mathcal{G}_0 - \mathcal{G}_{kE} + \sum_{e=0}^{E-1}\frac{\tilde{L}^2\eta_{ke}^2k^{2-\alpha}C^2}{\bar{\eta}_{ke}} \\ &\leq \Delta_h + \phi_y\Delta_y + \phi_u\Delta_u + \sum_{e=0}^{E-1}\frac{\tilde{L}^2\eta_{ke}^2k^{2-\alpha}C^2}{\bar{\eta}_{ke}}\end{aligned}$$

we define $\Delta_h = h(x_0) - h^*$ as the initial sub-optimality of the function, $\Delta_y = \|y_0 - y_{x_0}\|^2$ as the initial sub-optimality of the inner variable estimation, $\Delta_u = \|u_0 - u_{x_0}\|^2$ as the initial sub-optimality of the hyper-gradient estimation. Then suppose we choose $\eta_t = \eta$ be some

constant, then we have:

$$\frac{k\eta}{2}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \Delta_h + \phi_y\Delta_y + \phi_u\Delta_u + \tilde{L}^2 E k^{1-\alpha}C^2\eta$$

Divide by $kE\eta/2$ on both sides to have:

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{2}{kE\eta}\Big(\Delta_h + \phi_y\Delta_y + \phi_u\Delta_u\Big) + 2\tilde{L}^2 k^{-\alpha}C^2$$

Suppose we choose the largest $\eta = \frac{1}{k\tilde{L}_0}$ such that the conditions of Lemma 42-Lemma 45 are satisfied, then we have:

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{2\tilde{L}_0}{E}\Big(\Delta + \phi_y\Delta_y + \phi_u\Delta_u\Big) + 2\tilde{L}^2 k^{-\alpha}C^2$$

Then to reach an ϵ-stationary point, we need to have:

$$E \geq \frac{4\tilde{L}_0\Delta}{\epsilon^2}, \text{ and }, k \geq \left(\frac{4\tilde{L}^2C^2}{\epsilon^2}\right)^{1/\alpha}$$

in other words $T = Ek \geq \frac{4\tilde{L}_0(4\tilde{L}^2C^2)^{1/\alpha}\Delta}{\epsilon^{2+2/\alpha}}$. This completes the proof. □

3.6.2 Proof for conditional bilevel optimization problems

In this section we study the convergence rate of Algorithm 3.

Lemma 47. *For all $t \geq 0$ and any constant $k > 0$, suppose $\bar{\eta}_t < \frac{1}{4L}$, the iterates generated*

satisfy:

$$h(x_{t+k}) \le h(x_t) - \frac{k\eta}{8}\|\nabla h(x_t)\|^2 - \frac{k\eta}{4}\|\bar{\nu}_t\|^2 + 3\Big(1 + \frac{4\kappa^2(192C_f^2 + 96 * 24\tilde{L}_2^2)}{\mu^2} + \frac{48\tilde{L}_1^2}{\mu^2}\Big)\eta k^{1-\alpha}C^2 + 12\eta k L^2(1 - \frac{k\mu\rho}{2})^{E_l}\Delta_u + 3\eta k(2\tilde{L}_1^2 + 4 * 96\kappa^2\tilde{L}_2^2)(1 - \frac{k\mu\gamma}{2})^{E_l}\Delta_y$$

Proof. By the smoothness of f, follow similar derivation as in Eq. (3.19), we have:

$$h(x_{t+k}) \le h(x_t) - \frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 + \frac{\bar{\eta}_t}{2}\|\nabla h(x_t) - \bar{\nu}_t\|^2 - \frac{\bar{\eta}_t}{4}\|\bar{\nu}_t\|^2$$

In particular, for the term $\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}(\nu_{\bar{\tau}} - \nabla h(x_t))\|^2$, we have:

$$\begin{aligned}
&\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla h(x_t) - \nu_{\bar{\tau}}\Big)\Big\|^2 \\
&\le 3\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla h(x_t) - \nabla h(x_t;\xi_{\bar{\tau}})\Big)\Big\|^2 \\
&\quad + 3\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla h(x_t;\xi_{\bar{\tau}}) - \nabla h(x_{\bar{\tau}};\xi_{\bar{\tau}})\Big)\Big\|^2 + 3\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla h(x_{\bar{\tau}};\xi_{\bar{\tau}}) - \nu_{\bar{\tau}}\Big)\Big\|^2 \\
&\le 3\Big\|\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big(\nabla h(x_t) - \nabla h(x_t;\xi_{\bar{\tau}})\Big)\Big\|^2 \\
&\quad + 3\bar{L}^2\bar{\eta}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big\|x_t - x_{\bar{\tau}}\Big\|^2 + 3\bar{\eta}_t^{\tau}\sum_{\bar{\tau}=t}^{\tau-1}\eta_{\bar{\tau}}\Big\|\nabla h(x_{\bar{\tau}};\xi_{\bar{\tau}}) - \nu_{\bar{\tau}}\Big\|^2
\end{aligned} \tag{3.28}$$

For the third term, we follow similar derivation as in Eq. (3.13)-Eq. (3.16), we have:

$$\begin{aligned}
&\Big\|\Big(\nabla h(x_\tau;\xi_\tau) - \nu_\tau\Big)\Big\|^2 \\
&= \Big\|\Big(\nabla_x f(x_\tau, y_{x_\tau}^{\xi_\tau};\xi_\tau) - \nabla_{xy} g^{\xi_\tau}(x_\tau, y_{x_\tau}^{\xi_\tau})u_{x_\tau}^{\xi_\tau}
\end{aligned}$$

$$- \left(\nabla_x f(x_\tau, y_{\tau,T_l}; \xi_\tau) - \nabla_{xy} g^{\xi_\tau}(x_\tau, y_{\tau,T_l}) u_{\tau,T_l}\right)\Big\|^2$$

$$\leq 2\tilde{L}_1^2 \|y_{x_\tau}^{\xi_\tau} - y_{\tau,T_l}\|^2 + 4L^2 \|u_{x_\tau}^{\xi_\tau} - u_{\tau,T_l}\|^2 \tag{3.29}$$

where $\tilde{L}_1^2 = \left(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2}\right)$. Combine Eq. (3.28) and Eq. (3.29), we have:

$$h(x_{t+k}) \leq h(x_t) - \frac{\bar{\eta}_t}{2}\|\nabla h(x_t)\|^2 - \frac{\bar{\eta}_t}{4}\|\bar{\nu}_t\|^2 + \frac{3}{2\bar{\eta}_t}\Big\| \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(\nabla h(x_t) - \nabla h(x_t; \xi_\tau)\Big)\Big\|^2$$
$$+ \frac{3\bar{L}^2}{2} \sum_{\tau=t}^{t+k-1} \eta_\tau \Big\|x_t - x_\tau\Big\|^2 + \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(3\tilde{L}_1^2 \|y_{x_\tau}^{\xi_\tau} - y_{\tau,T_l}\|^2 + 6L^2 \|u_{x_\tau}^{\xi_\tau} - u_{\tau,T_l}\|^2\Big) \tag{3.30}$$

For the term $\sum_{\tau=t}^{t+k-1} \eta_\tau \Big\|x_t - x_\tau\Big\|^2$, by Eq. (3.28) and Eq. (3.29), we have:

$$\Big\|x_t - x_\tau\Big\|^2 \leq 2\| \sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \Big(\nu_{\bar{\tau}} - \nabla h(x_t)\Big)\|^2 + 2(\bar{\eta}_t^\tau)^2 \|\nabla h(x_t)\|^2$$
$$\leq 6\Big\| \sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \Big(\nabla h(x_t) - \nabla h(x_t; \xi_{\bar{\tau}})\Big)\Big\|^2 + 6\bar{L}^2 \bar{\eta}_t^\tau \sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \Big\|x_t - x_{\bar{\tau}}\Big\|^2$$
$$+ 6\bar{\eta}_t^\tau \sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \Big(2\tilde{L}_1^2 \|y_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - y_{\bar{\tau},T_l}\|^2 + 4L^2 \|u_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - u_{\bar{\tau},T_l}\|^2\Big) + 2(\bar{\eta}_t^\tau)^2 \|\nabla h(x_t)\|^2$$
$$\leq 6\eta_t^2 k^{2-\alpha} C^2 + 6\bar{L}^2 \bar{\eta}_t \sum_{\bar{\tau}=t}^{t+k-1} \eta_{\bar{\tau}} \Big\|x_t - x_{\bar{\tau}}\Big\|^2$$
$$+ 6\bar{\eta}_t \sum_{\bar{\tau}=t}^{t+k-1} \eta_{\bar{\tau}} \Big(2\tilde{L}_1^2 \|y_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - y_{\bar{\tau},T_l}\|^2 + 4L^2 \|u_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - u_{\bar{\tau},T_l}\|^2\Big) + 2(\bar{\eta}_t)^2 \|\nabla h(x_t)\|^2$$

Multiply η_τ on both sides and sum over $[t, t+k-1]$, we have:

$$(1 - 6\bar{L}^2(\bar{\eta}_t)^2) \sum_{\tau=t}^{t+k-1} \eta_\tau \Big\|x_t - x_\tau\Big\|^2 \leq 6(\bar{\eta}_t)^2 \sum_{\bar{\tau}=t}^{t+k-1} \eta_{\bar{\tau}} \Big(2\tilde{L}_1^2 \|y_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - y_{\bar{\tau},T_l}\|^2 + 4L^2 \|u_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - u_{\bar{\tau},T_l}\|^2\Big)$$
$$+ 2(\bar{\eta}_t)^3 \|\nabla h(x_t)\|^2 + 6\bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2$$

By the condition that $\bar{\eta}_t < \frac{1}{4\bar{L}}$, we have:

$$\sum_{\tau=t}^{t+k-1} \eta_\tau \Big\| x_t - x_\tau \Big\|^2 \leq 12(\bar{\eta}_t)^2 \sum_{\bar{\tau}=t}^{t+k-1} \eta_{\bar{\tau}} \Big(2\tilde{L}_1^2 \| y_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - y_{\bar{\tau},T_l} \|^2 + 4L^2 \| u_{x_{\bar{\tau}}}^{\xi_{\bar{\tau}}} - u_{\bar{\tau},T_l} \|^2 \Big)$$
$$+ 4(\bar{\eta}_t)^3 \| \nabla h(x_t) \|^2 + 12 \bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2 \tag{3.31}$$

Combine Eq. (3.31) with Eq. (3.30), then we have:

$$\begin{aligned}
& h(x_{t+k}) \\
& \leq h(x_t) - (\frac{\bar{\eta}_t}{2} - 6\bar{L}^2(\bar{\eta}_t)^3) \| \nabla h(x_t) \|^2 - \frac{\bar{\eta}_t}{4} \| \bar{\nu}_t \|^2 + \frac{3}{2\bar{\eta}_t} \Big\| \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(\nabla h(x_t) - \nabla h(x_t; \xi_\tau) \Big) \Big\|^2 \\
& \quad + 18\bar{L}^2 \bar{\eta}_t \eta_t^2 k^{2-\alpha} C^2 + (1 + 12(\bar{\eta}_t)^2 \bar{L}^2) \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(3\tilde{L}_1^2 \| y_{x_\tau}^{\xi_\tau} - y_{\tau,T_l} \|^2 + 6L^2 \| u_{x_\tau}^{\xi_\tau} - u_{\tau,T_l} \|^2 \Big) \\
& \leq h(x_t) - \frac{\bar{\eta}_t}{8} \| \nabla h(x_t) \|^2 - \frac{\bar{\eta}_t}{4} \| \bar{\nu}_t \|^2 + \frac{3}{\bar{\eta}_t} \eta_t^2 k^{2-\alpha} C^2 \\
& \quad + \sum_{\tau=t}^{t+k-1} \eta_\tau \Big(6\tilde{L}_1^2 \| y_{x_\tau}^{\xi_\tau} - y_{\tau,T_l} \|^2 + 12L^2 \| u_{x_\tau}^{\xi_\tau} - u_{\tau,T_l} \|^2 \Big)
\end{aligned} \tag{3.32}$$

where the second inequality uses the conditions that $\bar{\eta}_t < \frac{1}{4\bar{L}}$. By Lemma 49,we have:

$$\begin{aligned}
h(x_{t+k}) \leq h(x_t) & - \frac{k\eta}{8} \| \nabla h(x_t) \|^2 - \frac{k\eta}{4} \| \bar{\nu}_t \|^2 \\
& + 3\Big(1 + \frac{4\kappa^2 (192 C_f^2 + 96 * 24 \tilde{L}_2^2)}{\mu^2} + \frac{48\tilde{L}_1^2}{\mu^2} \Big) \eta k^{1-\alpha} C^2 \\
& + 12\eta k L^2 (1 - \frac{k\mu\rho}{2})^{E_l} \Delta_u + 3\eta k (2\tilde{L}_1^2 + 4 * 96\kappa^2 \tilde{L}_2^2)(1 - \frac{k\mu\gamma}{2})^{E_l} \Delta_y
\end{aligned} \tag{3.33}$$

where Δ_u and Δ_y denotes the upper bounds of the initial estimation errors of y_x and u_x.

This completes the proof. □

Lemma 48. *Suppose we have $\bar{\gamma}_l < \frac{1}{2L}$ and $\bar{\rho}_l < \frac{1}{2L}$, then $\sum_{l'=l}^{l+k-1} \gamma_{l'} \| y_{l'} - y_l \|^2$ and*

$\sum_{l'=l}^{l+k-1} \rho_{l'} \|u_l - u_{l'}\|^2$ *can be bounded as in Eq.* (3.34) *and Eq.* (3.35)

Proof. We first bound $\sum_{l'=l}^{l+k-1} \gamma_{l'} \|y_{l'} - y_l\|^2$, in fact, we have:

$$\begin{aligned}
\|y_l - y_{l'}\|^2 = \|\sum_{\bar{l}=l}^{l'-1} \gamma_{\bar{l}} w_{\bar{l}}\|^2 &\leq 2\|\sum_{\bar{l}=l}^{l'-1} \gamma_{\bar{l}} \Big(\nabla_y g^{\zeta_{\bar{l}}}(x, y_{\bar{l}}) - \nabla_y g(x, y_l)\Big)\|^2 + 2(\bar{\gamma}_l^{l'})^2 \|\nabla_y g(x, y_l)\|^2 \\
&\leq 4L^2 \bar{\gamma}_l^{l'} \sum_{\bar{l}=l}^{l'-1} \gamma_{\bar{l}} \|y_{\bar{l}} - y_l\|^2 + 2(\bar{\gamma}_l^{l'})^2 \|\nabla_y g(x, y_l)\|^2 \\
&\quad + 4\|\sum_{\bar{l}=l}^{l'-1} \gamma_{\bar{l}} \Big(\nabla_y g^{\zeta_{\bar{l}}}(x, y_l) - \nabla_y g(x, y_l)\Big)\|^2 \\
&\leq 4L^2 \bar{\gamma}_l \sum_{\bar{l}=l}^{l+k-1} \gamma_{\bar{l}} \|y_{\bar{l}} - y_l\|^2 + (\bar{\gamma}_l)^2 \|\nabla_y g(x, y_l)\|^2 + 4\gamma_l^2 k^{2-\alpha} C^2
\end{aligned}$$

Multiply $\gamma_{l'}$ on both sides, and then sum l' in $[l, l+k-1]$, we have:

$$(1 - 4L^2 \bar{\gamma}_l^2) \sum_{l'=l}^{l+k-1} \gamma_{l'} \|y_{l'} - y_l\|^2 \leq 2(\bar{\gamma}_l)^3 \|\nabla_y g(x, y_l)\|^2 + 4\bar{\gamma}_l \gamma_l^2 k^{2-\alpha} C^2$$

Since we have $\bar{\gamma}_l < \frac{1}{2L}$, we have:

$$\sum_{l'=l}^{l+k-1} \gamma_{l'} \|y_{l'} - y_l\|^2 \leq 4(\bar{\gamma}_l)^3 \|\nabla_y g(x, y_l)\|^2 + 8\bar{\gamma}_l \gamma_l^2 k^{2-\alpha} C^2 \tag{3.34}$$

Next for $\sum_{l'=l}^{l+k-1} \rho_{l'} \|u_l - u_{l'}\|^2$, we have:

$$\begin{aligned}
\|u_l - u_{l'}\|^2 = \|\sum_{\bar{l}=t}^{l'-1} \rho_{\bar{l}} q_{\bar{l}}\|^2 &\leq 2\|\sum_{\bar{l}=l}^{l'-1} \rho_{\bar{l}} \Big(q_{\bar{l}} - \nabla r_x(u_l)\Big)\|^2 + 2(\bar{\rho}_l^{l'})^2 \|\nabla r_x(u_l)\|^2 \\
&\leq 12\tilde{L}_2^2 (\bar{\rho}_l^{l'})^2 \|y_x - y_{T_l}\|^2 + 8L^2 \bar{\rho}_l^{l'} \sum_{\bar{l}=l}^{l'-1} \rho_{\bar{l}} \|u_l - u_{\bar{l}}\|^2 + 2(\bar{\rho}_l^{l'})^2 \|\nabla r_x(u_l)\|^2 \\
&\quad + \frac{24C_f^2}{\mu^2} \|\sum_{\bar{l}=l}^{l'-1} \rho_{\bar{l}} \Big(\nabla_{y^2} g(x, y_{T_l}) - \nabla_{y^2} g^{\zeta_{\bar{l}}}(x, y_{T_l})\Big)\|^2
\end{aligned}$$

$$\leq 12\tilde{L}_2^2(\bar{\rho}_l)^2\|y_x - y_{T_l}\|^2 + 8L^2\bar{\rho}_l \sum_{\bar{l}'=l}^{l+k-1} \rho_{\bar{l}'}\|u_l - u_{\bar{l}'}\|^2 + 2(\bar{\rho}_l)^2\|\nabla r_x(u_l)\|^2$$
$$+ \frac{24C_f^2}{\mu^2}\rho_l^2 k^{2-\alpha}C^2$$

Multiply $\rho_{l'}$ on both sides, and then sum l' in $[l, l+k-1]$, we have:

$$(1 - 8L^2\bar{\rho}_l^2) \sum_{l'=l}^{l+k-1} \rho_{l'}\|u_l - u_{l'}\|^2 \leq 12\tilde{L}_2^2(\bar{\rho}_l)^3\|y_x - y_{T_l}\|^2$$
$$+ 2(\bar{\rho}_l)^3\|\nabla r_x(u_l)\|^2 + \frac{24C_f^2}{\mu^2}\bar{\rho}_l\rho_l^2 k^{2-\alpha}C^2$$

Since we have $\bar{\rho}_l < \frac{1}{2L}$, we have:

$$\sum_{l'=l}^{l+k-1} \rho_{l'}\|u_l - u_{l'}\|^2 \leq 24\tilde{L}_2^2(\bar{\rho}_l)^3\|y_x - y_{T_l}\|^2 + 4(\bar{\rho}_l)^3\|\nabla r_x(u_l)\|^2 + \frac{48C_f^2}{\mu^2}\bar{\rho}_l\rho_l^2 k^{2-\alpha}C^2 \quad (3.35)$$

□

Lemma 49. *When $\bar{\gamma}_t < \frac{1}{128L\kappa}$ and $\bar{\rho}_t < \frac{1}{256L\kappa}$, the error of the inner variable y_l and the variable u_l are bounded through Eq. (3.36) and Eq. (3.37).*

Proof. Since the outer iteration t is fixed for each inner loop, we omit t and ξ_t in the notation for clarity. Furthermore, we assume performing the updates to y and u separately in Algorithm 3. Although the alternative updates in Algorithm 3 is appealing in practice as we can reuse $\nabla_y g$ for $\nabla_{y^2} g$, update y first and u next can avoid treating complicated higher order terms. Follow similar derivation of Lemma 43, if $\bar{\gamma}_t < \frac{1}{4L}$, we have:

$$\|y_{l+k} - y_x\|^2 \leq (1 - \frac{\mu\bar{\gamma}_l}{2})\|y_l - y_x\|^2 - \frac{\bar{\gamma}_l^2}{2}\|\nabla_y g(x, y_l)\|^2$$

$$+ 8L\kappa \sum_{l'=l}^{l+k-1} \gamma_{l'} \|y_{l'} - y_l\|^2 + \frac{8}{\mu\bar{\gamma}_l} \| \sum_{l'=l}^{l+k-1} \gamma_{l'} \Big(\nabla_y g(x, y_l) - \nabla_y g(x, y_l; \zeta_{l'}) \Big) \|^2$$

Combine with Eq. (3.34) and use the condition that $\bar{\gamma}_l < \frac{1}{128L\kappa} < \frac{1}{4L}$, we have:

$$\|y_{l+k} - y_x\|^2 \leq (1 - \frac{\mu\bar{\gamma}_l}{2}) \|y_l - y_x\|^2 - \frac{\bar{\gamma}_l^2}{4} \|\nabla_y g(x, y_l)\|^2 + 64L\kappa\bar{\gamma}_l\gamma_l^2 k^{2-\alpha} C^2 + \frac{8}{\mu\bar{\gamma}_l} \gamma_l^2 k^{2-\alpha} C^2$$

Suppose we perform E_l rounds, then by telescoping, we have:

$$\begin{aligned} \|y_{T_l} - y_x\|^2 \leq (1 - \frac{k\mu\gamma}{2})^{E_l} \|y_0 - y_x\|^2 \\ + \frac{24k^{-\alpha}C^2}{\mu^2} + \sum_{e=0}^{E_l - 1} (1 - \frac{k\mu\gamma}{2})^{E_l - e} \Big(- \frac{k^2\gamma^2}{4} \|\nabla_y g(x, y_{ek})\|^2 \Big) \end{aligned} \tag{3.36}$$

Next, follow similar derivations as in Lemma 44 and choose $\bar{\rho}_t < \frac{1}{4L}$, then we have:

$$\begin{aligned} \|u_{l+k} - u_x\|^2 \leq (1 - \frac{\mu\bar{\rho}_l}{2}) \|u_l - u_x\|^2 - \frac{(\bar{\rho}_l)^2}{2} \|\nabla r_x(u_l)\|^2 + \frac{24\tilde{L}_2^2(\bar{\rho}_t)}{\mu} \|y_x - y_{T_l}\|^2 \\ + 16L\kappa \sum_{l'=l}^{l+k-1} \rho_{l'} \|u_l - u_{l'}\|^2 \\ + \frac{48C_f^2}{\mu^3\bar{\rho}_l} \| \sum_{l'=l}^{l+k-1} \rho_{l'} \Big(\nabla_{y^2} g(x, y_{T_l}) - \nabla_{y^2} g(x, y_{T_l}; \zeta_{l'}) \Big) \|^2 \end{aligned}$$

Combine with Eq. (3.35) and use the condition that $\bar{\rho}_l < \frac{1}{256L\kappa} < \frac{1}{4L}$, we have:

$$\begin{aligned} \|u_{l+k} - u_x\|^2 \leq (1 - \frac{\mu\bar{\rho}_l}{2}) \|u_l - u_x\|^2 - \frac{(\bar{\rho}_l)^2}{4} \|\nabla r_x(u_l)\|^2 \\ + \frac{48\tilde{L}_2^2(\bar{\rho}_l)}{\mu} \|y_x - y_{T_l}\|^2 + \frac{96C_f^2}{\mu^3\bar{\rho}_l} \rho_l^2 k^{2-\alpha} C^2 \end{aligned}$$

we perform E_l rounds, then by telescoping, we have:

$$\|u_{T_l} - u_x\|^2 \leq (1 - \frac{k\mu\rho}{2})^{E_l}\|u_0 - u_x\|^2 + \frac{192C_f^2C^2k^{-\alpha}}{\mu^4} + \frac{96\tilde{L}_2^2}{\mu^2}\|y_x - y_{T_l}\|^2 + \sum_{e=0}^{E_l-1}(1 - \frac{k\mu\rho}{2})^{E_l-e}\Big(- \frac{k^2\rho^2}{4}\|\nabla_y r_x(u_{ek})\|^2\Big) \tag{3.37}$$

This completes the proof. □

Theorem 50. *Suppose Assumptions 26-29 and 32 are satisfied, and Assumption 30 is satisfied with some α and C for the example order we use in Algorithm 3. We choose learning rates $\eta_t = \eta = \frac{1}{8kL}$, $\gamma_t = \gamma = \frac{1}{256kL\kappa}$, $\rho_t = \rho = \frac{1}{512kL\kappa}$ and denote $T = R \times m$ be the total number of outer steps and $T_l = S \times \max(n_i)$ be the maximum inner steps. Then for any pair of values (E, k) which has $T = E \times k$ and $T_l = E_l \times k$, we have:*

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{8\Delta_h}{kE\eta} + C_u(1 - \frac{k\mu\rho}{2})^{E_l}\Delta_u + C_y(1 - \frac{k\mu\gamma}{2})^{E_l}\Delta_y + (C')^2C^2k^{-\alpha}$$

where C_u, C_y, C', $\tilde{L}$ are constants related to the smoothness parameters of $h(x)$, Δ_h, Δ_u, Δ_y are the initial sub-optimality, R and S are defined in Algorithm 3.

Proof. First, by Lemma 47, we have:

$$\begin{aligned} h(x_{t+k}) \leq h(x_t) &- \frac{k\eta}{8}\|\nabla h(x_t)\|^2 - \frac{k\eta}{4}\|\bar{\nu}_t\|^2 \\ &+ 3\Big(1 + \frac{4\kappa^2(192C_f^2 + 96 * 24\tilde{L}_2^2)}{\mu^2} + \frac{48\tilde{L}_1^2}{\mu^2}\Big)\eta k^{1-\alpha}C^2 \\ &+ 12\eta kL^2(1 - \frac{k\mu\rho}{2})^{E_l}\Delta_u + 3\eta k(2\tilde{L}_1^2 + 4 * 96\kappa^2\tilde{L}_2^2)(1 - \frac{k\mu\gamma}{2})^{E_l}\Delta_y \end{aligned}$$

Sum the above inequality over E phases to have:

$$\frac{k\eta}{8}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \Delta_h + 12\eta kEL^2(1-\frac{k\mu\rho}{2})^{E_l}\Delta_u \\ + 3\eta kE(2\tilde{L}_1^2 + 4*96\kappa^2\tilde{L}_2^2)(1-\frac{k\mu\gamma}{2})^{E_l}\Delta_y \\ + 3E\Big(1+\frac{4\kappa^2(192C_f^2+96*24\tilde{L}_2^2)}{\mu^2}+\frac{48\tilde{L}_1^2}{\mu^2}\Big)\eta k^{1-\alpha}C^2$$

Divide by $kE\eta/8$ on both sides to have:

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{8\Delta_h}{kE\eta} + 96L^2(1-\frac{k\mu\rho}{2})^{E_l}\Delta_u + 24(2\tilde{L}_1^2 + 4*96\kappa^2\tilde{L}_2^2)(1-\frac{k\mu\gamma}{2})^{E_l}\Delta_y \\ + 24\Big(1+\frac{4\kappa^2(192C_f^2+96*24\tilde{L}_2^2)}{\mu^2}+\frac{48\tilde{L}_1^2}{\mu^2}\Big)k^{-\alpha}C^2$$

By the condition $\eta < \frac{1}{4k\bar{L}}$, $\gamma < \frac{1}{128kL\kappa}$ and $\rho < \frac{1}{256kL\kappa}$, we choose $\eta = \frac{1}{8k\bar{L}}$, $\gamma = \frac{1}{256kL\kappa}$ and $\rho = \frac{1}{512kL\kappa}$, then we have:

$$\frac{1}{E}\sum_{e=0}^{E-1}\|\nabla h(x_{ke})\|^2 \leq \frac{64\bar{L}\Delta_h}{E} + 96L^2(1-\frac{k\mu\rho}{2})^{E_l}\Delta_u + 24(2\tilde{L}_1^2 + 4*96\kappa^2\tilde{L}_2^2)(1-\frac{k\mu\gamma}{2})^{E_l}\Delta_y \\ + 24\Big(1+\frac{4\kappa^2(192C_f^2+96*24\tilde{L}_2^2)}{\mu^2}+\frac{48\tilde{L}_1^2}{\mu^2}\Big)k^{-\alpha}C^2$$

Then to reach an ϵ-stationary point, we need to have:

$$E \geq \frac{128\bar{L}\Delta}{\epsilon^2}, \text{ and }, k^\alpha \geq \frac{48C^2}{\epsilon^2}\left(1+\frac{4\kappa^2(192C_f^2+96*24\tilde{L}_2^2)}{\mu^2}+\frac{48\tilde{L}_1^2}{\mu^2}\right)$$

and $E_l = O(log(\epsilon^{-1}))$. This completes the proof. □

Proposition 51. *Given Assumptions 28-29, Assumption 32 and Assumption 30 hold, then*

we have: $\| \sum_{\bar{\tau}=t}^{\tau-1} \eta_{\bar{\tau}} \Big(\nabla_y g(x_t, y_{x_t}) - \nabla_y g(x_{\bar{\tau}}, y_{\bar{\tau}}; \zeta_{\bar{\tau},1}) \Big) \|^2 \leq \eta^2 k^{2-\alpha} C^2$ *for any* $\tau \leq t+k-1$. *The same upper bound is achieved for other properties:* $\nabla h(x)$, $\nabla_x f(x,y)$,$\nabla_y g(x,y)$, $\nabla_{y^2} g(x,y)$, $\nabla_{xy} g(x,y)$

Proof. This proposition can be adapted from Lemma 2 of [63]. □

Proposition 52. *Suppose we have function* $g(y)$, *which is* L*-smooth and* μ*-strongly-convex, then suppose* $\gamma < \frac{1}{2L}$, *the progress made by one step of gradient descent is:*

$$\|y_{t+1} - y^*\|^2 \leq \Big(1 - \frac{\mu\gamma}{2}\Big)\|y_t - y^*\|^2 - \frac{\gamma^2}{2}\|\nabla_y g(y_t)\|^2 + \frac{4\gamma}{\mu}\|\nabla_y g(y_t) - w_t\|^2$$

where y^* *is the minimum of* $g(y)$ *and we have update rule* $y_{t+1} = y_t - \gamma\omega_t$.

Proof. First, by the strong convexity of of function $g(y)$, we have:

$$\begin{aligned} g(y^*) &\geq g(y_t) + \langle \nabla_y g(y_t), y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 \\ &= g(y_t) + \langle \nabla_y g(y_t), y^* - y_{t+1} \rangle + \langle \nabla_y g(y_t), y_{t+1} - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 \end{aligned} \tag{3.38}$$

Then by L-smoothness, we have:

$$\frac{L}{2}\|y_{t+1} - y_t\|^2 \geq g(y_{t+1}) - g(y_t) - \langle \nabla_y g(y_t), y_{t+1} - y_t \rangle \tag{3.39}$$

Combining the 3.38 with 3.39, we have:

$$\begin{aligned} g(y^*) &\geq g(y_{t+1}) + \langle \nabla_y g(y_t), y^* - y_t \rangle + \gamma \langle \nabla_y g(y_t), w_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L}{2}\|y_{t+1} - y_t\|^2 \\ &\geq g(y_{t+1}) + \frac{\gamma}{2}\|w_t\|^2 + \frac{\gamma}{2}\|\nabla_y g(y_t)\|^2 - \frac{\gamma}{2}\|w_t - \nabla_y g(y_t)\|^2 + \langle w_t, y^* - y_t \rangle \end{aligned}$$

$$
\begin{aligned}
&+ \langle \nabla_y g(y_t) - w_t, y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L\gamma^2}{2}\|\omega_t\|^2 \\
&\geq g(y_{t+1}) + \Big(\frac{\gamma}{2} - \frac{L\gamma^2}{2}\Big)\|w_t\|^2 + \frac{\gamma}{2}\|\nabla_y g(y_t)\|^2 - \frac{\gamma}{2}\|w_t - \nabla_y g(y_t)\|^2 + \langle w_t, y^* - y_t \rangle \\
&+ \langle \nabla_y g(y_t) - w_t, y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2
\end{aligned}
$$

By definition of y^*, we have $g(y^*) \geq g(y_{t+1})$. Thus, we obtain

$$
\begin{aligned}
0 \geq &\Big(\frac{\gamma}{2} - \frac{L\gamma^2}{2}\Big)\|w_t\|^2 + \frac{\gamma}{2}\|\nabla_y g(y_t)\|^2 - \frac{\gamma}{2}\|w_t - \nabla_y g(y_t)\|^2 + \langle w_t, y^* - y_t \rangle \\
&+ \langle \nabla_y g(y_t) - w_t, y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2
\end{aligned} \tag{3.40}
$$

Considering the outer bound of the second term $\langle \nabla_y g(y_t) - w_t, y^* - y_t \rangle$, we have

$$
- \langle \nabla_y g(y_t) - w_t, y^* - y_t \rangle \leq \frac{1}{\mu}\|\nabla_y g(y_t) - w_t\|^2 + \frac{\mu}{4}\|y^* - y_t\|^2
$$

Combining with Eq. 3.40:

$$
\begin{aligned}
0 \geq &\Big(\frac{\gamma}{2} - \frac{L\gamma^2}{2}\Big)\|w_t\|^2 + \frac{\gamma}{2}\|\nabla_y g(y_t)\|^2 - \Big(\frac{\gamma}{2} + \frac{1}{\mu}\Big)\|w_t - \nabla_y g(y_t)\|^2 + \langle w_t, y^* - y_t \rangle \\
&+ \frac{\mu}{4}\|y^* - y_t\|^2
\end{aligned}
$$

By $y_{t+1} = y_t - \gamma\omega_t$, we have:

$$
\begin{aligned}
\|y_{t+1} - y^*\|^2 = \|y_t - \gamma\omega_t - y^*\|^2 &= \|y_t - y^*\|^2 - 2\gamma\langle \omega_t, y_t - y^* \rangle + \gamma^2\|\omega_t\|^2 \\
&\leq \Big(1 - \frac{\mu\gamma}{2}\Big)\|y_t - y^*\|^2 - \gamma^2\|\nabla_y g(y_t)\|^2 + L\gamma^3\|\omega_t\|^2 \\
&\quad + \Big(\gamma^2 + \frac{2\gamma}{\mu}\Big)\|\nabla_y g(y_t) - w_t\|^2
\end{aligned}
$$

Then since we choose $\gamma < \frac{1}{4L} < \frac{1}{\mu}$, we obtain:

$$\|y_{t+1} - y^*\|^2 \leq \Big(1 - \frac{\mu\gamma}{2}\Big)\|y_t - y^*\|^2 - \frac{\gamma^2}{2}\|\nabla_y g(y_t)\|^2 + \frac{4\gamma}{\mu}\|\nabla_y g(y_t) - w_t\|^2$$

This completes the proof. □

Corollary 53. *For Algorithm 4 and Algorithm 5, suppose Assumptions (the bounded gradient assumption is not required for the minimax problem) and conditions are satisfied as in Theorem 31, to reach an ϵ-stationary point, we need $O(\epsilon^{-4})$ examples if they are sampled independently, while $O((\max(m,n))^p\epsilon^{-3})$ if some random-permutation-based without-replacement sampling is used.*

Corollary 54. *For Algorithm 6, suppose Assumptions and conditions are satisfied as in Theorem 33, to reach an ϵ-stationary point, we need $O(\epsilon^{-6})$ examples if they sampled independently, while $O((\max(m,n))^{2p}\epsilon^{-4})$ if examples are sampled following some random-permutation-based without-replacement sampling.*

It is straightforward to derive the above corollaries from Theorem 31 and Theorem 33, we omit the proof here.

3.7 More Experimental Details

In this section, we introduce more details of the experiments.

Table 3.2: **Comparisons of the Bilevel Opt. & Conditional Bilevel Opt. algorithms for finding an ϵ-stationary point**. An ϵ-stationary point is defined as $\|\nabla h(x)\| \leq \epsilon$. $Gc(f,\epsilon)$ and $Gc(g,\epsilon)$ denote the number of gradient evaluations *w.r.t.* $f(x,y)$ and $g(x,y)$; $JV(g,\epsilon)$ denotes the number of Jacobian-vector products; $HV(g,\epsilon)$ is the number of Hessian-vector products. m and n are the number of data examples for the outer and inner problems, in particular, n is the maximum number of inner problems for conditional bilevel optimization. Our methods have a dependence over example numbers $(max(m,n))^q$, q is a value decided by without-replacement sampling strategy and can have value in $[0,1]$ (A herding-based permutation [1] can let $q=0$).

Setting	Algorithm	$Gc(f,\epsilon)$	$Gc(g,\epsilon)$	$JV(g,\epsilon)$	$HV(g,\epsilon)$
B.O.	BSA [6]	$O(\epsilon^{-4})$	$O(\epsilon^{-6})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	TTSA [15]	$O(\epsilon^{-5})$	$O(\epsilon^{-5})$	$O(\epsilon^{-5})$	$O(\epsilon^{-5})$
	StocBiO [12]	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	SOBA [58]	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-r})$	$O(\epsilon^{-4})$
	MRBO [32]	$O(\epsilon^{-3})$	$O(\epsilon^{-3})$	$O(\epsilon^{-3})$	$O(\epsilon^{-3})$
	VRBO [32]	$O(\epsilon^{-3})$	$O(\epsilon^{-3})$	$O(\epsilon^{-3})$	$O(\epsilon^{-3})$
	SABA [58]	$O(max(m,n)^{2/3}\epsilon^{-2})$	$O(max(m,n)^{2/3}\epsilon^{-2})$	$O(max(m,n)^{2/3}\epsilon^{-2})$	$O(max(m,n)^{2/3}\epsilon^{-2})$
	SRBA [90]	$O(max(m,n)^{1/2}\epsilon^{-2})$	$O(max(m,n)^{1/2}\epsilon^{-2})$	$O(max(m,n)^{1/2}\epsilon^{-2})$	$O(max(m,n)^{1/2}\epsilon^{-2})$
	WiOR-BO(Ours)	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^q\epsilon^{-3})$
Cond. B.O.	DL-SGD [59]	$O(\epsilon^{-4})$	$O(\epsilon^{-6})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	RT-MLMC [59]	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$	$O(\epsilon^{-4})$
	WiOR-CBO (Ours)	$O((max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^{2q}\epsilon^{-4})$	$O(n(max(m,n))^q\epsilon^{-3})$	$O((max(m,n))^{2q}\epsilon^{-4})$

3.7.1 Invariant Risk-Minimization

The data generation process is as follows: we first randomly sample a ground truth x^*, then we randomly generate $m = 1000$ input variables c, and calculate the ground truth label $b = cx$, then for each c_i, we generate a set of $n = 100$ noisy observations by adding Guassian noise of scale 0.1. We also add a L_2 regularization term of strength 0.1 for the outer problem. During trainging, we use both the inner and outer learing rates of 0.001.

3.7.2 Hyper-Data Cleaning

The formulation of the problem is as follows:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq f(x,y_x) = \frac{1}{N^{(val)}}\sum_{n=1}^{N^{(val)}} \Theta(y_x;\xi_n^{val}) \text{ s.t. } y_x = \arg\min_{y\in\mathbb{R}^d} g(x,y) = \sum_{n=1}^{N^{(tr)}} x_n\Theta(y;\xi_n^{tr})$$

In the above formulation, we have a pair of (noisy) training set $\{\xi_n^{tr}\}_{n=1}^{N^{(tr)}}$ and validation set $\{\xi_n^{val}\}_{n=1}^{N^{(val)}}$, and $x_n, n \in [N^{(tr)}]$ are weights for training samples, y is the parameter of a model, and we denote the model by Θ. Note that y_x is the model learned over the weighted training set. We fit a model with 3 fully connected layers for the MNIST dataset. We also use L_2 regularization with coefficient 10^{-3} to satisfy the strong convexity condition. In the Experiments, we choose inner learning rate (γ, ρ) as 0.1 and outer learning rate η 1000.

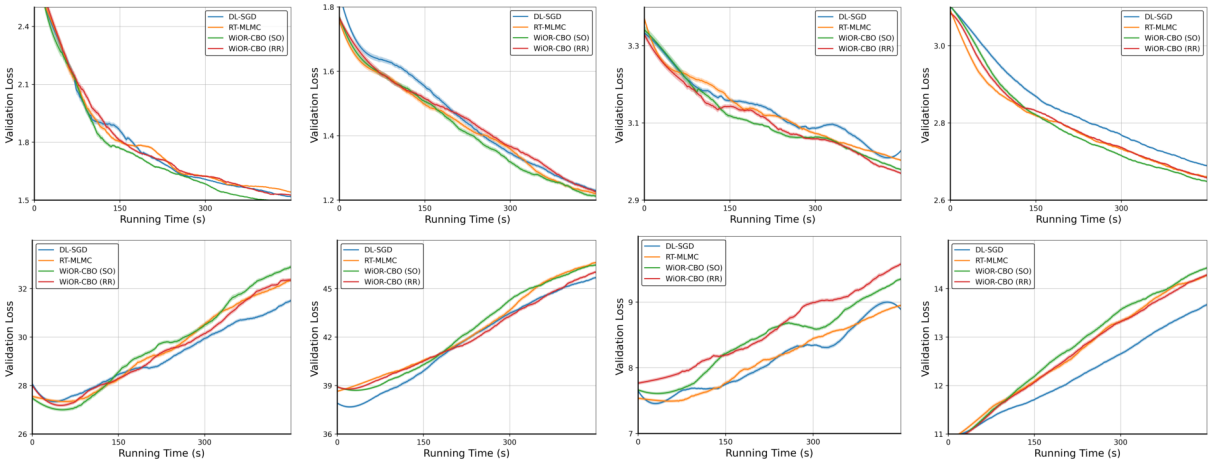


Figure 3.4: Comparison of different algorithms for the Hyper-Representation Task over the MiniImageNet Dataset. From Left to Right: 5-way-1-shot, 5-way-5-shot, 20-way-1-shot, 20-way-5-shot.

3.7.3 Hyper-Representation Learning

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq f(x, y_x) = \frac{1}{N}\sum_{n=1}^{N}\Big(\frac{1}{N_n^{val}}\sum_{i=1}^{N_n^{val}}\Theta(x, y_x^{(\mathcal{T}_n)}; \xi_i^{val})\Big)$$

$$\text{s.t. } y_x^{(\mathcal{T}_n)} = \arg\min_{y\in\mathbb{R}^d} g^{(\mathcal{T}_n)}(x, y) = \frac{1}{N_n^{tr}}\sum_{i=1}^{N_n^{tr}}\Theta(x, y; \xi_i^{tr})$$

In the above formulation, we have N tasks and each task $\mathcal{T}_n$ is defined by a pair of training set $\{\xi_i^{tr}\}_{i=1}^{N_n^{tr}}$ and validation set $\{\xi_i^{val}\}_{i=1}^{N_n^{val}}$. Θ defines the model, x is the parameter of the backbone model and y is the parameter of the linear classifier. In summary, the lower level problem is to learn the optimal linear classifier y given the backbone x, and the upper level problem is to learn the optimal backbone parameter x.

The Omniglot dataset includes 1623 characters from 50 different alphabets and each character consists of 20 samples. We follow the experimental protocols of [89] to divide the alphabets to train/validation/test with 33/5/12, respectively. We perform N-way-K-shot classification, more specifically, for each task, we randomly sample N characters and for each character, we sample K samples for training and 15 samples for validation. We augment the characters by performing rotation operations (multipliers of 90 degrees). We use a 4-layer convolutional neural network where each convolutional layer has 64 filters of 3×3 [91]. For the MiniImageNet, it has 64 training classes and 16 validation classes. Similar to Omniglot, we also perform the N-way-K-shot classification. We use a 4-layer convolutional neural network where each convolutional layer has 64 filters of 3×3 [91] for experiments. For the experiments, we use inner learning rates 0.4 and outer learning rates 0.1 for Omniglot related experiments and inner learning rates 0.01 and outer learning rates 0.05 for MiniImageNet-related experiments. We perform 4 inner gradient descent steps and set $K_max = 6$ for the RT-MLMC method. The experimental results for the MiniImageNet dataset is shown in Figure 3.4.

3.8 Comparison of WiOR-BO with Other Methods

In this section, we compare our algorithms with other variance reduction based algorithms for bilevel optimization. For the unconditional case, WiOR-BO obtain the same $O(\epsilon^{-3})$ rate as STORM-based algorithm MRBO [32], while is slower than SVRG-type algorithms which have $O(n^q\epsilon^{-2})$ such as VRBO [32], SABA [58] and SRBA [90]. This relationship is similar to that for the single level optimization problems. However, as a SGD-type algorithm, WiOR-BO is much simpler to implement in practice compared to STORM-based and ARVG-based algorithms. More specifically, WIOR-BO has fewer hyper-parameters (learning rates for inner and outer problems) to tune compared to MRBO, which has six independent hyper-parameters, and the optimal theoretical convergence rate is achieved only if the complicated conditions among hyper-parameters are satisfied in Theorem 1 of [32] are satisfied, and this requires significant effort in practice. Next, SRBO/SABA evaluates the full hyper-gradient at the start of each outer loop, where we need to evaluate the first and second order derivatives over all samples in one step, which is very expensive for modern ML models (such as the transformer model with billions of parameters). In contrast, our WiOR-BO never evaluates the full gradient. Note that the inner loop length $I = lcm(m, n)$ is analogous to the concept of "epoch" in single level optimization: where we go over the data samples following a given order, but at each step we evaluate gradient over a mini-batch of samples. The practical advantage of our WiOR-BO is further verified by the superior performance over MRBO and VRBO in the Hyper-Data Cleaning Task.

Part II

Nested Problems in Federated/Distributed Learning

Chapter 4: Communication-Efficient Federated Bilevel Optimization with Global and Local Lower Level Problems

4.1 Introduction

Bilevel optimization [16, 17] has increasingly drawn attention due to its wide-ranging applications in numerous machine learning tasks, including hyper-parameter optimization [37], meta-learning [38] and neural architecture search [41]. A bilevel optimization problem involves an upper problem and a lower problem, wherein the upper problem is a function of the minimizer of the lower problem. Recently, great progress has been made to solve this type of problems, particularly through the development of efficient single-loop algorithms that rely on diverse gradient approximation techniques [12]. However, the majority of existing bilevel optimization research concentrates on standard, non-distributed settings, and how to solve the bilevel optimization problems under distributed settings have received much less attention. Federated learning (FL) [92] is a recently promising distributed learning paradigm. In FL, a set of clients jointly solve a machine learning task under the coordination of a central server. To protect user privacy and mitigate communication overhead, clients perform multiple steps of local update before communicating with the server. A variety of algorithms [93, 94, 95, 96, 97] have been proposed to accelerate

Table 4.1: **Comparisons of the Federated/Non-federated bilevel optimization algorithms for finding an ϵ-stationary point**. $Gc(f,\epsilon)$ and $Gc(g,\epsilon)$ denote the number of gradient evaluations *w.r.t.* $f^{(m)}(x,y)$ and $g^{(m)}(x,y)$; $JV(g,\epsilon)$ denotes the number of Jacobian-vector products; $HV(g,\epsilon)$ is the number of Hessian-vector products; $\kappa = L/\mu$ is the condition number, $p(\kappa)$ is used when no dependence is provided. Sample complexities are measured by client.

Setting	Algorithm	Communication	$Gc(f,\epsilon)$	$Gc(g,\epsilon)$	$JV(g,\epsilon)$	$HV(g,\epsilon)$	Heterogeneity
Non-Fed	StocBiO [98]		$O(\kappa^5\epsilon^{-2})$	$O(\kappa^9\epsilon^{-2})$	$O(\kappa^5\epsilon^{-2})$	$O(\kappa^6\epsilon^{-2})$	
	MRBO [32]		$O(p(\kappa)\epsilon^{-1.5})$	$O(p(\kappa)\epsilon^{-1.5})$	$O(p(\kappa)\epsilon^{-1.5})$	$O(p(\kappa)\epsilon^{-1.5})$	
Federated	CommFedBiO [99]	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	✓
	FedNest [100]	$O(\kappa^9\epsilon^{-2})$	$O(\kappa^5\epsilon^{-2})$	$O(\kappa^9\epsilon^{-2})$	$O(\kappa^5\epsilon^{-2})$	$O(\kappa^9\epsilon^{-2})$	✓
	AggITD [101]	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	$O(p(\kappa)\epsilon^{-2})$	✓
	FedMBO [102]	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	✓
	SimFBO [103]	$O(p(\kappa)\epsilon^{-1})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	$O(M^{-1}p(\kappa)\epsilon^{-2})$	✓
	Local-BSGVR [104]	$O(p(\kappa)\epsilon^{-1})$	$O(M^{-1}p(\kappa)\epsilon^{-1.5})$	$O(M^{-1}p(\kappa)\epsilon^{-1.5})$	$O(M^{-1}p(\kappa)\epsilon^{-1.5})$	$O(M^{-1}p(\kappa)\epsilon^{-1.5})$	✗
	FedBiOAcc (Ours)	$\boldsymbol{O(\kappa^{19/3}\epsilon^{-1})}$	$\boldsymbol{O(M^{-1}\kappa^8\epsilon^{-1.5})}$	$\boldsymbol{O(M^{-1}\kappa^8\epsilon^{-1.5})}$	$\boldsymbol{O(M^{-1}\kappa^8\epsilon^{-1.5})}$	$\boldsymbol{O(M^{-1}\kappa^8\epsilon^{-1.5})}$	✓

this training process. However, most of these algorithms primarily address standard single-level optimization problems. In this work, we study the bilevel optimization problems in the Federated Learning setting and investigate the following research question: *Is it possible to develop communication-efficient federated algorithms tailored for bilevel optimization problems that also ensure a rapid convergence rate?*

More specifically, a general Federated Bilevel Optimization problem has the following form:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq \frac{1}{M}\sum_{m=1}^{M} f^{(m)}(x, y_x), \text{ s.t. } y_x = \arg\min_{y\in\mathbb{R}^d} \frac{1}{M}\sum_{m=1}^{M} g^{(m)}(x,y) \tag{4.1}$$

A federated bilevel optimization problem consists of an upper and a lower level problem, the upper problem $f(x,y) \coloneqq \frac{1}{M}\sum_{m=1}^{M} f^{(m)}(x,y)$ relies on the solution y_x of the lower problem, and $g(x,y) \coloneqq \frac{1}{M}\sum_{m=1}^{M} g^{(m)}(x,y)$. Meanwhile, both the upper and the lower level problems are federated: In Eq.(4.1), we have M clients, and each client has a local upper problem $f^{(m)}(x,y)$ and a lower level problem $g^{(m)}(x,y)$. Compared to single-level federated optimization problems, the estimation of the hyper-gradient in federated bilevel optimiza-

tion problems is much more challenging. In Eq.(4.1), the hyper-gradient is not linear $w.r.t$ the local hyper-gradients of clients, whereas the gradient of a single-level Federated Optimization problem is the average of local gradients. Consequently, directly applying the vanilla local-sgd method [92] to federated bilevel problems results in a large bias. In the literature [99, 100, 101, 102], researchers evaluate the hyper-gradient through multiple rounds of client-server communication, however, this approach leads to high communication overhead. In contrast, we view the hyper-gradient estimation as solving a quadratic federated problem and solving it with the local-sgd method. More specifically, we formulate the solution of the federated bilevel optimization as three intertwined federated problems: the upper problem, the lower problem and the quadratic problem for the hyper-gradient estimation. Then we address the three problems using alternating gradient descent steps, furthermore, to manage the noise of the stochastic gradient and obtain the fast convergence rate, we employ a momentum-based variance reduction technique.

Beyond the standard federated bilevel optimization problem as defined in Eq. 4.1, another variant of Federated Bilevel Optimization problem, which entails locally managed lower-level problems, is also frequently utilized in practical applications. For this type of problem, we can get an unbiased estimate of the global hyper-gradient using local hyper-gradient, thus we can solve it with a local-SGD like algorithm, named FedBiOAcc-Local. However, it is challenging to analyze the convergence of the algorithm. In particular, we need to bound the intertwined client drift error, which is intrinsic to FL and the bilevel-related errors *e.g.* the lower level solution bias. In fact, we prove that the FedBiOAcc-Local algorithm attains the same fast rate as FedBiO algorithm.

Finally, we highlight the main **contributions** of our paper as follows:

- We propose FedBiOAcc to solve Federated Bilevel Optimization problems, the algorithm evaluates the hypergradient of federated bilevel optimization problems efficiently and achieves optimal convergence rate through momentum-based variance reduction. FedBiOAcc has sample complexity of $O(\epsilon^{-1.5})$, communication complexity of $O(\epsilon^{-1})$ and achieves linear speed-up $w.r.t$ the number of clients.

- We study Federated Bilevel Optimization problem with local lower level problem for the first time, where we show the convergence of a modified version of FedBiOAcc, named FedBiOAcc-Local for this type of problems.

- We validate the efficacy of the proposed FedBiOAcc algorithm through two real-world tasks: Federated Data Cleaning and Federated Hyper-representation Learning.

Notations ∇ denotes full gradient, ∇_x denotes partial derivative for variable x, higher order derivatives follow similar rules. $[K]$ represents the sequence of integers from 1 to K, $\bar{x}$ represents average of the sequence of variables $\{x^{(m)}\}_{m=1}^{M}$. $\bar{t}_s$ represents the global communication timestamp s.

4.2 Related Works

Bilevel optimization dates back to at least the 1960s when [16] proposed a regularization method, and then followed by many research works [8, 17, 18, 19], while in machine learning community, similar ideas in the name of implicit differentiation were also used in Hyper-parameter Optimization [20, 21, 22, 23]. Early algorithms for Bilevel Optimization solved the accurate solution of the lower problem for each upper variable. Recently, researchers developed algorithms that solve the lower problem with a fixed number

of steps, and use the 'back-propagation through time' technique to compute the hyper-gradient [24, 25, 26, 27, 28]. Very Recently, it witnessed a surge of interest in using implicit differentiation to derive single loop algorithms [6, 11, 12, 13, 14, 15, 32, 58, 65, 66, 105]. In particular, [58, 65] proposes a way to iteratively evaluate the hyper-gradients to save computation. In this work, we view the hyper-gradient estimation of Federated Bilevel Optimization as solving a quadratic federated optimization problem and use a similar iterative evaluation rule as [58, 65] in local update.

The bilevel optimization problem is also considered in the more general settings. For example, bilevel optimization with multiple lower tasks is considered in [106], furthermore, [107, 108, 109, 110] studies the bilevel optimization problem in the decentralized setting, [111] studies the bilevel optimization problem in the asynchronous setting. In contrast, we study bilevel optimization problems under Federated Learning [92] setting. Federated learning is a promising privacy-preserving learning paradigm for distributed data. Compared to traditional data-center distributed learning, Federated Learning poses new challenges including data heterogeneity, privacy concerns, high communication cost, and unfairness. To deal with these challenges, various methods [96, 112, 113, 114, 115, 116] are proposed. However, bilevel optimization problems are less investigated in the federated learning setting. [117] considered the distributed bilevel formulation, but it needs to communicate the Hessian matrix for every iteration, which is computationally infeasible. More recently, FedNest [100] has been proposed to tackle the general federated nest problems, including federated bilevel problems. However, this method evaluates the full hyper-gradient at every iteration; this leads to high communication overhead; furthermore, FedNest also uses SVRG to accelerate the training. Similar works that evaluate the hyper-gradient with

multiple rounds of client-server communication are [99, 101, 102, 103]. Finally, there is a concurrent work [104] that investigates the possibility of local gradients on Federated Bilevel Optimization, however, it only considers the homogeneous case, this setting is quite constrained and much simpler than the more general heterogeneous case we considered. Furthermore, [104] only considers the case where both the upper and the lower problem are federated, and omit the equally important case where the lower level problem is not federated.

4.3 Federated Bilevel Optimization

4.3.1 Some Mild Assumptions

Note that the formulation of Eq.(4.1) is very general, and we consider the stochastic heterogeneous case in this work. More specifically, we assume:

$$f^{(m)}(x,y) \coloneqq \mathbb{E}_{\xi\sim\mathcal{D}_f^{(m)}}[f^{(m)}(x,y,\xi)],\ g^{(m)}(x,y) \coloneqq \mathbb{E}_{\xi\sim\mathcal{D}_g^{(m)}}[g^{(m)}(x,y;\xi)]$$

where $\mathcal{D}_f^{(m)}$ and $\mathcal{D}_g^{(m)}$ are some probability distributions. Furthermore, we assume the local objectives could be potentially different: $f^{(m)}(x,y) \neq f^{(k)}(x,y)$ or $g^{(m)}(x,y) \neq g^{(k)}(x,y)$ for $m \neq k, m, k \in [M]$. Furthermore, we assume the following assumptions in our subsequent discussion:

Assumption 55. *Function $f^{(m)}(x,y)$ is possibly non-convex and $g^{(m)}(x,y)$ is μ-strongly convex* w.r.t *y for any given x.*

Assumption 56. *Function $f^{(m)}(x,y)$ is L-smooth and has C_f-bounded gradient;*

Assumption 57. *Function $g^{(m)}(x,y)$ is L-smooth, and $\nabla_{xy}g^{(m)}(x,y)$ and $\nabla_{y^2}g^{(m)}(x,y)$ are Lipschitz continuous with constants L_{xy} and L_{y^2} respectively;*

Assumption 58. *We have unbiased stochastic first-order and second-order gradient oracle with bounded variance.*

Assumption 59. *For any $m, j \in [M]$ and $z = (x,y)$, we have: $\|\nabla f^{(m)}(z) - \nabla f^{(j)}(z)\| \leq \zeta_f$, $\|\nabla g^{(m)}(z) - \nabla g^{(j)}(z)\| \leq \zeta_g$, $\|\nabla_{xy} g^{(m)}(z) - \nabla_{xy} g^{(j)}(z)\| \leq \zeta_{g,xy}$, $\|\nabla_{y^2} g^{(m)}(z) - \nabla_{y^2} g^{(j)}(z)\| \leq \zeta_{g,yy}$, where ζ_f, ζ_g, $\zeta_{g,xy}$, $\zeta_{g,yy}$, are constants.*

As stated in The assumption 55, we study the **non-convex-strongly-convex** bilevel optimization problems, this class of problems is widely studied in the non-distributed bilevel literature [6, 29]. Furthermore, Assumption 56 and Assumption 57 are also standard assumptions made in the non-distributed bilevel literature. Assumption 58 is widely used in the study of stochastic optimization problems. For Assumption 59, gradient difference is widely used in single level Federated Learning literature as a measure of client heterogeneity [13, 118]. Please refer to the full version of Assumptions in Appendix.

4.3.2 The FedBiOAcc Algorithm

A major difficulty in solving a Federated Bilevel Optimization problem Eq. (4.1) is **evaluating the hyper-gradient $\nabla h(x)$**. For the function class (non-convex-strongly-convex) we consider, the explicit form of hypergradient $h(x)$ exists as $\nabla h(x) = \Phi(x, y_x)$, where $\Phi(x,y)$ is denoted as:

$$\Phi(x,y) = \nabla_x f(x,y) - \nabla_{xy} g(x,y) \times [\nabla_{y^2} g(x,y)]^{-1} \nabla_y f(x,y), \tag{4.2}$$

Algorithm 7 Accelerated Federated Bilevel Optimization (**FedBiOAcc**)

Input: Constants c_ω, c_ν, c_u, γ, η, τ, r; learning rate schedule $\{\alpha_t\}$, $t \in [T]$, initial state (x_1, y_1, u_1);
Initialization: Set $y_1^{(m)} = y_1$, $x_1^{(m)} = x_1$, $u_1^{(m)} = u_1$, $\omega_1^{(m)} = \nabla_y g^{(m)}(x_1, y_1, \mathcal{B}_y)$, $\nu_1^{(m)} = \nabla_x f^{(m)}(x_1, y_1; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_1, y_1; \mathcal{B}_{g,1}) u_1$ and $q_1 = \nabla_{y^2} g^{(m)}(x_1^{(m)}, y_1^{(m)}; \mathcal{B}_{g,2}) u_1 - \nabla_y f^{(m)}(x_1^{(m)}, y_1^{(m)}; \mathcal{B}_{f,2})$ for $m \in [M]$
for $t = 1$ **to** T **do**
  $\hat{y}_{t+1}^{(m)} = y_t^{(m)} - \gamma \alpha_t \omega_t^{(m)}$, $\hat{x}_{t+1}^{(m)} = x_t^{(m)} - \eta \alpha_t \nu_t^{(m)}$, $\hat{u}_{t+1}^{(m)} = \mathcal{P}_r(u_t^{(m)} - \tau \alpha_t q_t^{(m)})$
  Get $\hat{\omega}_{t+1}^{(m)}$, $\hat{\nu}_{t+1}^{(m)}$ and $\hat{q}_{t+1}^{(m)}$ following Eq. (4.7)
  if t mod I $= 0$ **then**
    $y_{t+1}^{(m)} = \frac{1}{M} \sum_{j=1}^M \hat{y}_{t+1}^{(j)}$; $x_{t+1}^{(m)} = \frac{1}{M} \sum_{j=1}^M \hat{x}_{t+1}^{(j)}$, $u_{t+1}^{(m)} = \frac{1}{M} \sum_{j=1}^M \hat{u}_{t+1}^{(j)}$
    $\omega_{t+1}^{(m)} = \frac{1}{M} \sum_{j=1}^M \hat{\omega}_{t+1}^{(j)}$, $\nu_{t+1}^{(m)} = \frac{1}{M} \sum_{j=1}^M \hat{\nu}_{t+1}^{(j)}$, $q_{t+1}^{(m)} = \frac{1}{M} \sum_{j=1}^M \hat{q}_{t+1}^{(j)}$,
  else
    $y_{t+1}^{(m)} = \hat{y}_{t+1}^{(m)}$, $x_{t+1}^{(m)} = \hat{x}_{t+1}^{(m)}$, $u_{t+1}^{(m)} = \hat{u}_{t+1}^{(m)}$
    $\omega_{t+1}^{(m)} = \hat{\omega}_{t+1}^{(m)}$, $\nu_{t+1}^{(m)} = \hat{\nu}_{t+1}^{(m)}$, $q_{t+1}^{(m)} = \hat{q}_{t+1}^{(m)}$
  end if
end for

Based on Assumption 55∼57, we can verify $\Phi(x, y_x)$ is the hyper-gradient [6]. But since the clients only have access to their local data, for $\forall m \in [M]$, the client evaluates:

$$\Phi^{(m)}(x, y) = \nabla_x f^{(m)}(x, y) - \nabla_{xy} g^{(m)}(x, y) \times [\nabla_{y^2} g^{(m)}(x, y)]^{-1} \nabla_y f^{(m)}(x, y), \tag{4.3}$$

It is straightforward to verify that $\Phi^{(m)}(x, y)$ is not an unbiased estimate of the full hyper-gradient, *i.e.* $\Phi(x, y_x) \neq \frac{1}{M} \sum_{m=1}^M \Phi^{(m)}(x, y_x)$. To address this difficulty, we can view the **Hyper-gradient computation as the process of solving a federated optimization problem**.

In fact, Evaluating Eq. (4.2) is equivalent to the following two steps: first, we solve the quadratic federated optimization problem $l(u)$:

$$\min_{u \in \mathbb{R}^d} l(u) = \frac{1}{M} \sum_{m=1}^M u^T (\nabla_{y^2} g^{(m)}(x, y)) u - \langle \nabla_y f^{(m)}(x, y), u \rangle \tag{4.4}$$

Suppose that we denote the solution of the above problem as u^*, then we have the following linear operation to get the hypergradient:

$$\nabla h(x) = \frac{1}{M}\sum_{m=1}^{M}\left(\nabla_x f^{(m)}(x, y_x) - \nabla_{xy} g^{(m)}(x, y_x)u^*\right) \tag{4.5}$$

Compared to the formulation Eq. (4.2), Eq. (4.4) and Eq. (4.5) are more suitable for the distributed setting. In fact, both Eq. (4.4) and Eq. (4.5) have a linear structure. Eq. (4.4) is a (single-level) quadratic federated optimization problem, and we could solve Eq. (4.4) through local-sgd [92], suppose that each client maintains a variable $u_t^{(m)}$, and performs the following update:

$$u_{t+1}^{(m)} = \mathcal{P}_r(u_t^{(m)} - \tau_t \nabla l^{(m)}(u_t^{(m)}; \mathcal{B}))$$
$$\nabla l^{(m)}(u_t^{(m)}; \mathcal{B}) = \nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,2}))u_t^{(m)} - \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,2})$$

where $\nabla l^{(m)}(u_t^{(m)}; \mathcal{B})$ is client m's the stochastic gradient of Eq. (4.4), and $(x_t^{(m)}, y_t^{(m)})$ denotes the upper and lower variable state at the timestamp t, the $\mathcal{P}_r(\cdot)$ denotes the projection to a bounded ball of radius-r. Note that Clients perform multiple local updates of $u_t^{(m)}$ before averaging. As for Eq. (4.5), each client evaluates $\nabla h^{(m)}(x)$ locally: $\nabla h^{(m)}(x) = \nabla_x f^{(m)}(x, y_x) - \nabla_{xy} g^{(m)}(x, y_x)u^*$ and the server averages $\nabla h^{(m)}(x)$ to get $\nabla h(x)$. In summary, the linear structure of Eq. (4.4) and Eq. (4.5) makes it suitable for local updates, therefore, reduce the communication cost.

More specifically, we perform alternative update of upper level variable $x_t^{(m)}$, the lower level variable $y_t^{(m)}$ and hyper-gradient computation variable $u_t^{(m)}$. For example, for each

client $m \in [M]$, we perform the following local updates:

$$y_{t+1}^{(m)} = y_t^{(m)} - \gamma_t \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y),\ u_{t+1}^{(m)} = \mathcal{P}_r(u_t^{(m)} - \tau_t \nabla l^{(m)}(u_t^{(m)}; \mathcal{B}))$$
$$x_{t+1}^{(m)} = x_t^{(m)} - \eta_t\Big(\nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,1}) u_t^{(m)}\Big) \tag{4.6}$$

Every I steps, the server averages clients' local states, this resembles the local-sgd method for single level federated optimization problems. Note that in the update of the upper variable $x_t^{(m)}$, we use $u_t^{(m)}$ as an estimation of u^* in Eq. (4.5). An algorithm follows Eq. (4.6) is shown in Algorithm 2 of Appendix and we refer to it as FedBiO.

Comparison with FedNest. The update rule of Eq. 4.6 is very different from that of FedNest [100] and its follow-ups [101, 102]. In FedNest, a sub-routine named FedIHGP is used to evaluate Eq. (4.2) at every global epoch. This involves multiple rounds of client-server communication and leads to higher communication overhead. In contrast, Eq. (4.6) formulates the hyper-gradient estimation as an quadratic federated optimization problem, and then solves three intertwined federated problems through alternative updates of x, y and u.

Note that Eq. 4.6 updates the related variables through vanilla gradient descent steps. In the non-federated setting, gradient-based methods such as stocBiO [12] requires large-batch size ($O(\epsilon^{-1})$) to reach an ϵ-stationary point, and we also analyze Algorithm 2 in Appendix to show the same dependence. To control the noise and remove the dependence over large batch size, we apply the momentum-based variance-reduction technique STORM [30]. In fact, Eq. (4.6) solves three intertwined optimization problems: the bilevel problem $h(x)$, the lower level problem $g(x, y)$ and the hyper-gradient computation problem

Eq (4.4). So we control the noise in the process of solving each of the three problems. More specifically, we have $\omega_t^{(m)}$, $\nu_t^{(m)}$ and $q_t^{(m)}$ to be the momentum estimator for $x_t^{(m)}$, $y_t^{(m)}$ and $u_t^{(m)}$ respectively, and we update them following the rule of STORM [30]:

$$
\begin{aligned}
\hat{\omega}_{t+1}^{(m)} &= \nabla_y g^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}, \mathcal{B}_y) + (1 - c_\omega \alpha_t^2)(\omega_t^{(m)} - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y)) \\
\hat{\nu}_{t+1}^{(m)} &= \Big(\nabla_x f^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}; \mathcal{B}_{g,1}) u_{t+1}^{(m)}\Big) \\
&\quad + (1 - c_\nu \alpha_t^2)\Big(\nu_t^{(m)} - \Big(\nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,1}) u_t^{(m)}\Big)\Big) \\
\hat{q}_{t+1}^{(m)} &= \Big(\nabla_{y^2} g^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}; \mathcal{B}_{g,2}) u_{t+1}^{(m)} - \nabla_y f^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}; \mathcal{B}_{f,2})\Big) \\
&\quad + (1 - c_u \alpha_t^2)\Big(q_t^{(m)} - \Big(\nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,2}) u_t^{(m)} - \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,2})\Big)\Big)
\end{aligned}
\tag{4.7}
$$

where c_ω, c_ν and c_u are constants, α_t is the learning rate. Then we update the $x_t^{(m)}$, $y_t^{(m)}$ and $u_t^{(m)}$ as follows:

$$
\hat{y}_{t+1}^{(m)} = y_t^{(m)} - \gamma \alpha_t \omega_t^{(m)}, \hat{x}_{t+1}^{(m)} = x_t^{(m)} - \eta \alpha_t \nu_t^{(m)}, \hat{u}_{t+1}^{(m)} = \mathcal{P}_r(u_t^{(m)} - \tau \alpha_t q_t^{(m)}) \tag{4.8}
$$

where γ, η, τ are constants and α_t is the learning rate. The FedBiOAcc algorithm following Eq. (4.8) is summarized in Algorithm 7. As shown in line 6 and 12 of Algorithm 7, Every I iterations, we average both variables and the momentum.

4.3.3 Convergence Analysis

In this section, we study the convergence property for the FedBiOAcc algorithm. For any $t \in [T]$, we define the following virtual sequence:

$$\bar{x}_t = \frac{1}{M}\sum_{m=1}^{M} x_t^{(m)}, \bar{y}_t = \frac{1}{M}\sum_{m=1}^{M} y_t^{(m)}, \bar{u}_t = \frac{1}{M}\sum_{m=1}^{M} u_t^{(m)}$$

we denote the average of the momentum similarly as $\bar{\omega}_t$, $\bar{\nu}_t$ and $\bar{q}_t$. Then we consider the following Lyapunov function $\mathcal{G}_t$:

$$\begin{aligned}\mathcal{G}_t = h(\bar{x}_t) + \frac{18\eta\tilde{L}^2}{\mu\gamma}(\|\bar{y}_t - y_{\bar{x}_t}\|^2 + \|\bar{u}_t - u_{\bar{x}_t}\|^2) + \frac{9bM\eta}{64\alpha_t}\|\bar{\omega}_t - \frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\|^2 \\ + \frac{9bM\eta}{64\alpha_t}\|\bar{q}_t - \frac{1}{M}\sum_{m=1}^{M}(\nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)})u_t^{(m)} - \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}))\|^2 \\ + \frac{9bM\eta}{64\alpha_t}\|\bar{\nu}_t - \frac{1}{M}\sum_{m=1}^{M}(\nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)})u_t^{(m)})\|^2 \end{aligned} \tag{4.9}$$

where $y_{\bar{x}_t}$ denotes the solution of the lower level problem $g(\bar{x}_t, \cdot)$,

$$u_{\bar{x}_t} = [\nabla_{y^2} g(\bar{x}, y_{\bar{x}})]^{-1}\nabla_y f(\bar{x}, y_{\bar{x}})$$

denotes the solution of Eq (4.4) at state $\bar{x}_t$. Besides, γ, η, τ are learning rates and $L, \tilde{L}$ are constants. Note that the first three terms of $\mathcal{G}_t$: $h(\bar{x}_t)$, $\|\bar{y}_t - y_{\bar{x}_t}\|^2$, $\|\bar{u}_t - u_{\bar{x}_t}\|^2$ measures the errors of three federated problems: the upper level problem, the lower level problem and the hyper-gradient estimation. Then the last three terms measure the estimation error of the momentum variables: $\bar{\omega}_t$, $\bar{\nu}_t$ and $\bar{q}_t$. The convergence proof primarily concentrates

on bounding these errors, please see Lemma C.2 - C.6 in the Appendix for more details. Meanwhile, as in the single level federated optimization problems, local updates lead to client-drift error. More specifically, we need to bound $\|x_t^{(m)} - \bar{x}_t\|^2$, $\|y_t^{(m)} - \bar{y}_t\|^2$ and $\|u_t^{(m)} - \bar{u}_t\|^2$, please see Lemma C.7 - C.11 for more details. Finally, we have the following convergence theorem:

Theorem 60. *Suppose in Algorithm 7, we choose learning rate* $\alpha_t = \frac{\delta}{(u+t)^{1/3}}, t \in [T]$, *for some constant* δ *and* u, *and let* c_ν, c_ω, c_u *choose some value,* η, γ *and* τ, r *be some small values decided by the Lipschitz constants of* $h(x)$, *we choose the minibatch size to be* $b_x = b_y = b$ *and the first batch to be* $b_1 = O(Ib)$, *then we have:*

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] = O\Big(\frac{\kappa^{19/3}I}{T} + \frac{\kappa^{16/3}}{(bMT)^{2/3}}\Big)$$

To reach an ϵ*-stationary point, we need* $T = O(\kappa^8(bM)^{-1}\epsilon^{-1.5})$, $I = O(\kappa^{5/3}(bM)^{-1}\epsilon^{-0.5})$.

As stated in the Theorem, to reach an ϵ-stationary point, we need $T = O(\kappa^8(bM)^{-1}\epsilon^{-1.5})$, then the sample complexity for each client is $Gc(f,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$, $Gc(g,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$, $Jv(g,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$, $Hv(g,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$. So FedBiOAcc achieves the linear speed up *w.r.t.* to the number of clients M. Next, suppose we choose $I = O(\kappa^{5/3}(bM)^{-1}\epsilon^{-0.5})$, then the number of communication round $E = O(\kappa^{19/3}\epsilon^{-1})$. This matches the optimal communication complexity of the single level optimization problems as in the STEM [119]. Furthermore, compared to FedNest and its variants, FedBiOAcc has improved both the communication complexity and the iteration complexity. As for LocalBSCVR [104], FedBiOAcc obtains same rate, but incorporates the heterogeneous case. Note that it is much more challenging to analyze the heterogeneous case. In fact, if we

assume homogeneous clients, we have local hyper-gradient (Eq. (4.3)) equals the global hyper-gradient (Eq. (4.2)), then we do not need to use the quadratic federated optimization problem view in Section 3.2, while the theoretical analysis is also simplified significantly.

4.4 Federated Bilevel Optimization with Local Lower Level Problems

In this section, we consider an alternative formulation of the Federated Bilevel Optimization problems as follows:

$$\min_{x\in\mathbb{R}^p} h(x) := \frac{1}{M}\sum_{m=1}^{M} f^{(m)}(x, y_x^{(m)}),\ \text{s.t. } y_x^{(m)} = \arg\min_{y\in\mathbb{R}^d} g^{(m)}(x, y) \tag{4.10}$$

Same as Eq. (4.1), Eq. (4.10) has a federated upper level problem, however, Eq. (4.10) has a unique lower level problem for each client, which is different from Eq. (4.1). In fact, federated bilevel optimization problem Eq (4.10) can be viewed as a special type of standard federated learning problems. If we denote $h^{(m)}(x) = f^{(m)}(x, y_x^{(m)})$, then Eq. (4.10) can be written as $\min_{x\in\mathbb{R}^p} h(x) := \frac{1}{M}\sum_{m=1}^{M} h^{(m)}(x)$. But due to the bilevel structure of $h^{(m)}(x)$, Eq. (4.10) is more challenging than the standard Federated Learning problems.

Hyper-gradient Estimation. Assume Assumption 55$\sim$Assumption 57 hold, then the hyper-gradient is $\Phi(x, y_x) = \frac{1}{M}\sum_{m=1}^{M} \Phi^{(m)}(x, y_x)$, where $\Phi^{(m)}(x, y)$ is defined in Eq. (4.3), in other words, the local hyper-gradient $\Phi^{(m)}(x, y)$ is an unbiased estimate of the full hyper-gradient. This fact makes it possible to solve Eq. (4.10) with local-sgd like methods. More specifically, we solve the local bilevel problem $h^{(m)}(x)$ multiple steps on each client and then the server averages the local states from clients. Please refer to Algorithm 3 and the variance-reduction acceleration Algorithm 4 in the Appendix. For ease of reference, we

name them FedBiO-Local and FedBiOAcc-Local, respectively.

Several challenges exist in analyzing FedBiO-Local and FedBiOAcc-Local. First, Eq. (4.3) involves Hessian inverse, so we only evaluate it approximately through the Neumann series [120] as:

$$\Phi^{(m)}(x,y;\xi_x) = \nabla_x f^{(m)}(x,y;\xi_f) - \tau\nabla_{xy} g^{(m)}(x,y;\xi_g) \\ \times \sum_{q=-1}^{Q-1} \prod_{j=Q-q}^{Q} (I - \tau\nabla_{y^2} g^{(m)}(x,y;\xi_j))\nabla_y f^{(m)}(x,y;\xi_f) \tag{4.11}$$

where $\xi_x = \{\xi_j(j=1,\ldots,Q), \xi_f, \xi_g\}$, and we assume its elements are mutually independent. $\Phi^{(m)}(x,y;\xi_x)$ is a biased estimate of $\Phi^{(m)}(x,y)$, but with bounded bias and variance (Please see Proposition D.2 for more details.) Furthermore, to reduce the computation cost, each client solves the local lower level problem approximately and we update the upper and lower level variable alternatively. The idea of alternative update is widely used in the non-distributed bilevel optimization [12, 32]. However, in the federated setting, client variables drift away when performing multiple local steps. As a result, the variable drift error and the bias caused by inexact solution of the lower level problem intertwined with each other. For example, in the local update, clients optimize the lower level variable $y^{(m)}$ towards the minimizer $y^{(m)}_{x^{(m)}}$, but after the communication step, $x^{(m)}$ is smoothed among clients, as a result, the target of $y^{(m)}_t$ changes which causes a huge bias.

In the appendix, we show the FedBiOAcc-Local algorithm achieves the same optimal convergence rate as FedBiOAcc, which has iteration complexity $O(\epsilon^{-1.5})$ and communication complexity $O(\epsilon^{-1})$. However, since the lower level problem in Eq. (4.10) is unique for each client, FedBiOAcc-Local does not have the property of linear speed-up *w.r.t* the

number of clients as FedBiOAcc does.

4.5 Numerical Experiments

In this section, we assess the performance of the proposed FedBiOAcc algorithm through two federated bilevel tasks: Federated Data Cleaning and Federated Hyper Representation Learning. The Federated Data Cleaning task involves global lower level problems, while the Hyper-representation Learning task involves local lower level problems. The implementation is carried out using PyTorch, and the Federated Learning environment is simulated using the PyTorch.Distributed package. Our experiments were conducted on servers equipped with an AMD EPYC 7763 64-core CPU and 8 NVIDIA V100 GPUs.

4.5.1 Federated Data Cleaning

In this section, we consider the Federated Data Cleaning task. In this task, we are given a noisy training dataset whose labels are corrupted by noise and a clean validation set. Then we aim to find weights for training samples such that a model that is learned over the weighted training set performs well on the validation set. This is a federated bilevel problem when the noisy training set is distributed over multiple clients. The formulation of the task is included in Appendix B.1. This task is a specialization of Eq. (4.1).

Dataset and Baselines. We create 10 clients and construct datasets based on MNIST [85]. For the training set, each client randomly samples 4500 images (no overlap among clients) from 10 classes and then randomly uniformly perturb the labels of ρ ($0 \leq \rho \leq 1$) percent samples. For the validation set, each client randomly selects 50 clean images

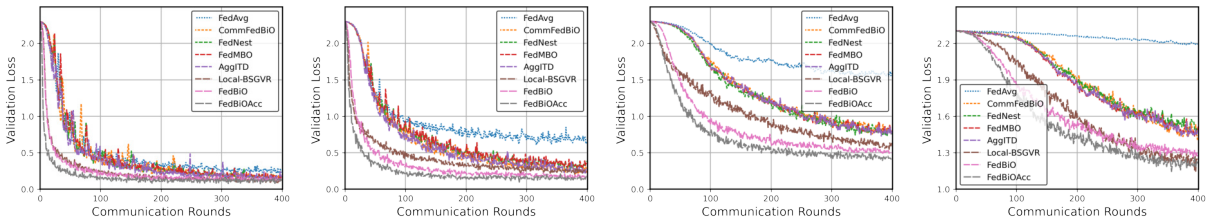


Figure 4.1: Validation Error vs Communication Rounds. From Left to Right: $\rho = 0.1, 0.4, 0.8, 0.95$. The local step I is set as 5 for FedBiO, FedBiOAcc and FedAvg.

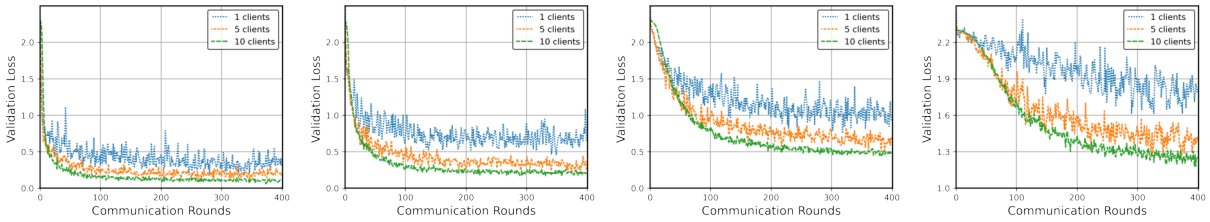


Figure 4.2: Validation Error vs Communication Rounds with different number of clients per epoch. From Left to Right: $\rho = 0.1, 0.4, 0.8, 0.95$. The local step I is set as 5.

from a different class. In other words, the m_{th} client only has validation samples from the m_{th} class. This single-class validation setting introduces a high level of heterogeneity, such that individual clients are unable to conduct local cleaning due to they only have clean samples from one class. In our experiments, we test our FedBiOAcc algorithm, including the FedBiO algorithm (Algorithm 2 in Appendix) which does not use variance reduction; additionally, we also consider some baseline methods: a baseline that directly performs FedAvg [92] on the noisy dataset, this helps to verify the usefulness of data cleaning; Local-BSGVR [104], FedNest [100], CommFedBiO [99], AggITD [101] and FedMBO [102]. Note that Local-BSGVR is designed for the homogeneous setting, and the last four baselines all need multiple rounds of client-server communication to evaluate the hyper-gradient at each global epoch. We perform grid search to find the best hyper-parameters for each method and report the best results. Specific choices are included in Appendix B.1.

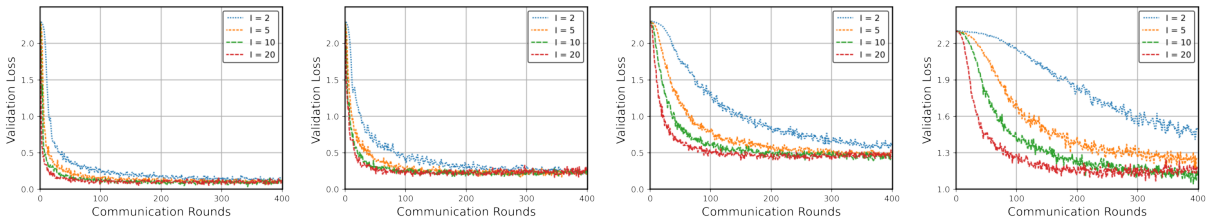


Figure 4.3: Validation Error vs Communication Rounds with different number of local steps I. From Left to Right: $\rho = 0.1, 0.4, 0.8, 0.95$.

In figure 4.1, we compare the performance of different methods at various noise levels ρ. Note that the larger the ρ value, the more noisy the training data are. The noise level can be illustrated by the performance of the FedAvg algorithm, which learns over the noisy data directly. As shown in the figure, FedAvg learns almost nothing when $\rho = 0.95$. Next, our algorithms are robust under various heterogeneity levels. When the noise level in the training set increases as the value of ρ increases, learning relies more on the signal from the heterogeneous validation set, and our algorithms consistently outperform other baselines. Finally, in figure 4.2, we vary the number of clients sampled per epoch, and the experimental results show that our FedBiOAcc converges faster with more clients in the training per epoch; in figure 4.3, we vary the number of local steps under different noisy levels. Interestingly, the algorithm benefit more from the local training under larger noise.

4.5.2 Federated Hyper-Representation Learning

In the Hyper-representation learning task, we learn a hyper-representation of the data such that a linear classifier can be learned quickly with a small number of data samples. A mathematical formulation of the task is included in Appendix B.2. Note that this task is an instantiation of Eq. (4.10), due to the fact that each client has its own tasks, and thus only

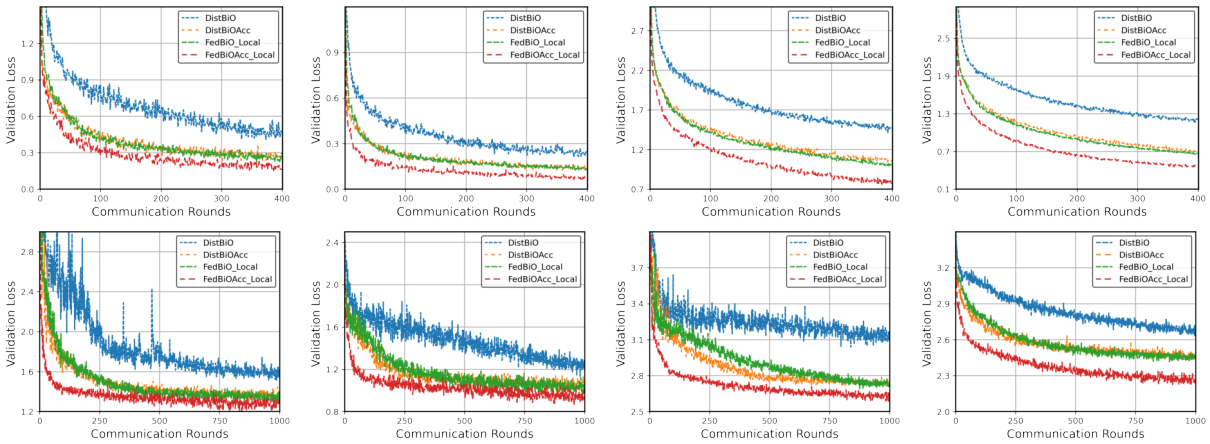


Figure 4.4: Validation Error vs Communication Rounds. The top row shows the result for the Omniglot Dataset and the bottom row shows MiniImageNet. From Left to Right: 5-way-1-shot, 5-way-5-shot, 20-way-1-shot, 20-way-5-shot. The local step I is set to 5.

the upper level problem is federated. We consider the Omniglot [87] and MiniImageNet [88] data sets. As in the non-distributed setting, we perform N-way-K-shot classification.

In this experiment, we compare FedBiOAcc-Local (Algorithm 4 in the Appendix) with three baselines FedBiO-Local (Algorithm 3 in the Appendix), DistBiO and DistBiOAcc. Note that DistBiO and DistBiOAcc are the distributed version of FedBiO-Local and FedBiOAcc-Local, respectively. In the experiments, we implement DistBiO and DistBiOAcc by setting the local steps as 1 for FedBiO-Local and FedBiOAcc-Local. We perform grid search for the hyper-parameter selection for both methods and choose the best ones, the specific choices of hyper-parameters are deferred to Appendix B.2. The results are summarized in Figure 4.4 (full results are included in Figure 5 and Figure 6 of Appendix. As shown by the results, FedBiOAcc converges faster than the baselines on both datasets and on all four types of classification tasks, which demonstrates the effectiveness of variance reduction and multiple steps of local training.

4.6 Assumptions

In this section, we restate all assumptions needed in our proof below:

Assumption 61 (Assumption 1)**.** *The function $f^{(m)}(x,y)$ is possibly non-convex and $g^{(m)}(x,y)$ is μ-strongly convex* w.r.t *y for any given x,* i.e. *for any y_1, $y_2 \in \mathbb{R}^d$, we have:*

$$g^{(m)}(x,y_1) \geq g^{(m)}(x,y_2) + \langle \nabla_y g^{(m)}(x,y_2), y_2 - y_1 \rangle + \frac{\mu}{2}||y_2 - y_1||^2.$$

Assumption 62 (Assumption 2)**.** *Function $f^{(m)}(x,y)$ is L-Lipschitz,* i.e. *for for any x_1, $x_2 \in \mathcal{X}$ and for any y_1, $y_2 \in \mathbb{R}^d$, and we denote $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$, then we have:*

$$f^{(m)}(z_1) \leq f^{(m)}(z_2) + \langle \nabla f^{(m)}(z_2), z_1 - z_2 \rangle + \frac{L}{2}||z_1 - z_2||^2.$$

or equivalently: $||\nabla f^{(m)}(z_1) - \nabla f^{(m)}(z_2)|| \leq L||z_1 - z_2||$. We also assume and $f^{(m)}(x,y)$ has C_f-bounded gradient, i.e. *for for any $x \in \mathcal{X}$ and any $y \in \mathbb{R}^d$, and we denote $z = (x,y)$, then we have $||\nabla f(z)|| \leq C_f$.*

Assumption 63 (Assumption 3)**.** *Function $g^{(m)}(x,y)$ is L-Lipschitz.* i.e. *for for any x_1, $x_2 \in \mathcal{X}$ and for any y_1, $y_2 \in \mathbb{R}^d$, and we denote $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$, then we have:*

$$g^{(m)}(z_1) \leq g^{(m)}(z_2) + \langle \nabla g^{(m)}(z_2), z_1 - z_2 \rangle + \frac{L}{2}||z_1 - z_2||^2.$$

equivalently: $||\nabla g^{(m)}(z_1) - \nabla g^{(m)}(z_2)|| \leq L||z_1 - z_2||$. For higher-order derivatives, we have:

a) $\nabla_{xy} g^{(m)}(x,y)$ and $\nabla_{y^2} g^{(m)}(x,y)$ are Lipschitz continuous with constant L_{xy} and L_{y^2}

respectively, i.e. *for for any* $x_1,\ x_2 \in \mathcal{X}$ *and for any* $y_1,\ y_2 \in \mathbb{R}^d$, *and we denote* $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$, *then we have:* $||\nabla_{xy} g^{(m)}(z_1) - \nabla_{xy} g^{(m)}(z_2)|| \leq L_{xy}||z_1 - z_2||$ *and* $||\nabla_{y^2} g^{(m)}(z_1) - \nabla_{y^2} g^{(m)}(z_2)|| \leq L_{y^2}||z_1 - z_2||$.

Assumption 64 (Assumption 4)**.** *We have an unbiased stochastic first order and second order derivative oracle with bounded variance, more specifically, denote* $z = (x, y)$, *we have:*

a) we have $\nabla f^{(m)}(z;\xi)$, *such that:* $E[\nabla f^{(m)}(z;\xi)] = \nabla f^{(m)}(z)$ *and* $var(\nabla f^{(m)}(z;\xi)) \leq \sigma^2$.

b) we have $\nabla g^{(m)}(z;\xi)$, *such that:* $E[\nabla g^{(m)}(z;\xi)] = \nabla g^{(m)}(z)$ *and* $var(\nabla g^{(m)}(z;\xi)) \leq \sigma^2$.

c) we have $\nabla_{y^2} g^{(m)}(z,\xi)$, *such that:* $E[\nabla_{y^2} g^{(m)}(z;\xi)] = \nabla_{y^2} g^{(m)}(z)$ *and* $var(\nabla_{y^2} g^{(m)}(z;\xi)) \leq \sigma^2$;

d) we have $\nabla_{xy} g^{(m)}(z;\xi)$, *such that:* $E[\nabla_{xy} g^{(m)}(z;\xi)] = \nabla_{xy} g^{(m)}(z)$ *and* $var(\nabla_{xy} g^{(m)}(x,y;\xi)) \leq \sigma^2$;

Assumption 65 (Assumption 5)**.** *For any* $m, j \in [M]$ *and* $z = (x, y)$, *we have:* $\|\nabla f^{(m)}(z) - \nabla f^{(j)}(z)\| \leq \zeta_f$, $\|\nabla g^{(m)}(z) - \nabla g^{(j)}(z)\| \leq \zeta_g$, $\|\nabla_{xy} g^{(m)}(z) - \nabla_{xy} g^{(j)}(z)\| \leq \zeta_{g,xy}$, $\|\nabla_{y^2} g^{(m)}(z) - \nabla_{y^2} g^{(j)}(z)\| \leq \zeta_{g,yy}$, *where* ζ_f, ζ_g, $\zeta_{g,xy}$, $\zeta_{g,yy}$, *are constants.*

Assumption 66 (Assumption 6)**.** *For any* $m, j \in [M]$ *and* $z = (x, y)$, *we have:* $\|\nabla f^{(m)}(z) - \nabla f^{(j)}(z)\| \leq \zeta_f$, $\|\nabla_{xy} g^{(m)}(z) - \nabla_{xy} g^{(j)}(z)\| \leq \zeta_{g,xy}$, $\|\nabla_{y^2} g^{(m)}(z) - \nabla_{y^2} g^{(j)}(z)\| \leq \zeta_{g,yy}$, $\|y_x^{(m)} - y_x^{(j)}\| \leq \zeta_{g^*}$, *where* ζ_f, $\zeta_{g,xy}$, $\zeta_{g,yy}$, ζ_{g^*} *are constants.*

4.7 Proof for Global Lower Level Problem

This section includes proofs related to the Federated Bilevel Optimization problems with global lower level problems (Eq. 4.1). First, we have the global and local hypergradient $\nabla h(x) = \Phi(x, y_x)$, $\nabla h^{(m)}(x) = \Phi^{(m)}(x, y_x)$ as defined in Eq. 4.2 and Eq. 4.3, and the following proposition:

Proposition 67. *Suppose Assumptions 56 and 57 hold, the following statements hold:*

a) y_x *is Lipschitz continuous in* x *with constant* $\rho = \kappa$*, where* $\kappa = \frac{L}{\mu}$ *is the condition number of* $g(x, y)$*.*

b) $\|\Phi(x_1; y_1) - \Phi(x_2; y_2)\|^2 \leq \hat{L}^2(\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2)$*, where* $\hat{L} = O(\kappa^2)$*.*

c) $h(x)$ *is Lipschitz continuous in* x *with constant* $\bar{L}$ *i.e., for any given* $x_1, x_2 \in X$*, we have* $\|\nabla h(x_2) - \nabla h(x_1)\| \leq \bar{L}\|x_2 - x_1\|$ *where* $\bar{L} = O(\kappa^3)$*.*

This is a standard results in bilevel optimization and we omit the proof here.

4.7.1 Proof for the FedBiOAcc Algorithm

In this section, we prove the convergence of the FedBiOAcc Algorithm. To simplify the notation, we denote

$$\mu_{t,\xi}^{(m)} = \nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_{f,1}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_{g,1}) u_t^{(m)},$$

and we have:

$$\mathbb{E}_{\xi}[\mu_{t,\xi}^{(m)}] = \nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}) u_t^{(m)}$$

where the expectation is *w.r.t* $\{\xi_{f,1}, \xi_{g,1}\}$ at iteration t, we denote $\mu_t^{(m)} = \mathbb{E}_{\xi}[\mu_{t,\xi}^{(m)}]$ for short. Similarly, we denote

$$p_{t,\xi}^{(m)} = \nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_{g,2}) u_t^{(m)} + \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_{f,2}),$$

and we have:

$$\mathbb{E}_{\xi}[p_{t,\xi}^{(m)}] = \nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}) u_t^{(m)} + \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}).$$

where the expectation is *w.r.t* $\{\xi_{f,2}, \xi_{g,2}\}$ at iteration t, we denote $p_t^{(m)} = \mathbb{E}_{\xi}[p_{t,\xi}^{(m)}]$ for short.

4.7.1.1 Hyper-Gradient Bias and Inner-Gradient Bias

Lemma 68. *Suppose we have* $c_u \alpha_t^2 < 1$*, then we have:*

$$\begin{aligned}
\mathbb{E}\Big[\Big\|\bar{q}_t - \bar{p}_t\Big\|^2\Big] \leq (1 - c_u\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{q}_{t-1} - \bar{p}_{t-1}\Big\|^2\Big] + \frac{2(c_u\alpha_{t-1}^2)^2}{b_x M}\sigma^2 \\
+ \frac{4\tilde{L}_2^2}{b_x M^2}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big] \\
+ \frac{8L^2}{b_x M^2}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|u_t^{(m)} - u_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}$$

where $\tilde{L}_2^2 = \Big(L^2 + \frac{2L_{y^2}^2 C_f^2}{\mu^2}\Big)$ *and the expectation outside is* w.r.t *all the stochasity of the algorithm.*

Proof. First, we have:

$$
\begin{aligned}
&\mathbb{E}\Big[\Big\|\bar{q}_t - \bar{p}_t]\Big\|^2\Big] = \mathbb{E}\Big[\Big\|\bar{p}_{t,\mathcal{B}_x} + (1 - c_u\alpha_{t-1}^2)(\bar{q}_{t-1} - \bar{p}_{t-1,\mathcal{B}_x}) - \bar{p}_t\Big\|^2\Big] \\
&= \mathbb{E}\Big[\Big\|(1 - c_u\alpha_{t-1}^2)\Big(\bar{q}_{t-1} - \bar{p}_{t-1}\Big) + \Big(\bar{p}_{t,\mathcal{B}_x} - \bar{p}_t + (1 - c_u\alpha_{t-1}^2)(\bar{p}_{t-1} - \bar{p}_{t-1,\mathcal{B}_x})\Big)\Big\|^2\Big] \\
&\leq (1 - c_u\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{q}_{t-1} - \bar{p}_{t-1}\Big\|^2\Big] + \mathbb{E}\Big[\Big\|\bar{p}_{t,\mathcal{B}_x} - \bar{p}_t + (1 - c_u\alpha_{t-1}^2)(\bar{p}_{t-1} - \bar{p}_{t-1,\mathcal{B}_x})\Big\|^2\Big] \\
&\leq (1 - c_u\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{q}_{t-1} - \bar{p}_{t-1}\Big\|^2\Big] \\
&\quad + \frac{1}{b_x^2 M^2}\sum_{m=1}^{M}\sum_{\xi_x \in \mathcal{B}_x}\mathbb{E}\Big[\Big\|p_{t,\xi_x}^{(m)} - p_t^{(m)} + (1 - c_u\alpha_{t-1}^2)(p_{t-1}^{(m)} - p_{t-1,\xi_x}^{(m)})\Big\|^2\Big]
\end{aligned}
$$

where the first inequality uses the fact that the cross product term is zero in expectation, the condition that $c_\nu\alpha_t^2 < 1$ and the second inequality follows that samples are independent among clients. We denote the second term of above as T_1, then we have:

$$
\begin{aligned}
T_1 &\overset{(a)}{\leq} 2(c_u\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\|p_{t,\xi_x}^{(m)} - p_t^{(m)}\Big\|^2\Big] + 2(1 - c_u\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\|p_{t,\xi_x}^{(m)} - p_{t-1,\xi_x}^{(m)} - (p_t^{(m)} - p_{t-1}^{(m)})\Big\|^2\Big] \\
&\overset{(b)}{\leq} 2(c_u\alpha_{t-1}^2)^2\sigma^2 + 2\mathbb{E}\Big[\Big\|p_{t,\xi_x}^{(m)} - p_{t-1,\xi_x}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

where inequality (a) follows the generalized triangle inequality; (b) and the bounded variance assumption. We denote the second term above as $T_{1,2}$, we have:

$$
\begin{aligned}
T_{1,2} = 2\mathbb{E}\Big\|&\nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_{g,2})u_t^{(m)} + \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_{f,2}) \\
&- \Big(\nabla_{y^2} g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \xi_{g,2})u_{t-1}^{(m)} + \nabla_y f^{(m)}(x_t^{(m)}, y_{t-1}^{(m)}; \xi_{f,2})\Big)\Big\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq 4\mathbb{E}\Big\|\nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,1}) - \nabla_y f^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \mathcal{B}_{f,1})\Big\|^2 \\
&\quad + 4\mathbb{E}\Big\|\nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,1}) u_t^{(m)} - \nabla_{y^2} g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \mathcal{B}_{g,1}) u_{t-1}^{(m)}\Big\|^2 \\
&\leq 4\Big(L^2 + \frac{2L_{y^2}^2 C_f^2}{\mu^2}\Big)\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big] + 8L^2\mathbb{E}\Big[\Big\|u_t^{(m)} - u_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

Combine everything together finishes the proof. □

Lemma 69. *Suppose we have $c_\nu \alpha_t^2 < 1$, then we have:*

$$
\begin{aligned}
\mathbb{E}\Big[\Big\|\bar{\nu}_t - \bar{\mu}_t\Big\|^2\Big] \leq (1 - c_\nu\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{\nu}_{t-1} - \bar{\mu}_{t-1}\Big\|^2\Big] + \frac{2(c_\nu\alpha_{t-1}^2)^2}{b_x M}\sigma^2 \\
+ \frac{4\tilde{L}_1^2}{b_x M^2}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big] \\
+ \frac{8L^2}{b_x M^2}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|u_t^{(m)} - u_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

where $\tilde{L}_1^2 = \Big(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2}\Big)$ and the expectation outside is w.r.t *all the stochasity of the algorithm.*

Proof. First, we have:

$$
\begin{aligned}
&\mathbb{E}\Big[\Big\|\bar{\nu}_t - \bar{\mu}_t]\Big\|^2\Big] = \mathbb{E}\Big[\Big\|\bar{\mu}_{t,\mathcal{B}_x} + (1 - c_\nu\alpha_{t-1}^2)(\bar{\nu}_{t-1} - \bar{\mu}_{t-1,\mathcal{B}_x}) - \bar{\mu}_t\Big\|^2\Big] \\
&= \mathbb{E}\Big[\Big\|(1 - c_\nu\alpha_{t-1}^2)\Big(\bar{\nu}_{t-1} - \bar{\mu}_{t-1}\Big) + \Big(\bar{\mu}_{t,\mathcal{B}_x} - \bar{\mu}_t + (1 - c_\nu\alpha_{t-1}^2)(\bar{\mu}_{t-1} - \bar{\mu}_{t-1,\mathcal{B}_x})\Big)\Big\|^2\Big] \\
&\leq (1 - c_\nu\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{\nu}_{t-1} - \bar{\mu}_{t-1}\Big\|^2\Big] + \mathbb{E}\Big[\Big\|\bar{\mu}_{t,\mathcal{B}_x} - \bar{\mu}_t + (1 - c_\nu\alpha_{t-1}^2)(\bar{\mu}_{t-1} - \bar{\mu}_{t-1,\mathcal{B}_x})\Big\|^2\Big] \\
&\leq (1 - c_\nu\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{\nu}_{t-1} - \bar{\mu}_{t-1}\Big\|^2\Big] \\
&\quad + \frac{1}{b_x^2 M^2}\sum_{m=1}^{M}\sum_{\xi_x \in \mathcal{B}_x}\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mu_t^{(m)} + (1 - c_\nu\alpha_{t-1}^2)(\mu_{t-1}^{(m)} - \mu_{t-1,\xi_x}^{(m)})\Big\|^2\Big]
\end{aligned}
$$

where the first inequality uses the fact that the cross product term is zero in expectation,

the condition that $c_\nu \alpha_t^2 < 1$ and the second inequality follows that samples are independent among clients. We denote the second term of above as T_1, then we have:

$$
\begin{aligned}
T_1 &\overset{(a)}{\leq} 2(c_\nu \alpha_{t-1}^2)^2 \mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mu_t^{(m)}\Big\|^2\Big] + 2(1 - c_\nu \alpha_{t-1}^2)^2 \mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mu_{t-1,\xi_x}^{(m)} - (\mu_t^{(m)} - \mu_{t-1}^{(m)})\Big\|^2\Big] \\
&\overset{(b)}{\leq} 2(c_\nu \alpha_{t-1}^2)^2 \sigma^2 + 2\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mu_{t-1,\xi_x}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

where inequality (a) follows the generalized triangle inequality; (b) and the bounded variance assumption. We denote the second term above as $T_{1,2}$, we have:

$$
\begin{aligned}
T_{1,2} &= 2\mathbb{E}\Big\|\nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,1}) u_t^{(m)} \\
&\qquad - \Big(\nabla_x f^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \mathcal{B}_{g,1}) u_{t-1}^{(m)}\Big)\Big\|^2 \\
&\leq 4\mathbb{E}\Big\|\nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,1}) - \nabla_x f^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \mathcal{B}_{f,1})\Big\|^2 \\
&\qquad + 4\mathbb{E}\Big\|\nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,1}) u_t^{(m)} - \nabla_{xy} g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}; \mathcal{B}_{g,1}) u_{t-1}^{(m)}\Big\|^2 \\
&\leq 4\Big(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2}\Big)\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big] + 8L^2\mathbb{E}\Big[\Big\|u_t^{(m)} - u_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

Combine everything together finishes the proof. □

Lemma 70. *Suppose we have $c_\omega \alpha_{t-1}^2 < 1$, then for $t \neq \bar{t}_s$, with $s \in [S]$, we have:*

$$
\begin{aligned}
&\mathbb{E}\Big[\Big\|\bar{\omega}_t - \frac{1}{M}\sum_{m=1}^M \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2\Big] \\
&\leq (1 - c_\omega \alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{\omega}_{t-1} - \frac{1}{M}\sum_{m=1}^M \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)})\Big\|^2\Big] + \frac{2(c_\omega \alpha_{t-1}^2)^2 \sigma^2}{b_y M} \\
&\qquad + \frac{2L^2}{b_y M^2}\sum_{m=1}^M \mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. First, we have:

$$
\begin{aligned}
&\mathbb{E}\Big[\Big\|\bar{\omega}_t - \frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2\Big] \\
&= \mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{m=1}^{M}\Big(\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y) \\
&\qquad + (1-c_\omega\alpha_{t-1}^2)(\omega_{t-1}^{(m)} - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \mathcal{B}_y)) - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big)\Big\|^2\Big] \\
&= \mathbb{E}\Big[\Big\|(1-c_\omega\alpha_{t-1}^2)(\bar{\omega}_{t-1} - \frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) \\
&\qquad + \frac{1}{M}\sum_{m=1}^{M}\Big(\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y) - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}) \\
&\qquad\quad + (1-c_\omega\alpha_{t-1}^2)(\nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \mathcal{B}_y))\Big)\Big\|^2\Big] \\
&\overset{(a)}{\leq} (1-c_\omega\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{\omega}_{t-1} - \frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)})\Big\|^2\Big] \\
&\qquad + \frac{1}{b_y^2 M^2}\sum_{m=1}^{M}\mathbb{E}\sum_{\xi_y\in\mathcal{B}_y}\Big[\Big\|\Big(\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \xi_y) - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}) \\
&\qquad + (1-c_\omega\alpha_{t-1}^2)(\nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \xi_y))\Big)\Big\|^2\Big]
\end{aligned}
$$

where inequality (a) uses the fact that the cross product term is zero in expectation and the condition that $c_\omega\alpha_t^2 < 1, t \in [T]$, furthermore, the samples are sampled independently on clients.

We denote the second term in the above inequality as T_1, we have:

$$
\begin{aligned}
T_1 &\overset{(b)}{\leq} 2(c_\omega\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\|\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \xi_y) - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2\Big] \\
&\quad + 2(1-c_\omega\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\| - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}) \\
&\quad + \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \xi_y) + \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \xi_y)\Big\|^2\Big]
\end{aligned}
$$

$$\overset{(c)}{\leq} 2(c_\omega \alpha_{t-1}^2)^2\sigma^2 + 2\mathbb{E}\Big[\Big\|\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \xi_y) - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \xi_y)\Big\|^2\Big]$$
$$\overset{(d)}{\leq} 2(c_\omega \alpha_{t-1}^2)^2\sigma^2 + 2L^2\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big]$$

inequality (b) uses the generalized triangle inequality; inequality (c) follows the bounded variance assumption 58, Proposition 111; inequality (d) uses the smoothness assumption 57.

□

4.7.1.2 Lower Problem Solution Error

Lemma 71. *Suppose we choose $\gamma \leq \frac{1}{2L}$ and $\alpha_t < 1$. Then for $t \in [T]$, we have:*

$$\begin{aligned}\|\bar{y}_{t+1} - y_{\bar{x}_{t+1}}\|^2 \leq \Big(1 - \frac{\mu\gamma\alpha_t}{4}\Big)\|\bar{y}_t - y_{\bar{x}_t}\|^2 - \frac{\gamma^2\alpha_t}{4}\|\bar{\omega}_t\|^2 + \frac{9\kappa^2\eta^2\alpha_t}{2\mu\gamma}\|\bar{\nu}_t\|^2 \\ + \frac{9\gamma\alpha_t L^2}{\mu M}\sum_{m=1}^{M}\Big[\|x_t^{(m)} - \bar{x}_t\|^2 + \|y_t^{(m)} - \bar{y}_t\|^2\Big] \\ + \frac{9\gamma\alpha_t}{\mu}\|\frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}) - \bar{w}_t\|^2\end{aligned}$$

Proof. First, we exploit Proposition 114, and choose the function $g(\bar{x}_t, \cdot)$, by assumption it is L smooth and μ strongly convex, and we choose $\gamma < \frac{1}{2L}$ and $\alpha_t < 1$, thus:

$$\|\bar{y}_{t+1} - y_{\bar{x}_t}\|^2 \leq (1 - \frac{\mu\gamma\alpha_t}{2})\|\bar{y}_t - y_{\bar{x}_t}\|^2 - \frac{\gamma^2\alpha_t}{4}\|\bar{\omega}_t\|^2 + \frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(\bar{x}_t, \bar{y}_t) - \bar{w}_t\|^2. \quad (4.12)$$

Next, we decompose the term $\|\bar{y}_{t+1} - y_{\bar{x}_{t+1}}\|^2$ as follows:

$$\|\bar{y}_{t+1} - y_{\bar{x}_{t+1}}\|^2 \leq (1 + \frac{\mu\gamma\alpha_t}{4})\|\bar{y}_{t+1} - y_{\bar{x}_t}\|^2 + (1 + \frac{4}{\mu\gamma\alpha_t})\|y_{\bar{x}_t} - y_{\bar{x}_{t+1}}\|^2$$

$$\leq (1+\frac{\mu\gamma\alpha_t}{4})\|\bar{y}_{t+1}-y_{\bar{x}_t}\|^2+(1+\frac{4}{\mu\gamma\alpha_t})\kappa^2\|\bar{x}_t-\bar{x}_{t+1}\|^2 \tag{4.13}$$

where the second inequality is due to case a) of Proposition 3.9. Combining the above inequalities 4.12 and 4.13, we have

$$\begin{aligned}\|\bar{y}_{t+1}-y_{\bar{x}_{t+1}}\|^2 \leq{}& (1+\frac{\mu\gamma\alpha_t}{4})(1-\frac{\mu\gamma\alpha_t}{2})\|\bar{y}_t-y_{\bar{x}_t}\|^2-(1+\frac{\mu\gamma\alpha_t}{4})\frac{\gamma^2\alpha_t}{4}\|\bar{\omega}_t\|^2\\ &+(1+\frac{\mu\gamma\alpha_t}{4})\frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(\bar{x}_t,\bar{y}_t)-\bar{w}_t\|^2+(1+\frac{4}{\mu\gamma\alpha_t})\kappa^2\eta^2\alpha_t^2\|\bar{\nu}_t\|^2\end{aligned}$$

Since we choose $\gamma \leq \frac{1}{2L}$, $\alpha_t < 1$, we have:

$$(1+\frac{\mu\gamma\alpha_t}{4})(1-\frac{\mu\gamma\alpha_t}{2}) = 1-\frac{\mu\gamma\alpha_t}{4}-\frac{\mu^2\gamma^2\alpha_t^2}{8} \leq 1-\frac{\mu\gamma\alpha_t}{4}$$

and $-(1+\frac{\mu\gamma\alpha_t}{4}) \leq -1, (1+\frac{\mu\gamma\alpha_t}{4}) \leq \frac{9}{8}$, $\mu\gamma\alpha_t < \frac{1}{2}$. Thus, we have

$$\begin{aligned}\|\bar{y}_{t+1}-y_{\bar{x}_{t+1}}\|^2 \leq{}& \big(1-\frac{\mu\gamma\alpha_t}{4}\big)\|\bar{y}_t-y_{\bar{x}_t}\|^2-\frac{\gamma^2\alpha_t}{4}\|\bar{\omega}_t\|^2\\ &+\frac{9\gamma\alpha_t}{2\mu}\underbrace{\|\nabla_y g(\bar{x}_t,\bar{y}_t)-\bar{w}_t\|^2}_{T_1}+\frac{9\kappa^2\eta^2\alpha_t}{2\mu\gamma}\|\bar{\nu}_t\|^2\end{aligned}$$

For the term T_1 in the inequality above, we have:

$$\begin{aligned}\|\nabla_y g(\bar{x}_t,\bar{y}_t)-\bar{w}_t\|^2 &\leq 2\|\nabla_y g(\bar{x}_t,\bar{y}_t)-\frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)})\|^2\\ &\quad+2\|\frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)})-\bar{w}_t\|^2\\ &\leq \frac{2L^2}{M}\sum_{m=1}^{M}\Big[\|x_t^{(m)}-\bar{x}_t\|^2+\|y_t^{(m)}-\bar{y}_t\|^2\Big]\end{aligned}$$

$$+ 2\|\frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}) - \bar{w}_t\|^2$$

This completes the proof. □

Lemma 72. *Suppose we choose* $\tau \leq \frac{1}{2L}$ *and* $\alpha_t < 1$, $r = \frac{C_f}{\mu}$. *Then for* $t \in [T]$, *we have:*

$$\|\bar{u}_{t+1} - u_{\bar{x}_{t+1}}\|^2 \leq \Big(1 - \frac{\mu\tau\alpha_t}{4}\Big)\|\bar{u}_t - u_{\bar{x}_t}\|^2 - \frac{\tau^2\alpha_t}{4}\|\bar{q}_t\|^2 + \frac{9\kappa^2\eta^2\alpha_t}{2\mu\tau}\|\bar{\nu}_t\|^2 + \frac{9\tau\alpha_t}{\mu}\|\bar{p} - \bar{q}_t\|^2$$
$$+ \frac{18\tau\alpha_t\tilde{L}_2^2}{\mu M}\sum_{m=1}^{M}\Big[\|x_t^{(m)} - \bar{x}_t\|^2 + \|y_t^{(m)} - \bar{y}_t\|^2\Big] + \frac{18\tau\alpha_t L^2}{M}\sum_{m=1}^{M}\Big\|u_t^{(m)} - \bar{u}_t\Big\|^2$$

where $\tilde{L}_2^2 = (L^2 + \frac{2L_{y^2}^2 C_f^2}{\mu^2})$ *is a constant.*

Proof. First, we exploit Proposition 114, and choose the function

$$\frac{1}{2}x^T\nabla_{y^2}g(\bar{x}, y_{\bar{x}})x - \nabla_y f(\bar{x}, y_{\bar{x}})^T x$$

, by assumption it is L smooth and μ strongly convex, and we choose $\tau < \frac{1}{2L}$ and $\alpha_t < 1$, thus:

$$\|\bar{u}_{t+1} - u_{\bar{x}_t}\|^2 \leq (1 - \frac{\mu\tau\alpha_t}{2})\|\bar{u}_t - u_{\bar{x}_t}\|^2 - \frac{\tau^2\alpha_t}{4}\|\bar{q}_t\|^2 + \frac{4\tau\alpha_t}{\mu}\|\nabla_{y^2}g(\bar{x}, y_{\bar{x}})\bar{u}_t - \nabla_y f(\bar{x}, y_{\bar{x}}) - \bar{q}_t\|^2.$$

where we also use the fact that

$$\|\bar{u}_{t+1} - u_{\bar{x}_t}\|^2 \leq \|\bar{u}_t - \tau\alpha_t\bar{q}_t - u_{\bar{x}_t}\|^2$$

for $r = \frac{C_f}{\mu} \geq \|u_{\bar{x}_t}\|$. Next, we decompose the term $\|\bar{u}_{t+1} - u_{\bar{x}_{t+1}}\|^2$ as follows:

$$\begin{aligned}\|\bar{u}_{t+1} - u_{\bar{x}_{t+1}}\|^2 &\leq (1 + \frac{\mu\tau\alpha_t}{4})\|\bar{u}_{t+1} - u_{\bar{x}_t}\|^2 + (1 + \frac{4}{\mu\tau\alpha_t})\|u_{\bar{x}_t} - u_{\bar{x}_{t+1}}\|^2 \\ &\leq (1 + \frac{\mu\tau\alpha_t}{4})\|\bar{u}_{t+1} - u_{\bar{x}_t}\|^2 + (1 + \frac{4}{\mu\tau\alpha_t})\bar{L}^2\|\bar{x}_t - \bar{x}_{t+1}\|^2\end{aligned}$$

where the second inequality is due to case a) of Proposition 3.9. Combining the above inequalities 4.12 and 4.13, we have:

$$\begin{aligned}\|\bar{u}_{t+1} - u_{\bar{x}_{t+1}}\|^2 \leq (1 - \frac{\mu\tau\alpha_t}{4})\|\bar{u}_t - u_{\bar{x}_t}\|^2 - \frac{\tau^2\alpha_t}{4}\|\bar{q}_t\|^2 \\ + \frac{9\tau\alpha_t}{2\mu}\underbrace{\|\nabla_{y^2} g(\bar{x}, y_{\bar{x}})\bar{u}_t - \nabla_y f(\bar{x}, y_{\bar{x}}) - \bar{q}_t\|^2}_{T_1} + \frac{9\bar{L}^2\eta^2\alpha_t}{2\mu\tau}\|\bar{\nu}_t\|^2\end{aligned}$$

where we use the fact that $\tau \leq \frac{1}{2L}$, $\alpha_t < 1$ For the term T_1 in the inequality above, we have:

$$\begin{aligned}T_1 &\leq 2\|\nabla_{y^2} g(\bar{x}, y_{\bar{x}})\bar{u}_t - \nabla_y f(\bar{x}, y_{\bar{x}}) - \bar{p}_t\|^2 + 2\|\bar{p}_t - \bar{q}_t\|^2 \\ &\leq 2\Big\|\nabla_{y^2} g(\bar{x}, y_{\bar{x}})\bar{u}_t - \nabla_y f(\bar{x}, y_{\bar{x}}) \\ &\quad - \frac{1}{M}\sum_{m=1}^{M}\Big(\nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)})u_t^{(m)} + \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)})\Big)\Big\|^2 + 2\|\bar{p}_t - \bar{q}_t\|^2\end{aligned}$$

We denote the first term of the above inequality as $T_{1,1}$, we have:

$$\begin{aligned}T_{1,1} &\leq 4\Big\|\nabla_y f(\bar{x}, y_{\bar{x}}) - \frac{1}{M}\sum_{m=1}^{M}\Big(\nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)})\Big)\Big\|^2 \\ &\quad + 4\Big\|\nabla_{y^2} g(\bar{x}, y_{\bar{x}})\bar{u}_t - \frac{1}{M}\sum_{m=1}^{M}\Big(\nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)})u_t^{(m)}\Big)\Big\|^2 \\ &\leq \Big(\frac{4L^2}{M} + \frac{8L_{y^2}^2 C_f^2}{\mu^2 M}\Big)\sum_{m=1}^{M}\Big[\Big\|x_t^{(m)} - \bar{x}_t\Big\|^2 + \Big\|y_t^{(m)} - \bar{u}_t\Big\|^2\Big] + \frac{4L^2}{M}\sum_{m=1}^{M}\Big\|u_t^{(m)} - \bar{u}_t\Big\|^2\end{aligned}$$

Combine everything completes the proof. □

4.7.1.3 Upper Variable Drift

Lemma 73. *For any $t \neq \bar{t}_s, s \in [S]$, we have:*

$$\|x_t^{(m)} - \bar{x}_t\|^2 \leq I\eta^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \alpha_\ell^2 \left\|\nu_\ell^{(m)} - \bar{\nu}_\ell\right\|^2$$
$$\|y_t^{(m)} - \bar{y}_t\|^2 \leq I\gamma^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \alpha_\ell^2 \left\|\omega_\ell^{(m)} - \bar{\omega}_\ell\right\|^2$$
$$\|u_t^{(m)} - \bar{u}_t\|^2 \leq I\tau^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \alpha_\ell^2 \left\|q_\ell^{(m)} - \bar{q}_\ell\right\|^2$$

Proof. Note from Algorithm and the definition of $\bar{t}_s$ that at $t = \bar{t}_s$ with $s \in [S]$, $x_t^{(m)} = \bar{x}_t$, for all k. For $t \neq \bar{t}_s$, with $s \in [S]$, we have: $x_t^{(m)} = x_{t-1}^{(m)} - \eta\alpha_{t-1}\nu_{t-1}^{(m)}$, this implies that: $x_t^{(m)} = x_{\bar{t}_{s-1}}^{(m)} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\alpha_\ell\nu_\ell^{(m)}$ and $\bar{x}_t = \bar{x}_{\bar{t}_{s-1}} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\alpha_\ell\bar{\nu}_\ell$. So for $t \neq \bar{t}_s$, with $s \in [S]$ we have:

$$\begin{aligned}\|x_t^{(m)} - \bar{x}_t\|^2 &= \Big\|x_{\bar{t}_{s-1}}^{(m)} - \bar{x}_{\bar{t}_{s-1}} - \Big(\sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\alpha_\ell\nu_\ell^{(m)} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\alpha_\ell\bar{\nu}_\ell\Big)\Big\|^2 = \Big\|\sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\alpha_\ell\Big(\nu_\ell^{(m)} - \bar{\nu}_\ell\Big)\Big\|^2 \\ &\leq I\eta^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \alpha_\ell^2 \left\|\nu_\ell^{(m)} - \bar{\nu}_\ell\right\|^2\end{aligned}$$

We can derive the bound for $\|y_t^{(m)} - \bar{y}_t\|^2$ and$\|u_t^{(m)} - \bar{u}_t\|^2$ similarly. This completes the proof. □

Lemma 74. *Suppose $\eta\alpha_t < \frac{1}{16I\tilde{L}_1}$, then for $t \neq \bar{t}_s, s \in [S]$, we have:*

$$\sum_{m=1}^{M} \mathbb{E}\|\hat{\nu}_t^{(m)} - \bar{\nu}_t\|^2$$

$$
\begin{aligned}
&\leq \Big(1+\frac{17}{16I}\Big)\sum_{m=1}^{M}\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2+8I\tilde{L}_1^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[2\|\eta\bar{\nu}_{t-1}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big]\\
&\quad+16IL^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\|\tau q_{t-1}^{(m)}\|^2++128I(c_\nu\alpha_{t-1}^2)^2\tilde{L}_1^2\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)}-\bar{x}_t\Big\|^2\\
&\quad+\Big\|y_t^{(m)}-\bar{y}_t\Big\|^2\Big]+32I(c_\nu\alpha_{t-1}^2)^2L^2\sum_{m=1}^{M}\mathbb{E}\Big\|u_t^{(m)}-\bar{u}_t\Big\|^2\\
&\quad+8IM(c_\nu\alpha_{t-1}^2)^2\frac{\sigma^2}{b_x}+32IM(c_\nu\alpha_{t-1}^2)^2\zeta_f^2+64I(c_\nu\alpha_{t-1}^2)^2M\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}
\end{aligned}
$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. For $t\neq\bar{t}_s$, we have:

$$
\begin{aligned}
\mathbb{E}\|\hat{\nu}_t^{(m)}-\bar{\nu}_t\|^2&=\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}+(1-c_\nu\alpha_{t-1}^2)\Big(\nu_{t-1}^{(m)}-\mu_{t-1,\mathcal{B}_x}^{(m)}\Big)\\
&\qquad-\Big(\bar{\mu}_{t,\mathcal{B}_x}+(1-c_\nu\alpha_{t-1}^2)\Big(\bar{\nu}_{t-1}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big)\Big\|^2\\
&=\mathbb{E}\Big\|(1-c_\nu\alpha_{t-1}^2)\Big(\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\Big)+\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}\\
&\qquad-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2\\
&\overset{(a)}{\leq}(1+\frac{1}{I})(1-c_\nu\alpha_{t-1}^2)^2\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2\\
&\qquad+\Big(1+I\Big)\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2\\
&\leq\Big(1+\frac{1}{I}\Big)\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2\\
&\qquad+\Big(1+I\Big)\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2 \qquad (4.14)
\end{aligned}
$$

where (a) follows from the the generalized triangle inequality.

Next we bound the second term of the above inequality:

$$
\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2
$$

$$\leq 2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-\big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\big)\Big\|^2+2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big\|^2$$

where the inequality follows the triangle inequality. We bound the two terms separately, for the first term, we have:

$$\begin{aligned}
&\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-\big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\big)\Big\|^2\overset{(a)}{\leq}\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\mu_{t-1,\mathcal{B}_x}^{(m)}\Big\|^2\\
&\leq\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_x f^{(m)}(x_t^{(m)},y_t^{(m)};\xi_{f,1})-\nabla_{xy}g^{(m)}(x_t^{(m)},y_t^{(m)};\xi_{g,1})u_t^{(m)}\\
&\quad-\big(\nabla_x f^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)};\xi_{f,1})-\nabla_{xy}g^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)};\xi_{g,1})u_{t-1}^{(m)}\big)\Big\|^2\\
&\overset{(b)}{\leq}2\Big(L^2+\frac{2L_{xy}^2C_f^2}{\mu^2}\Big)\sum_{m=1}^{M}\mathbb{E}\Big[\|x_t^{(m)}-x_{t-1}^{(m)}\|^2+\|y_t^{(m)}-y_{t-1}^{(m)}\|^2\Big]+4L^2\sum_{m=1}^{M}\mathbb{E}\|u_t^{(m)}-u_{t-1}^{(m)}\|^2\\
&\leq 2\tilde{L}_1^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|\eta\nu_{t-1}^{(m)}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big]+4L^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\|\tau q_{t-1}^{(m)}\|^2 \qquad (4.15)
\end{aligned}$$

where (a) follows Proposition 111; (b) follows Proposition 67 and the fact that $\hat{x}_t^{(m)}=x_t^{(m)}$ when $t\neq\bar{t}_s$; Next for the second term, we have:

$$\begin{aligned}
&\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big\|^2=\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}-\big(\bar{\mu}_{t-1,\mathcal{B}_x}-\bar{\mu}_{t-1}\big)+\mu_{t-1}^{(m)}-\bar{\mu}_{t-1}\Big\|^2\\
&\overset{(a)}{\leq}2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}-\big(\bar{\mu}_{t-1,\mathcal{B}_x}-\bar{\mu}_{t-1}\big)\Big\|^2+2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1}^{(m)}-\bar{\mu}_{t-1}\Big\|^2\\
&\overset{(b)}{\leq}\underbrace{2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}\Big\|^2}_{T_1}+\underbrace{2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1}^{(m)}-\bar{\mu}_{t-1}\Big\|^2}_{T_2} \qquad (4.16)
\end{aligned}$$

Note for the term T_1 of Eq. 4.16, we have $\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}\Big\|^2\leq\frac{\sigma^2}{b_x}$ by the bounded variance

assumption; Next for the term T_2, we have:

$$
\begin{aligned}
T_2 = \sum_{m=1}^{M} \Big\| & \nabla_x f^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \nabla_{xy} g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) u_{t-1}^{(m)} \\
& - \frac{1}{M} \sum_{j=1}^{M} \Big(\nabla_x f^{(j)}(x_{t-1}^{(j)}, y_{t-1}^{(j)}) - \nabla_{xy} g^{(j)}(x_{t-1}^{(j)}, y_{t-1}^{(j)}) u_{t-1}^{(j)} \Big) \Big\|^2 \\
\leq 16 & \Big(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2} \Big) \sum_{m=1}^{M} \mathbb{E} \Big[\Big\| x_t^{(m)} - \bar{x}_t \Big\|^2 + \Big\| y_t^{(m)} - \bar{y}_t \Big\|^2 \Big] \\
& + 4L^2 \sum_{m=1}^{M} \mathbb{E} \Big\| u_t^{(m)} - \bar{u}_t \Big\|^2 + 4M\zeta_f^2 + \frac{8MC_f^2 \zeta_{g,xy}^2}{\mu^2}
\end{aligned}
$$

Finally, combine Eq. 4.15, Eq. 4.16 with Eq. 4.14 and use the fact that $I \geq 1$, we have:

$$
\begin{aligned}
& \sum_{m=1}^{M} \mathbb{E} \| \hat{\nu}_t^{(m)} - \bar{\nu}_t \|^2 \\
& \leq \Big(1 + \frac{1}{I} \Big) \sum_{m=1}^{M} \mathbb{E} \| \nu_{t-1}^{(m)} - \bar{\nu}_{t-1} \|^2 \\
& \quad + 8I\tilde{L}_1^2 \alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E} \Big[\underbrace{\| \eta \nu_{t-1}^{(m)} \|^2}_{T_1} + \| \gamma \omega_{t-1}^{(m)} \|^2 \Big] + 16IL^2 \alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E} \| \tau q_{t-1}^{(m)} \|^2 \\
& \quad + 128I(c_\nu \alpha_{t-1}^2)^2 \tilde{L}_1^2 \sum_{m=1}^{M} \mathbb{E} \Big[\Big\| x_t^{(m)} - \bar{x}_t \Big\|^2 + \Big\| y_t^{(m)} - \bar{y}_t \Big\|^2 \Big] \\
& \quad + 32I(c_\nu \alpha_{t-1}^2)^2 L^2 \sum_{m=1}^{M} \mathbb{E} \Big\| u_t^{(m)} - \bar{u}_t \Big\|^2 \\
& \quad + 8IM(c_\nu \alpha_{t-1}^2)^2 \frac{\sigma^2}{b_x} + 32IM(c_\nu \alpha_{t-1}^2)^2 \zeta_f^2 + 64I(c_\nu \alpha_{t-1}^2)^2 M \frac{C_f^2 \zeta_{g,xy}^2}{\mu^2}
\end{aligned}
$$

We separate the term T_1 with triangle inequality to get:

$$
\begin{aligned}
& \sum_{m=1}^{M} \mathbb{E} \| \hat{\nu}_t^{(m)} - \bar{\nu}_t \|^2 \\
& \leq \Big(1 + \frac{1}{I} + 16I\tilde{L}_1^2 \eta^2 \alpha_{t-1}^2 \Big) \sum_{m=1}^{M} \mathbb{E} \| \nu_{t-1}^{(m)} - \bar{\nu}_{t-1} \|^2
\end{aligned}
$$

$$
\begin{aligned}
&+ 8I\tilde{L}_1^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[2\|\eta\bar{\nu}_{t-1}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big]+16IL^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\|\tau q_{t-1}^{(m)}\|^2\\
&+ 128I(c_\nu\alpha_{t-1}^2)^2\tilde{L}_1^2\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)}-\bar{x}_t\Big\|^2+\Big\|y_t^{(m)}-\bar{y}_t\Big\|^2\Big]\\
&+ 32I(c_\nu\alpha_{t-1}^2)^2L^2\sum_{m=1}^{M}\mathbb{E}\Big\|u_t^{(m)}-\bar{u}_t\Big\|^2\\
&+ 8IM(c_\nu\alpha_{t-1}^2)^2\frac{\sigma^2}{b_x}+32IM(c_\nu\alpha_{t-1}^2)^2\zeta_f^2+64I(c_\nu\alpha_{t-1}^2)^2M\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}
\end{aligned}
$$

This completes the proof. □

Lemma 75. *Suppose $\gamma\alpha_t<\frac{1}{16IL}$, then for $t\neq\bar{t}_s, s\in[S]$, we have:*

$$
\begin{aligned}
&\sum_{m=1}^{M}\mathbb{E}\|\omega_t^{(m)}-\bar{\omega}_t\|^2\\
&\leq\Big(1+\frac{33}{32I}\Big)\sum_{m=1}^{M}\mathbb{E}\|\omega_{t-1}^{(m)}-\bar{\omega}_{t-1}\|^2+4IL^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[2\|\gamma\bar{\omega}_{t-1}\|^2+\|\eta\nu_{t-1}^{(m)}\|^2\Big]\\
&\quad+8IM(c_\omega\alpha_{t-1}^2)^2\frac{\sigma^2}{b_y}+16IM(c_\omega\alpha_{t-1}^2)^2\zeta_g^2+16IL^2(c_\omega\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_{t-1}^{(m)}-\bar{x}_{t-1}\|^2\Big]\\
&\quad+16IL^2(c_\omega\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|y_{t-1}^{(m)}-\bar{y}_{t-1}\|^2\Big]
\end{aligned}
$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. By the update step in Line 7 of Algorithm 7, for $t\neq\bar{t}_s$, we have:

$$
\begin{aligned}
&\mathbb{E}\|\hat{\omega}_t^{(m)}-\bar{\omega}_t\|^2\\
&=\mathbb{E}\Big\|(1-c_\omega\alpha_{t-1}^2)\Big(\omega_{t-1}^{(m)}-\bar{\omega}_{t-1}\Big)+\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)},\mathcal{B}_y)\\
&\quad-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_t^{(j)},y_t^{(j)},\mathcal{B}_y)\\
&\quad-(1-c_\omega\alpha_{t-1}^2)\Big(\nabla_y g^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)},\mathcal{B}_y)-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_{t-1}^{(j)},y_{t-1}^{(j)},\mathcal{B}_y)\Big)\Big\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq (1+\frac{1}{I})(1-c_\omega\alpha_{t-1}^2)^2\mathbb{E}\|\omega_{t-1}^{(m)}-\bar{\omega}_{t-1}\|^2 \\
&\quad + (1+I)\mathbb{E}\Big\|\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)},\mathcal{B}_y)-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_t^{(j)},y_t^{(j)},\mathcal{B}_y) \\
&\quad - (1-c_\omega\alpha_{t-1}^2)\Big(\nabla_y g^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)},\mathcal{B}_y)-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_{t-1}^{(j)},y_{t-1}^{(j)},\mathcal{B}_y)\Big)\Big\|^2 \qquad (4.17)
\end{aligned}
$$

where the inequality follows from the the generalized triangle inequality and the condition that $c_\omega\alpha_t^2<1$.

Next we denote the second term in Eq. 4.17 as T_1, then we have:

$$
\begin{aligned}
T_1 \leq\ & 2\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)},\mathcal{B}_y)-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(m)}(x_t^{(j)},y_t^{(j)},\mathcal{B}_y) \\
&- \Big(\nabla_y g^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)},\mathcal{B}_y)-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_{t-1}^{(j)},y_{t-1}^{(m)},\mathcal{B}_y)\Big)\Big\|^2 \\
&+ 2(c_\omega\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_y g^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)},\mathcal{B}_y)-\frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_{t-1}^{(j)},y_{t-1}^{(j)},\mathcal{B}_y)\Big\|^2
\end{aligned}
$$

We bound the two terms separately, we denote them as $T_{1,1}$ and $T_{1,2}$ separately, then we have:

$$
\begin{aligned}
T_{1,1} &\overset{(a)}{\leq} \sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)},\mathcal{B}_y)-\nabla_y g^{(m)}(x_{t-1}^{(m)},y_{t-1}^{(m)},\mathcal{B}_y)\Big\|^2 \\
&\overset{(b)}{\leq} L^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_t^{(m)}-x_{t-1}^{(m)}\|^2+\|y_t^{(m)}-y_{t-1}^{(m)}\|^2\Big] \leq L^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|\eta\nu_{t-1}^{(m)}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big]
\end{aligned} \qquad (4.18)
$$

where (a) follows Proposition 111; (b) follows Proposition 67.b) and the fact that $\hat{x}_t^{(m)}=$

$x_t^{(m)}$ and $\hat{y}_t^{(m)} = y_t^{(m)}$ when $t \neq \bar{t}_s$; Next for the second term, we have:

$$
\begin{aligned}
T_{1,2} = \sum_{m=1}^{M} \mathbb{E}\Big\| & \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \mathcal{B}_y) - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) \\
& - \frac{1}{M}\sum_{j=1}^{M}\Big(\nabla_y g^{(j)}(x_{t-1}^{(j)}, y_{t-1}^{(j)}, \mathcal{B}_y) - \nabla_y g^{(j)}(x_{t-1}^{(j)}, y_{t-1}^{(j)})\Big) \\
& + \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(x_{t-1}^{(j)}, y_{t-1}^{(j)})\Big\|^2 \\
\overset{(b)}{\leq} 2\sum_{m=1}^{M} \mathbb{E}\Big\| & \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}, \mathcal{B}_y) - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)})\Big\|^2 \\
& + 4\sum_{m=1}^{M}\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\|\nabla g^{(m)}(\bar{x}_{t-1}, \bar{y}_{t-1}) - \nabla_y g^{(j)}(\bar{x}_{t-1}, \bar{y}_{t-1})\|^2 \\
& + 4\sum_{m=1}^{M} \mathbb{E}\Big\| \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \nabla_y g^{(m)}(\bar{x}_{t-1}, \bar{y}_{t-1}) \\
& + \frac{1}{M}\sum_{j=1}^{M}\nabla_y g^{(j)}(\bar{x}_{t-1}, \bar{y}_{t-1}) - \nabla_y g^{(j)}(x_{t-1}^{(j)}, y_{t-1}^{(j)})\Big\|^2
\end{aligned} \tag{4.19}
$$

We denote the three terms above as $T_{1,2,1} - T_{1,2,3}$ respectively. For the term $T_{1,2,1}$ of Eq. 4.19, we have $T_{1,2,1} \leq 2M\sigma^2/b_y$ by the bounded variance assumption; For the term $T_{1,2,2}$ of Eq. 4.19, by the bounded intra-node heterogeneity assumption we have $T_{1,2,2} \leq 4M\zeta_g^2$. Finally, For the term $T_{1,2,3}$ of Eq. 4.19:

$$
\begin{aligned}
T_{1,2,3} &\leq 4\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)}) - \nabla_y g^{(m)}(\bar{x}_{t-1}, \bar{y}_{t-1})\Big\|^2 \\
&\leq 4L^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2\Big] + 4L^2\sum_{m=1}^{M}\mathbb{E}\Big[\|y_{t-1}^{(m)} - \bar{y}_{t-1}\|^2\Big]
\end{aligned}
$$

Finally, combine Eq. 4.17, Eq. 4.18 with Eq. 4.19 and use the fact that $I \geq 1$, we have:

$$
\sum_{m=1}^{M}\mathbb{E}\|\hat{\omega}_t^{(m)} - \bar{\omega}_t\|^2 \leq \Big(1 + \frac{1}{I}\Big)\sum_{m=1}^{M}\mathbb{E}\|\omega_{t-1}^{(m)} - \bar{\omega}_{t-1}\|^2
$$

$$
\begin{aligned}
&+ 4IL^2\alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E}\Big[\underbrace{\|\gamma\omega_{t-1}^{(m)}\|^2}_{T_1} + \|\eta\nu_{t-1}^{(m)}\|^2 \Big] + 8IM(c_\omega\alpha_{t-1}^2)^2 \frac{\sigma^2}{b_y} \\
&+ 16IM(c_\omega\alpha_{t-1}^2)^2\zeta_g^2 + 16IL^2(c_\omega\alpha_{t-1}^2)^2 \sum_{m=1}^{M} \mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2\Big] \\
&+ 16IL^2(c_\omega\alpha_{t-1}^2)^2 \sum_{m=1}^{M} \mathbb{E}\Big[\|y_{t-1}^{(m)} - \bar{y}_{t-1}\|^2\Big]
\end{aligned}
$$

We separate the term T_1 with triangle inequality to get:

$$
\begin{aligned}
\sum_{m=1}^{M} \mathbb{E}\|\hat{\omega}_t^{(m)} - \bar{\omega}_t\|^2 \leq &\left(1 + \frac{1}{I} + 8IL^2\gamma^2\alpha_{t-1}^2\right) \sum_{m=1}^{M} \mathbb{E}\|\omega_{t-1}^{(m)} - \bar{\omega}_{t-1}\|^2 \\
&+ 4IL^2\alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E}\Big[2\|\gamma\bar{\omega}_{t-1}\|^2 + \|\eta\nu_{t-1}^{(m)}\|^2\Big] \\
&+ 8IM(c_\omega\alpha_{t-1}^2)^2 \frac{\sigma^2}{b_y} \\
&+ 16IM(c_\omega\alpha_{t-1}^2)^2\zeta_g^2 + 16IL^2(c_\omega\alpha_{t-1}^2)^2 \sum_{m=1}^{M} \mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2\Big] \\
&+ 16IL^2(c_\omega\alpha_{t-1}^2)^2 \sum_{m=1}^{M} \mathbb{E}\Big[\|y_{t-1}^{(m)} - \bar{y}_{t-1}\|^2\Big]
\end{aligned}
$$

This completes the proof. □

Lemma 76. *Suppose $\tau\alpha_t < \frac{1}{32IL}$, then for $t \neq \bar{t}_s, s \in [S]$, we have:*

$$
\begin{aligned}
\sum_{m=1}^{M} \mathbb{E}\|\hat{q}_t^{(m)} - \bar{q}_t\|^2 \leq &\left(1 + \frac{33}{32I}\right) \sum_{m=1}^{M} \mathbb{E}\|q_{t-1}^{(m)} - \bar{q}_{t-1}\|^2 + 8I\tilde{L}_2^2\alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E}\Big[\|\gamma\omega_{t-1}^{(m)}\|^2 + \|\eta\nu_{t-1}^{(m)}\|^2\Big] \\
&+ 32IL^2\alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E}\|\tau^2\bar{q}_{t-1}\|^2 + 8IM(c_u\alpha_{t-1}^2)^2 \frac{\sigma^2}{b_x} \\
&+ 16IM(c_u\alpha_{t-1}^2)^2\zeta_f^2 + 32IM(c_u\alpha_{t-1}^2)^2 \frac{C_f^2\zeta_{g,yy}^2}{\mu^2} \\
&+ 64I(c_u\alpha_{t-1}^2)^2\tilde{L}_2^2 \sum_{m=1}^{M} \mathbb{E}\Big[\Big\|x_t^{(m)} - \bar{x}_t\Big\|^2 + \Big\|y_t^{(m)} - \bar{y}_t\Big\|^2\Big] \\
&+ 16I(c_u\alpha_{t-1}^2)^2L^2 \sum_{m=1}^{M} \mathbb{E}\Big\|u_t^{(m)} - \bar{u}_t\Big\|^2
\end{aligned}
$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. For $t \neq \bar{t}_s$, we have:

$$
\begin{aligned}
&\mathbb{E}\|\hat{q}_t^{(m)} - \bar{q}_t\|^2 \\
&= \mathbb{E}\Big\|(1 - c_u\alpha_{t-1}^2)\Big(q_{t-1}^{(m)} - \bar{q}_{t-1}\Big) + p_{t,\mathcal{B}_x}^{(m)} - \bar{p}_{t,\mathcal{B}_x} - (1 - c_u\alpha_{t-1}^2)\Big(p_{t-1,\mathcal{B}_x}^{(m)} - \bar{p}_{t-1,\mathcal{B}_x}\Big)\Big\|^2 \\
&\leq (1 + \frac{1}{I})(1 - c_u\alpha_{t-1}^2)^2\mathbb{E}\|q_{t-1}^{(m)} - \bar{q}_{t-1}\|^2 \\
&\quad + (1 + I)\mathbb{E}\Big\|p_{t,\mathcal{B}_x}^{(m)} - \bar{p}_{t,\mathcal{B}_x} - (1 - c_u\alpha_{t-1}^2)\Big(p_{t-1,\mathcal{B}_x}^{(m)} - \bar{p}_{t-1,\mathcal{B}_x}\Big)\Big\|^2
\end{aligned}
$$

where the inequality follows from the the generalized triangle inequality and the condition that $c_u\alpha_t^2 < 1$.

Next we sum over M for the second term in Eq. 4.17 and denote it as T_1, then we have:

$$
T_1 \leq 2\sum_{m=1}^{M}\mathbb{E}\Big\|p_{t,\mathcal{B}_x}^{(m)} - \bar{p}_{t,\mathcal{B}_x} - \Big(p_{t-1,\mathcal{B}_x}^{(m)} - \bar{p}_{t-1,\mathcal{B}_x}\Big)\Big\|^2 + 2(c_u\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big\|p_{t-1,\mathcal{B}_x}^{(m)} - \bar{p}_{t-1,\mathcal{B}_x}\Big\|^2
$$

We bound the two terms separately, we denote them as $T_{1,1}$ and $T_{1,2}$ separately, then we have:

$$
\begin{aligned}
T_{1,1} &\overset{(a)}{\leq} \sum_{m=1}^{M}\mathbb{E}\Big\|p_{t,\mathcal{B}_x}^{(m)} - p_{t-1,\mathcal{B}_x}^{(m)}\Big\|^2 \\
&\overset{(b)}{\leq} 2\Big(L^2 + \frac{2L_{y^2}^2C_f^2}{\mu^2}\Big)\sum_{m=1}^{M}\mathbb{E}\Big[\|x_t^{(m)} - x_{t-1}^{(m)}\|^2 + \|y_t^{(m)} - y_{t-1}^{(m)}\|^2\Big] + 4L^2\sum_{m=1}^{M}\mathbb{E}\|u_t^{(m)} - u_{t-1}^{(m)}\|^2 \\
&\leq 2\tilde{L}_2^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|\eta\nu_{t-1}^{(m)}\|^2 + \|\gamma\omega_{t-1}^{(m)}\|^2\Big] + 4L^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\|\tau^2 q_{t-1}^{(m)}\|^2
\end{aligned}
$$

where (a) follows Proposition 111; (b) follows Proposition 67 and the fact that $\hat{x}_t^{(m)} = x_t^{(m)}$

and $\hat{y}_t^{(m)} = y_t^{(m)}$ when $t \neq \bar{t}_s$; Next for the second term, we have:

$$
\begin{aligned}
T_{1,2} &= \sum_{m=1}^{M} \mathbb{E}\Big\| p_{t-1,\mathcal{B}_x}^{(m)} - p_{t-1}^{(m)} - \big(\bar{p}_{t-1,\mathcal{B}_x} - \bar{p}_{t-1}\big) + p_{t-1}^{(m)} - \bar{p}_{t-1} \Big\|^2 \\
&\overset{(b)}{\leq} 2 \sum_{m=1}^{M} \mathbb{E}\Big\| p_{t-1,\mathcal{B}_x}^{(m)} - p_{t-1}^{(m)} \Big\|^2 + 2 \sum_{m=1}^{M} \mathbb{E}\Big\| p_{t-1}^{(m)} - \bar{p}_{t-1} \Big\|^2
\end{aligned}
$$

We denote the two terms above as $T_{1,2,1}, T_{1,2,2}$ respectively. For the term $T_{1,2,1}$ of Eq. 4.19, we have $T_{1,2,1} \leq 2M\sigma^2/b_x$ by the bounded variance assumption; For the term $T_{1,2,2}$ of Eq. 4.19, we have

$$
\begin{aligned}
T_{1,2,2} \leq 16\Big(L^2 + \frac{2L_{y^2}^2 C_f^2}{\mu^2}\Big) \sum_{m=1}^{M} \mathbb{E}\Big[\Big\| x_t^{(m)} - \bar{x}_t \Big\|^2 + \Big\| y_t^{(m)} - \bar{y}_t \Big\|^2 \Big] \\
+ 4L^2 \sum_{m=1}^{M} \mathbb{E}\Big\| u_t^{(m)} - \bar{u}_t \Big\|^2 + 4M\zeta_f^2 + \frac{8MC_f^2\zeta_{g,yy}^2}{\mu^2}
\end{aligned}
$$

Finally, combine everythin together and use the fact that $I \geq 1$, we have:

$$
\begin{aligned}
\sum_{m=1}^{M} \mathbb{E}\|\hat{q}_t^{(m)} - \bar{q}_t\|^2 \leq \Big(1 + \frac{1}{I}\Big) \sum_{m=1}^{M} \mathbb{E}\|q_{t-1}^{(m)} - \bar{q}_{t-1}\|^2 + 8I\tilde{L}_2^2\alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E}\Big[\|\gamma\omega_{t-1}^{(m)}\|^2 + \|\eta\nu_{t-1}^{(m)}\|^2\Big] \\
+ 16IL^2\alpha_{t-1}^2 \sum_{m=1}^{M} \mathbb{E} \underbrace{\|\tau^2 q_{t-1}^{(m)}\|^2}_{T_1} \\
+ 8IM(c_u\alpha_{t-1}^2)^2\frac{\sigma^2}{b_x} + 16IM(c_u\alpha_{t-1}^2)^2\zeta_f^2 + 32IM(c_u\alpha_{t-1}^2)^2\frac{C_f^2\zeta_{g,yy}^2}{\mu^2} \\
+ 64I(c_u\alpha_{t-1}^2)^2\tilde{L}_2^2 \sum_{m=1}^{M} \mathbb{E}\Big[\Big\| x_t^{(m)} - \bar{x}_t \Big\|^2 \\
+ \Big\| y_t^{(m)} - \bar{y}_t \Big\|^2 \Big] + 16I(c_u\alpha_{t-1}^2)^2 L^2 \sum_{m=1}^{M} \mathbb{E}\Big\| u_t^{(m)} - \bar{u}_t \Big\|^2
\end{aligned}
$$

We separate the term T_1 with triangle inequality to get:

$$\begin{aligned}\sum_{m=1}^{M}\mathbb{E}\|\hat{q}_t^{(m)}-\bar{q}_t\|^2 \le &\Big(1+\frac{1}{I}+32IL^2\tau^2\alpha_{t-1}^2\Big)\sum_{m=1}^{M}\mathbb{E}\|q_{t-1}^{(m)}-\bar{q}_{t-1}\|^2\\ &+8I\tilde{L}_2^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|\gamma\omega_{t-1}^{(m)}\|^2+\|\eta\nu_{t-1}^{(m)}\|^2\Big]\\ &+32IL^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\|\tau^2\bar{q}_{t-1}\|^2+8IM(c_u\alpha_{t-1}^2)^2\frac{\sigma^2}{b_x}\\ &+16IM(c_u\alpha_{t-1}^2)^2\zeta_f^2+32IM(c_u\alpha_{t-1}^2)^2\frac{C_f^2\zeta_{g,yy}^2}{\mu^2}\\ &+64I(c_u\alpha_{t-1}^2)^2\tilde{L}_2^2\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)}-\bar{x}_t\Big\|^2+\Big\|y_t^{(m)}-\bar{y}_t\Big\|^2\Big]\\ &+16I(c_u\alpha_{t-1}^2)^2L^2\sum_{m=1}^{M}\mathbb{E}\Big\|u_t^{(m)}-\bar{u}_t\Big\|^2\end{aligned}$$

This completes the proof. □

Next, to simply the notation, we denote $A_t = \mathbb{E}\|\bar{\nu}_t-\bar{\mu}_t\|^2$, $B_t = \mathbb{E}\|\bar{y}_t-y_{\bar{x}_t}\|^2$, $C_t = \mathbb{E}\|\bar{\omega}_t-\frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)})\|^2$, $D_t = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\nu_t^{(m)}-\bar{\nu}_t\|^2$, $E_t = \mathbb{E}\|\bar{\nu}_t\|^2$, $F_t = \mathbb{E}\|\bar{\omega}_t\|^2$, $G_t = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\omega_t^{(m)}-\bar{\omega}_t\|^2$, $H_t = \mathbb{E}[\|\bar{q}_t-\bar{p}_t\|^2]$, $I_t = \mathbb{E}[\|\bar{u}_t-u_{\bar{x}_t}\|^2]$, $J_t = \mathbb{E}\|q_t^{(m)}-\bar{q}_t\|^2$, $Q_t = \mathbb{E}\|\bar{q}_t\|^2$.

Lemma 77. *For $\eta < \min(\frac{\tilde{L}^2}{c_\nu},\frac{\tilde{L}^2}{c_\omega},\frac{\tilde{L}^2}{c_u},1)$, $\gamma < \min(\frac{\tilde{L}^2}{c_\nu},\frac{\tilde{L}^2}{c_\omega},\frac{\tilde{L}^2}{c_u},1)$, $\tau < \min(\frac{\tilde{L}^2}{c_\nu},\frac{\tilde{L}^2}{c_u},\frac{1}{2})$ and $\alpha_t < \frac{1}{16\tilde{L}I}$, where $\tilde{L} = max(\tilde{L}_1,\tilde{L}_2)$, we have:*

$$\begin{aligned}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t \le \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(&\alpha_t E_t+\alpha_t F_t+\alpha_t Q_t+\frac{c_\omega^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_y}+\frac{c_\omega^2\alpha_t^3}{\tilde{L}^2}\zeta_g^2\\ &+\frac{c_\nu^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_x}+\frac{c_u^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_x}+\frac{c_\nu^2\alpha_t^3\zeta_f^2}{\tilde{L}^2}+\frac{c_u^2\alpha_t^3\zeta_f^2}{\tilde{L}^2}+\frac{2c_\nu^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{4c_u^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,yy}^2}{\mu^2}\Big)\\ \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t \le \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(&\alpha_t E_t+\alpha_t F_t+\alpha_t Q_t+\frac{2c_\omega^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_y}+\frac{2c_\omega^2\alpha_t^3}{\tilde{L}^2}\zeta_g^2\end{aligned}$$

$$+\frac{c_\nu^2\alpha_t^3}{\tilde{L}^2}\frac{2\sigma^2}{b_x}+\frac{c_u^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_x}+\frac{c_\nu^2\alpha_t^3\zeta_f^2}{\tilde{L}^2}+\frac{c_u^2\alpha_t^3\zeta_f^2}{\tilde{L}^2}+\frac{c_\nu^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{2c_u^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,yy}^2}{\mu^2}\Big)$$

$$\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s}\alpha_t J_t \le \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\alpha_t F_t+\alpha_t E_t+\alpha_t Q_t+\frac{c_\omega^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_y}+\frac{c_\omega^2\alpha_t^3}{\tilde{L}^2}\zeta_g^2$$

$$+\frac{c_u^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_x}+\frac{c_\nu^2\alpha_t^3}{\tilde{L}^2}\frac{\sigma^2}{b_x}+\frac{c_u^2\alpha_t^3\zeta_f^2}{\tilde{L}^2}+\frac{c_\nu^2\alpha_t^3\zeta_f^2}{2\tilde{L}^2}+\frac{c_\nu^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{40c_u^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,yy}^2}{\mu^2}\Big)$$

Proof. Based on Lemma 74, for $t \neq \bar{t}_s$, we have:

$$\begin{aligned}
D_t \le{}& \Big(1+\frac{17}{16I}\Big)D_{t-1}+16I\tilde{L}_1^2\alpha_{t-1}^2\eta^2E_{t-1}+16I\tilde{L}_1^2\alpha_{t-1}^2\gamma^2F_{t-1}+16I\tilde{L}_1^2\alpha_{t-1}^2\gamma^2G_{t-1}\\
&+32IL^2\tau^2\alpha_{t-1}^2J_{t-1}+32IL^2\tau^2\alpha_{t-1}^2Q_{t-1}+8Ic_\nu^2\alpha_{t-1}^4\frac{\sigma^2}{b_x}+32Ic_\nu^2\alpha_{t-1}^4\zeta_f^2\\
&+64Ic_\nu^2\alpha_{t-1}^4\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}+128I^2\tilde{L}_1^2\eta^2c_\nu^2\alpha_{t-1}^4\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2D_l+128I^2\tilde{L}_1^2\gamma^2c_\nu^2\alpha_{t-1}^4\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2G_l\\
&+32I^2L^2\tau^2c_\nu^2\alpha_{t-1}^4\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2J_l
\end{aligned}$$

while for $t=\bar{t}_s$, we have $D_{\bar{t}_s}=1/M\sum_{m=1}^M\mathbb{E}\|\nu_{\bar{t}_s}^{(m)}-\bar{\nu}_{\bar{t}_s}\|^2=0$. Apply the above equation recursively from $\bar{t}_{s-1}+1$ to t. so we have:

$$\begin{aligned}
D_t \le{}& \sum_{\ell=\bar{t}_{s-1}}^{\ell}\Big(1+\frac{17}{16I}\Big)^{t-\ell}\Big(16I\tilde{L}_1^2\alpha_\ell^2\eta^2E_\ell+16I\tilde{L}_1^2\alpha_\ell^2\gamma^2F_\ell+16I\tilde{L}_1^2\alpha_\ell^2\gamma^2G_\ell+32IL^2\tau^2\alpha_\ell^2J_\ell\\
&+32IL^2\tau^2\alpha_\ell^2Q_\ell+8Ic_\nu^2\alpha_\ell^4\frac{\sigma^2}{b_x}+32Ic_\nu^2\alpha_\ell^4\zeta_f^2+64Ic_\nu^2\alpha_\ell^4\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}\\
&+128I^2\tilde{L}_1^2\eta^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{l}}^2D_{\bar{\ell}}+128I^2\tilde{L}_1^2\gamma^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{\ell}}^2G_{\bar{\ell}}\\
&+32I^2L^2\tau^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{\ell}}^2J_{\bar{\ell}}\Big)\\
\le{}& \sum_{\ell=\bar{t}_{s-1}}^{t-1}\Big(48I\tilde{L}_1^2\alpha_\ell^2\eta^2E_\ell+48I\tilde{L}_1^2\alpha_\ell^2\gamma^2F_\ell+48I\tilde{L}_1^2\alpha_\ell^2\gamma^2G_\ell+96IL^2\tau^2\alpha_\ell^2J_\ell\\
&+96IL^2\tau^2\alpha_\ell^2Q_\ell+24Ic_\nu^2\alpha_\ell^4\frac{\sigma^2}{b_x}+96Ic_\nu^2\alpha_\ell^4\zeta_f^2+192Ic_\nu^2\alpha_\ell^4\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}
\end{aligned}$$

$$+ 384I^2\tilde{L}_1^2\eta^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{l}}^2D_{\bar{\ell}} + 384I^2\tilde{L}_1^2\gamma^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{\ell}}^2G_{\bar{\ell}}$$
$$+ 96I^2L^2\tau^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{\ell}}^2J_{\bar{\ell}}\Big)$$

The second inequality uses the fact that $t-l\leq I$ and the inequality $log(1+a/x)\leq a/x$ for $x>-a$, so we have $(1+a/x)^x\leq e^a$, Then we choose $a=17/16$ and $x=I$. Finally, we use the fact that $e^{17/16}\leq 3$.

Next we multiply α_t over both sides and take sum from $\bar{t}_{s-1}+1$ to $\bar{t}_s$, we have:

$$\begin{aligned}
&\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s}\alpha_tD_t\\
&\leq\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\Big(48I\tilde{L}_1^2\alpha_\ell^2\eta^2E_\ell+48I\tilde{L}_1^2\alpha_\ell^2\gamma^2F_\ell+48I\tilde{L}_1^2\alpha_\ell^2\gamma^2G_\ell+96IL^2\tau^2\alpha_\ell^2J_\ell\\
&\quad+96IL^2\tau^2\alpha_\ell^2Q_\ell+24Ic_\nu^2\alpha_\ell^4\frac{\sigma^2}{b_x}+96Ic_\nu^2\alpha_\ell^4\zeta_f^2+192Ic_\nu^2\alpha_\ell^4\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}\\
&\quad+384I^2\tilde{L}_1^2\eta^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{l}}^2D_{\bar{\ell}}+384I^2\tilde{L}_1^2\gamma^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{\ell}}^2G_{\bar{\ell}}+96I^2L^2\tau^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1}\alpha_{\bar{\ell}}^2J_{\bar{\ell}}\Big)\\
&\overset{(a)}{\leq}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(3I\tilde{L}_1\alpha_t^2\eta^2E_t+3I\tilde{L}_1\alpha_t^2\gamma^2F_t+3I\tilde{L}_1\alpha_t^2\gamma^2G_t+6IL\tau^2\alpha_t^2J_t\\
&\quad+6IL\tau^2\alpha_t^2Q_t+\frac{3Ic_\nu^2\alpha_t^4}{2\tilde{L}}\frac{\sigma^2}{b_x}+\frac{6Ic_\nu^2\alpha_t^4\zeta_f^2}{\tilde{L}}+\frac{12Ic_\nu^2\alpha_t^4}{\tilde{L}}\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}\\
&\quad+32I^2\tilde{L}_1\eta^2c_\nu^2\alpha_t^4\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2D_\ell+32I^2\tilde{L}_1\gamma^2c_\nu^2\alpha_t^4\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2G_\ell+6I^2L\tau^2c_\nu^2\alpha_t^4\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2J_\ell\Big)\\
&\overset{(b)}{\leq}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\frac{3\eta^2}{16}\alpha_tE_t+\frac{3\gamma^2}{16}\alpha_tF_t+\frac{3\gamma^2}{16}\alpha_tG_t+\frac{3\tau^2}{8}\alpha_tJ_t+\frac{3\tau^2}{8}\alpha_tQ_t\\
&\quad+\frac{3c_\nu^2\alpha_t^3}{32\tilde{L}^2}\frac{\sigma^2}{b_x}+\frac{3c_\nu^2\alpha_t^3\zeta_f^2}{8\tilde{L}^2}+\frac{3c_\nu^2\alpha_t^3}{4\tilde{L}^2}\frac{C_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{\eta^2c_\nu^2}{8*16^3I^2\tilde{L}^4}\alpha_tD_t\\
&\quad+\frac{\gamma^2c_\nu^2}{8*16^3I^2\tilde{L}^4}\alpha_tG_t+\frac{3\tau^2c_\nu^2}{8*16^4I^2\tilde{L}^4}\alpha_tJ_t\Big)
\end{aligned}$$

In inequalities (a) and (b), we use $\alpha_t < \frac{1}{16\tilde{L}I} \leq \frac{1}{16\tilde{L}_1 I}$. Note that $\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s} \alpha_t D_t = \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_t D_t$ as $D_{\bar{t}_s} = D_{\bar{t}_{s-1}} = 0$.

Then if we choose $\eta < \frac{\tilde{L}^2}{c_\nu}$ and $\gamma < \frac{\tilde{L}^2}{c_\nu}$, $\tau < \frac{\tilde{L}^2}{c_\nu}$, we have

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_t D_t \leq \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \Big(\frac{\eta^2}{4}\alpha_t E_t + \frac{\gamma^2}{4}\alpha_t F_t + \frac{\gamma^2}{2}\alpha_t G_t + \tau^2 \alpha_t J_t + \frac{\tau^2}{2}\alpha_t Q_t + \frac{c_\nu^2 \alpha_t^3}{8\tilde{L}^2}\frac{\sigma^2}{b_x} + \frac{c_\nu^2 \alpha_t^3 \zeta_f^2}{2\tilde{L}^2} + \frac{c_\nu^2 \alpha_t^3}{\tilde{L}^2}\frac{C_f^2 \zeta_{g,xy}^2}{\mu^2} \Big) \tag{4.20}$$

Based on Lemma 75, for $t \neq \bar{t}_s$, we have:

$$\begin{aligned} G_t \leq & \left(1 + \frac{33}{32I}\right) G_{t-1} + 8IL^2\eta^2\alpha_{t-1}^2 D_{t-1} + 8IL^2\eta^2\alpha_{t-1}^2 E_{t-1} + 8IL^2\gamma^2\alpha_{t-1}^2 F_{t-1} \\ & + 8Ic_\omega^2\alpha_{t-1}^4\frac{\sigma^2}{b_y} + 16Ic_\omega^2\alpha_{t-1}^4\zeta_g^2 + 16I^2L^2\eta^2c_\omega^2\alpha_{t-1}^4 \sum_{\ell=\bar{t}_{s-1}}^{t-2} \alpha_l^2 D_l \\ & + 16I^2L^2\gamma^2c_\omega^2\alpha_{t-1}^4 \sum_{\ell=\bar{t}_{s-1}}^{t-2} \alpha_l^2 G_l \end{aligned}$$

Follow similar derivation, by recursively applying the above inequality, we have:

$$\begin{aligned} G_t \leq & \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big(24IL^2\eta^2\alpha_\ell^2 D_\ell + 24IL^2\eta^2\alpha_\ell^2 E_\ell + 24IL^2\gamma^2\alpha_\ell^2 F_\ell \\ & + 24Ic_\omega^2\alpha_\ell^4\frac{\sigma^2}{b_y} + 48Ic_\omega^2\alpha_\ell^4\zeta_g^2 + 48I^2L^2\eta^2c_\omega^2\alpha_\ell^4 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} \alpha_{\bar{\ell}}^2 D_{\bar{\ell}} + 48I^2L^2\gamma^2c_\omega^2\alpha_\ell^4 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} \alpha_{\bar{\ell}}^2 G_{\bar{\ell}} \Big) \end{aligned}$$

Next we multiply α_t over both sides and take sum from $\bar{t}_{s-1}+1$ to $\bar{t}_s$, use the condition that $\alpha_t < \frac{1}{16I\tilde{L}} < \frac{1}{16IL}$, $\eta < \frac{\tilde{L}^2}{c_\omega}$ and $\gamma < \frac{\tilde{L}^2}{c_\omega}$, we have:

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_t G_t \leq \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \Big(\frac{1}{2}\eta^2\alpha_t D_t + \frac{1}{8}\eta^2\alpha_t E_t + \frac{1}{8}\gamma^2\alpha_t F_t + \frac{c_\omega^2\alpha_t^3}{8\tilde{L}^2}\frac{\sigma^2}{b_y} + \frac{c_\omega^2\alpha_t^3}{8\tilde{L}^2}\zeta_g^2\Big) \tag{4.21}$$

Based on Lemma 76, we have:

$$\begin{aligned}
J_t \leq & \left(1 + \frac{33I}{32I}\right) J_{t-1} + 16I\tilde{L}_2^2\tau^2\alpha_{t-1}^2 G_{t-1} + 16I\tilde{L}_2^2\tau^2\alpha_{t-1}^2 F_{t-1} + 16I\tilde{L}_2^2\eta^2\alpha_{t-1}^2 D_{t-1} \\
& + 16I\tilde{L}_2^2\eta^2\alpha_{t-1}^2 E_{t-1} + 16IL^2\tau^2\alpha_{t-1}^2 Q_{t-1} + 8Ic_u^2\alpha_{t-1}^4\frac{\sigma^2}{b_x} \\
& + 16Ic_u^2\alpha_{t-1}^4\zeta_f^2 + 32Ic_u^2\alpha_{t-1}^4\frac{C_f^2\zeta_{g,yy}^2}{\mu^2} + 64I^2c_u^2\eta^2\alpha_{t-1}^4\tilde{L}_2^2\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2 D_l \\
& + 64I^2c_u^2\gamma^2\alpha_{t-1}^4\tilde{L}_2^2\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2 G_l + 16I^2c_u^2\tau^2\alpha_{t-1}^4 L^2\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2 J_l
\end{aligned}$$

Suppose we have $\alpha_t < \frac{1}{16\tilde{L}I}$, $\eta < \frac{\tilde{L}^2}{c_u}$, $\gamma < \frac{\tilde{L}^2}{c_u}$, $\tau < \frac{\tilde{L}^2}{c_u}$

$$\begin{aligned}
\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s} \alpha_t J_t \leq \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \Big(& \frac{\tau^2}{2}\alpha G_t + \frac{\tau^2}{4}\alpha_t F_t + \frac{\eta^2}{2}\alpha_t D_t + \frac{\eta^2}{4}\alpha_{t-1}E_t \\
& + \frac{\tau^2}{4}\alpha_t Q_t + \frac{c_u^2\alpha_t^3}{8\tilde{L}^2}\frac{\sigma^2}{b_x} + \frac{c_u^2\alpha_t^3\zeta_f^2}{4\tilde{L}^2} + \frac{3c_u^2\alpha_t^3}{\tilde{L}^2}\frac{C_f^2\zeta_{g,yy}^2}{\mu^2}\Big)
\end{aligned} \tag{4.22}$$

Next, we combine Eq. 4.20, Eq. 4.21 and Eq. 4.22 to have the result in the lemma. □

4.7.1.4 Descent Lemma

Lemma 78. *For all $t \in [\bar{t}_{s-1}, \bar{t}_s - 1]$, the iterates generated satisfy:*

$$\mathbb{E}\left\|\nabla h(\bar{x}_t) - \bar{\mu}_t\right\|^2 \leq \frac{2\tilde{L}_1^2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + 2\left\|\bar{y}_t - y_t^{(m)}\right\|^2$$

$$+ 2\Big\|y_{\bar{x}_t} - \bar{y}_t\Big\|^2\Big] + 4L^2\mathbb{E}\Big\|u_{\bar{x}_t} - \bar{u}_t\Big\|^2$$

where we denote $u_{\bar{x}_t} = [\nabla_{y^2} g(\bar{x}_t, y_{\bar{x}_t})]^{-1}\nabla_y f(\bar{x}_t, y_{\bar{x}_t})$ *and* $\tilde{L}_1^2 = \Big(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2}\Big)$ *is a constant.*

Proof. This lemma follows the same derivation as Lemma 88. □

Lemma 79. *Suppose* $\eta\alpha_t < \frac{1}{2\bar{L}}$*, for all* $t \in [\bar{t}_{s-1}, \bar{t}_s - 1]$ *and* $s \in [S]$*, the iterates generated satisfy:*

$$\begin{aligned}\mathbb{E}\Big[h(\bar{x}_{t+1})\Big] \leq \mathbb{E}\Big[h(\bar{x}_t)\Big] - \frac{\eta\alpha_t}{4}\mathbb{E}\Big[\Big\|\bar{\nu}_t\Big\|^2\Big] - \frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] + \eta\alpha_t\mathbb{E}\Big[\Big\|\bar{u}_t - \bar{\nu}_t\Big\|^2\Big] \\ + \frac{2\tilde{L}_1^2\eta\alpha_t}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\bar{x}_t - x_t^{(m)}\Big\|^2 + 2\Big\|\bar{y}_t - y_t^{(m)}\Big\|^2 \\ + 2\Big\|y_{\bar{x}_t} - \bar{y}_t\Big\|^2\Big] + 4L^2\eta\alpha_t\mathbb{E}\Big\|u_{\bar{x}_t} - \bar{u}_t\Big\|^2\end{aligned}$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. By the smoothness of $h(x)$ we have:

$$\begin{aligned}\mathbb{E}[h(\bar{x}_{t+1})] &\leq \mathbb{E}\Big[h(\bar{x}_t) + \langle\nabla h(\bar{x}_t), \bar{x}_{t+1} - \bar{x}_t\rangle + \frac{\bar{L}}{2}\|\bar{x}_{t+1} - \bar{x}_t\|^2\Big] \\ &\overset{(a)}{=} \mathbb{E}\Big[h(\bar{x}_t) - \eta\alpha_t\langle\nabla h(\bar{x}_t), \bar{\nu}_t\rangle + \frac{\eta^2\alpha_t^2\bar{L}}{2}\|\bar{\nu}_t\|^2\Big] \\ &\overset{(b)}{=} \mathbb{E}\Big[h(\bar{x}_t) - \frac{\eta\alpha_t}{2}\Big\|\bar{\nu}_t\Big\|^2 - \frac{\eta\alpha_t}{2}\|\nabla h(\bar{x}_t)\|^2 + \frac{\eta\alpha_t}{2}\Big\|\nabla h(\bar{x}_t) - \bar{\nu}_t\Big\|^2 + \frac{\eta\alpha_t^2\bar{L}}{2}\Big\|\bar{\nu}_t\Big\|^2\Big] \\ &= \mathbb{E}\Big[h(\bar{x}_t) - \frac{\eta\alpha_t}{4}\Big\|\bar{\nu}_t\Big\|^2 - \frac{\eta\alpha_t}{2}\|\nabla h(\bar{x}_t)\|^2 + \frac{\eta\alpha_t}{2}\underbrace{\Big\|\nabla h(\bar{x}_t) - \bar{\nu}_t\Big\|^2}_{T_1}\Big]\end{aligned}$$

where equality (a) follows from the iterate update given in Algorithm 7; (b) uses $\langle a, b\rangle =$

$\frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a-b\|^2]$ and $\eta\alpha_t < \frac{1}{2\tilde{L}}$; For the term T_1, we have:

$$\mathbb{E}\Big[\Big\|\nabla h(\bar{x}_t) - \bar{\nu}_t\Big\|^2\Big] \leq 2\mathbb{E}\Big[\Big\|\nabla h(\bar{x}_t) - \bar{u}_t\Big\|^2\Big] + 2\mathbb{E}\Big[\Big\|\bar{u}_t - \bar{\nu}_t\Big\|^2\Big]$$

Use Lemma 69 for the first term and combine everything together finishes the proof. □

4.7.1.5 Proof of Convergence Theorem

We first denote the following potential function $\mathcal{G}(t)$:

$$\begin{aligned}\mathcal{G}_t = h(\bar{x}_t) + \frac{9bM\eta}{64\alpha_t}\Big\|\bar{\nu}_t - \bar{\mu}_t\Big\|^2 + \frac{18\eta\tilde{L}^2}{\mu\gamma}\Big\|\bar{y}_t - y_{\bar{x}_t}\Big\|^2 + \frac{9bM\eta}{64\alpha_t}\Big\|\bar{q}_t - \bar{p}_t\Big\|^2 \\ + \frac{9bM\eta}{64\alpha_t}\Big\|\bar{\omega}_t - \frac{1}{M}\sum_{m=1}^{M}\nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2 + \frac{18\eta L^2}{\mu\tau}\Big\|\bar{u}_t - u_{\bar{x}_t}\Big\|^2\end{aligned}$$

Furthermore, we have constants $\tilde{L}_1^2 = \Big(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2}\Big)$ and $\tilde{L}_2^2 = \Big(L^2 + \frac{2L_{y_2}^2 C_f^2}{\mu^2}\Big)$, to ease the writing, without loss of generality, we assume the second order Lipschitz constants $L_{xy} = L_{y^2}$, as a result $\tilde{L}_1^2 = \tilde{L}_2^2$, we denote it as $\tilde{L}$ in the subsequent proof.

Theorem 80. *Suppose we choose* $c_\nu = \frac{64}{9bM} + \frac{2}{3b^2M^2}$, $c_\omega = \frac{48^2}{bM\mu^2} + \frac{2}{3b^2M^2}$, $c_u = \frac{48^2}{bM\mu^2} + \frac{2}{3b^2M^2}$ $u = (bM\sigma)^2\bar{u}$, *where* $\bar{u} = \max\Big(2, 16^2 I^3\tilde{L}^2, c_\nu^{3/2}, c_\omega^{3/2}\Big)$, $\delta = \frac{(bM\sigma)^{2/3}}{(16\tilde{L})^{1/3}}$, $\alpha_t = \frac{\delta}{(u+t)^{1/3}}, t \in [T]$, $\gamma < \min\Big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, \frac{\tilde{L}^2}{c_\nu}, \frac{\tilde{L}^2}{c_\omega}, \frac{\tilde{L}^2}{c_u}, \frac{1}{2\tilde{L}}, 1\Big)$, $\eta < \min\Big(\frac{\mu\gamma}{36\kappa\tilde{L}}, \frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, \frac{\tilde{L}^2}{c_\nu}, \frac{\tilde{L}^2}{c_\omega}, \frac{\tilde{L}^2}{c_u}, \frac{1}{2\tilde{L}}, 1\Big)$, $\tau < \min\Big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, \frac{\tilde{L}^2}{c_\nu}, \frac{\tilde{L}^2}{c_u}, \frac{1}{2\tilde{L}}, \frac{1}{2}\Big)$ *where* C_1 *is a constant, we set the mini-batch size* $b_x = b_y = b$ *and the first batch with size* $b_1 = O(Ib)$, $r = \frac{C_f}{\mu}$, *then we have:*

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] = O\Big(\frac{\kappa^{19/3} I}{T} + \frac{\kappa^{16/3}}{(bMT)^{2/3}}\Big)$$

To reach an ϵ-stationary point, we need $T = O(\kappa^8(bM)^{-1}\epsilon^{-1.5})$, $I = O(\kappa^{5/3}(bM)^{-1}\epsilon^{-0.5})$.

Proof. By the condition that $u \geq c_\nu^{3/2}\delta^3$, it is straightforward to verify that $c_\nu\alpha_t^2 < 1$. By Lemma 69, we have:

$$\frac{A_t}{\alpha_{t-1}} - \frac{A_{t-1}}{\alpha_{t-2}} \leq \left(\alpha_{t-1}^{-1} - \alpha_{t-2}^{-1} - c_\nu\alpha_{t-1}\right)A_{t-1} + \frac{2c_\nu^2\alpha_{t-1}^3\sigma^2}{bM} + \frac{16L^2\tau^2\alpha_{t-1}}{bM}(J_{t-1} + Q_{t-1}) + \frac{8\tilde{L}^2\eta^2\alpha_{t-1}}{bM}(D_{t-1} + E_{t-1}) + \frac{8\tilde{L}^2\gamma^2\alpha_{t-1}}{bM}(F_{t-1} + G_{t-1})$$

where we choose $b_x = b_y = b$. For $\alpha_{t-1}^{-1} - \alpha_{t-2}^{-1}$, we have:

$$\alpha_t^{-1} - \alpha_{t-1}^{-1} = \frac{(u + \sigma^2 t)^{1/3}}{\delta} - \frac{(u + \sigma^2(t-1))^{1/3}}{\delta} \overset{(a)}{\leq} \frac{\sigma^2}{3\delta(u + \sigma^2(t-1))^{2/3}} \overset{(b)}{\leq} \frac{2^{2/3}\sigma^2\delta^2}{3\delta^3(u + \sigma^2 t)^{2/3}} \overset{(c)}{=} \frac{2^{2/3}\sigma^2}{3\delta^3}\alpha_t^2 \leq \frac{2}{3Ib^2M^2}\alpha_t \leq \frac{2^{2/3}\sigma^2}{3\delta^3}\alpha_t^2 \leq \frac{2}{3b^2M^2}\alpha_t$$

where inequality (a) results from the concavity of $x^{1/3}$ as: $(x+y)^{1/3} - x^{1/3} \leq y/3x^{2/3}$, inequality (b) used the fact that $u_t \geq 2\sigma^2$, inequality (c) uses the definition of α_t, By choosing $c_\nu = \frac{64}{9bM} + \frac{2}{3b^2M^2}$, we have:

$$\frac{A_t}{\alpha_{t-1}} - \frac{A_{t-1}}{\alpha_{t-2}} \leq -\frac{64}{9bM}\alpha_{t-1}A_{t-1} + \frac{2c_\nu^2\alpha_{t-1}^3\sigma^2}{bM} + \frac{16L^2\tau^2\alpha_{t-1}}{bM}(J_{t-1} + Q_{t-1}) + \frac{8\tilde{L}^2\eta^2\alpha_{t-1}}{bM}(D_{t-1} + E_{t-1}) + \frac{8\tilde{L}^2\gamma^2\alpha_{t-1}}{bM}(F_{t-1} + G_{t-1})$$

Next, we telescope from $\bar{t}_{s-1} + 1$ to $\bar{t}_s$:

$$\left(\frac{A_{\bar{t}_s}}{\alpha_{\bar{t}_s-1}} - \frac{A_{\bar{t}_{s-1}}}{\alpha_{\bar{t}_{s-1}-1}}\right) \leq -\frac{64}{9bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t A_t + \frac{2c_\nu^2\sigma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3 + \frac{16\tilde{L}^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t$$

$$+\frac{8\tilde{L}^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t+\frac{8\tilde{L}^2\gamma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t+\frac{16\tilde{L}^2\gamma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t$$
$$+\frac{32L^2\tau^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t J_t+\frac{16L^2\tau^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t Q_t \tag{4.23}$$

Next, we follow similar derivation as $A_t/\alpha_{t-1}-A_{t-1}/\alpha_{t-2}$. By Lemma 70. we choose $c_\omega=\frac{48^2}{bM\mu^2}+\frac{2}{3b^2M^2}$, to obtain:

$$\frac{C_t}{\alpha_{t-1}}-\frac{C_{t-1}}{\alpha_{t-2}}\le-\frac{48^2\alpha_{t-1}}{bM\mu^2}C_{t-1}+\frac{2c_\omega^2\alpha_{t-1}^3\sigma^2}{bM}$$
$$+\frac{4L^2\eta^2\alpha_{t-1}}{bM}(D_{t-1}+E_{t-1})+\frac{4L^2\gamma^2\alpha_{t-1}}{bM}(F_{t-1}+G_{t-1})$$

Then telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$, we have:

$$\frac{C_{\bar{t}_s}}{\alpha_{\bar{t}_s-1}}-\frac{C_{\bar{t}_{s-1}}}{\alpha_{\bar{t}_{s-1}-1}}$$
$$\le-\frac{48^2}{bM\mu^2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t C_t+\frac{2c_\omega^2\sigma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3+\frac{16L^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t$$
$$+\frac{8L^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t+\frac{8L^2\gamma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t+\frac{16L^2\gamma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t \tag{4.24}$$

Next from Lemma 68, we choose $c_u=\frac{48^2}{bM\mu^2}+\frac{2}{3b^2M^2}$, to obtain:

$$\frac{H_t}{\alpha_{t-1}}-\frac{H_{t-1}}{\alpha_{t-2}}\le-\frac{48^2\alpha_{t-1}}{bM\mu^2}H_{t-1}+\frac{2c_u^2\alpha_{t-1}^3}{bM}\sigma^2+\frac{8\eta^2\alpha_{t-1}\tilde{L}^2}{bM}(D_{t-1}+E_{t-1})$$
$$+\frac{8\gamma^2\alpha_{t-1}\tilde{L}^2}{bM}(F_{t-1}+G_{t-1})+\frac{8\tau^2\alpha_{t-1}L^2}{bM}(J_{t-1}+Q_{t-1})$$

Then telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$, we have:

$$\frac{H_{\bar{t}_s}}{\alpha_{\bar{t}_s-1}} - \frac{H_{\bar{t}_{s-1}}}{\alpha_{\bar{t}_{s-1}-1}} \leq -\frac{48^2}{bM\mu^2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t H_t + \frac{2c_u^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3\sigma^2 + \frac{8\eta^2\tilde{L}^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_{t-1}(D_t+E_t)$$
$$+ \frac{8\gamma^2\tilde{L}^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_{t-1}(F_t+G_t) + \frac{8\tau^2L^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t(J_t+Q_{t-1}) \tag{4.25}$$

Next from Lemma 71, for $t \neq \bar{t}_s$, we have:

$$B_{t+1} - B_t \leq -\frac{\mu\gamma\alpha_t B_t}{4} - \frac{\gamma^2\alpha_t F_t}{4} + \frac{9\gamma\alpha_t C_t}{\mu} + \frac{9\kappa^2\eta^2\alpha_t E_t}{2\mu\gamma}$$
$$+ \frac{9\gamma\alpha_t L^2}{\mu}\sum_{\ell=\bar{t}_{s-1}}^{t-1} I\eta^2\alpha_\ell^2 D_\ell + \frac{9\gamma\alpha_t L^2}{\mu}\sum_{\ell=\bar{t}_{s-1}}^{t-1} I\gamma^2\alpha_\ell^2 G_\ell$$

When $t=\bar{t}_s$, we do not have the last two terms in the above inequality. Next, we telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$ and have:

$$\begin{aligned} B_{\bar{t}_s} - B_{\bar{t}_{s-1}} &\leq -\frac{\mu\gamma}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t - \frac{\gamma^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t + \frac{9\gamma}{\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t C_t + \frac{9\kappa^2\eta^2}{2\mu\gamma}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t \\ &\quad + \frac{9I\eta^2\gamma L^2}{\mu}\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 D_\ell + \frac{9I\gamma^3L^2}{\mu}\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 G_\ell \\ &\leq -\frac{\mu\gamma}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t - \frac{\gamma^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t + \frac{9\gamma}{\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t C_t + \frac{9\kappa^2\eta^2}{2\mu\gamma}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t \\ &\quad + \frac{9L^2\eta^2\gamma}{16^2\hat{L}^2\mu}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell D_\ell + \frac{9L^2\gamma^3}{16^2\hat{L}^2\mu}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell G_\ell \end{aligned} \tag{4.26}$$

where we use the fact that $\alpha_t < \frac{1}{16\hat{L}I}$. Next, from Lemma 72, we have:

$$I_{t+1} - I_t \leq -\frac{\mu\tau\alpha_t}{4}I_t - \frac{\tau^2\alpha_t}{4}Q_t + \frac{9\kappa^2\eta^2\alpha_t}{2\mu\tau}E_t + \frac{9\tau\alpha_t}{\mu}H_t$$

$$+\frac{18I\eta^2\tau\alpha_t\tilde{L}^2}{\mu}\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 D_\ell+\frac{18I\gamma^2\tau\alpha_t\tilde{L}^2}{\mu}\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 G_\ell+18I\tau^3\alpha_t L^2\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 J_\ell$$

when $t=\bar{t}_s$, we do not have the last three terms in the above inequality. Next, we telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$ and have:

$$\begin{aligned}
I_{\bar{t}_s}-I_{\bar{t}_{s-1}} &\leq -\frac{\mu\tau}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t I_t-\frac{\tau^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t Q_t+\frac{9\kappa^2\eta^2}{2\mu\tau}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t+\frac{9\tau}{\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t H_t\\
&\quad+\frac{18I\eta^2\tau\tilde{L}^2}{\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 D_\ell+\frac{18I\gamma^2\tau\tilde{L}^2}{\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 G_\ell\\
&\quad+18I\tau^3L^2\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell^2 J_\ell\\
&\leq -\frac{\mu\tau}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t I_t-\frac{\tau^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t Q_t+\frac{9\kappa^2\eta^2}{2\mu\tau}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t+\frac{9\tau}{\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t H_t\\
&\quad+\frac{18\eta^2\tau}{16^2\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t+\frac{18\gamma^2\tau}{16^2\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t+\frac{18\tau^3L^2}{16^2\tilde{L}^2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t J_t
\end{aligned}\tag{4.27}$$

Next, by Lemma 79, when $t+1\neq\bar{t}_s$, we have:

$$\begin{aligned}
\mathbb{E}[h(\bar{x}_{t+1})] &\leq \mathbb{E}[h(\bar{x}_t)]-\frac{\eta\alpha_t}{4}E_t-\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2]+\eta\alpha_t A_t+4\tilde{L}^2\eta\alpha_t B_t+4L^2\eta\alpha_t I_t\\
&\quad+2\tilde{L}^2I\eta^3\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2 D_l+4\tilde{L}^2I\gamma^2\eta\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2 G_l
\end{aligned}$$

When $t=\bar{t}_s$, we do not have the last two terms. Next, we telescope from $\bar{t}_{s-1}$ to $\bar{t}_s-1$ to have:

$$\begin{aligned}
&\mathbb{E}[h(\bar{x}_{\bar{t}_s})-h(\bar{x}_{\bar{t}_{s-1}})]\\
&\leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{4}E_t-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2]+4L^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t I_t+\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\eta\alpha_t A_t
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} 4\tilde{L}^2\eta\alpha_t B_t + 2\tilde{L}^2 I\eta^3 \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t \sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2 D_l + 4\tilde{L}^2 I\gamma^2\eta \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t \sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2 G_l \\
\leq &-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{4}E_t - \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2] + 4L^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t I_t + \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\eta\alpha_t A_t \\
&+\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} 4\tilde{L}^2\eta\alpha_t B_t + \frac{\eta^3}{128}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t + \frac{\gamma^2\eta}{64}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t
\end{aligned} \tag{4.28}
$$

In the inequality, we use the fact that $\bar{t}_s - \bar{t}_{s-1} \leq I$, $\alpha_t < \frac{1}{16\tilde{L}I}$.

Combine Eq. (4.23),Eq. (4.24), Eq. (4.26) and Eq. (4.28) and we have:

$$
\begin{aligned}
&\mathbb{E}[\mathcal{G}_{\bar{t}_s}] - \mathbb{E}[\mathcal{G}_{\bar{t}_{s-1}}] \\
\leq &-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2] + \Big(\frac{9\eta c_\omega^2\sigma^2}{32} + \frac{9\eta c_\nu^2\sigma^2}{32} + \frac{9\eta c_u^2\sigma^2}{32}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3 \\
&-\frac{\tilde{L}^2\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t - \frac{L^2\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t I_t - \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\frac{9\eta\gamma\tilde{L}^2}{2\mu} - \frac{9\eta\gamma^2\tilde{L}^2}{4} - \frac{9\eta\gamma^2L^2}{8}\Big)\alpha_t F_t \\
&-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\frac{1}{4} - \frac{81\kappa^2\tilde{L}^2\eta^2}{\mu^2\gamma^2} - \frac{81\kappa^2L^2\eta^2}{\mu^2\tau^2} - \frac{9L^2\eta^2}{8} - \frac{9\tilde{L}^2\eta^2}{4}\Big)\eta\alpha_t E_t \\
&-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\frac{9\tau\eta L^2}{2\mu} - \frac{9\eta\tau^2L^2}{4}\Big)\alpha_t Q_t + \Big(\frac{81\kappa^2}{64} + \frac{9L^2}{4} + 9\tilde{L}^2\Big)\tau^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t J_t \\
&+\Big(\frac{1}{128} + \frac{81\kappa^2}{128} + \frac{81\kappa^2}{64} + \frac{9L^2}{4} + \frac{9\tilde{L}^2}{2}\Big)\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t \\
&+\Big(\frac{1}{64} + \frac{81\kappa^2}{128} + \frac{81\kappa^2}{64} + \frac{9\tilde{L}^2}{4} + \frac{9L^2}{2}\Big)\gamma^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t
\end{aligned}
$$

By the condition that $\eta < \frac{\mu\gamma}{36\kappa\tilde{L}}$ and $\gamma \leq \frac{1}{2L} < \frac{1}{2\mu}$. Next, we denote:

$$
C_1 = \frac{1}{64} + \frac{81\kappa^2}{32} + 9\tilde{L}^2 = O(\kappa^2)
$$

Then, we have:

$$
\begin{aligned}
&\mathbb{E}[\mathcal{G}_{\bar{t}_s}] - \mathbb{E}[\mathcal{G}_{\bar{t}_{s-1}}] \\
&\leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] + \Big(\frac{9\eta c_\omega^2\sigma^2}{32} + \frac{9\eta c_\nu^2\sigma^2}{32} + \frac{9\eta c_u^2\sigma^2}{32}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3 \\
&\quad - \frac{9\eta\gamma^2\tilde{L}^2}{8}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t - \frac{\eta}{8}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t - \frac{\tilde{L}^2\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t - \frac{L^2\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t I_t \\
&\quad - \frac{9\eta\tau^2L^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t Q_t + C_1\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t + C_1\gamma^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_t + C_1\tau^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t J_t \quad (4.29)
\end{aligned}
$$

Combine Eq. (4.29) with Lemma 77, and use the condition that $\eta < \min\Big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, 1\Big)$, $\gamma < \min\Big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, 1\Big)$ and $\tau < \min\Big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, 1\Big)$ we have:

$$
\mathbb{E}[\mathcal{G}_{\bar{t}_s}] - \mathbb{E}[\mathcal{G}_{\bar{t}_{s-1}}] \leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] + C_{\sigma,\zeta}\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3
$$

For ease of notation, we denote

$$
C_{\sigma,\zeta} = \Big(4c_\omega^2\sigma^2 + 4c_u^2\sigma^2 + 4c_\nu^2\sigma^2 + 3c_u^2\zeta_f^2 + 3c_\nu^2\zeta_f^2 + 3c_\omega^2\zeta_g^2 + \frac{3c_\nu^2C_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{120c_u^2C_f^2\zeta_{g,yy}^2}{\mu^2}\Big).
$$

Next, sum over all $s \in [S]$ (assume $T = SI + 1$ without loss of generality), we have:

$$
\mathbb{E}[\mathcal{G}_T] - \mathbb{E}[\mathcal{G}_1] \leq -\sum_{t=1}^{T-1}\frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] + \eta C_{\sigma,\zeta}\sum_{t=1}^{T-1}\alpha_t^3
$$

Rearranging the terms and use the fact that α_t is non-increasing, we have:

$$
\frac{\eta\alpha_T}{2}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] \leq \mathbb{E}[\mathcal{G}_1] - \mathbb{E}[\mathcal{G}_T] + \eta C_{\sigma,\zeta}\sum_{t=1}^{T-1}\alpha_t^3
$$

$$\leq h(x_1) - h^* + \frac{9bM\eta A_1}{64\alpha_1} + \frac{18\eta\tilde{L}^2 B_1}{\mu\gamma} + \frac{9bM\eta C_1}{64\alpha_1} + \frac{9bM\eta H_1}{64\alpha_1} + \frac{18\eta L^2 I_1}{\mu\tau} + \eta C_{\sigma,\zeta}\sum_{t=1}^{T-1}\alpha_t^3$$

where we use $\mathcal{G}_T \geq h^*$ (h^* is the optimal value of h), and for the last term, we use the following fact:

$$\sum_{t=1}^{T}\alpha_t^3 = \sum_{t=1}^{T}\frac{\delta^3}{u+\sigma^2 t} \leq \sum_{t=1}^{T}\frac{\delta^3}{\sigma^2+\sigma^2 t} = \frac{\delta^3}{\sigma^2}\sum_{t=1}^{T}\frac{1}{1+t} \leq \frac{\delta^3}{\sigma^2}\ln(T+1) = \frac{b^2M^2\ln(T+1)}{16\tilde{L}}$$

the first inequality follows $u_t > \sigma^2$, the last inequality follows Proposition 112.

Next, we denote the initial sub-optimality as $\Delta = h(\bar{x}_1) - h^*$, initial inner variable estimation error *i.e.* $B_1 = \|y_1 - y_{x_1}\|^2 \leq \Delta_y$ and the initial hyper-gradient computation error $I_1 = \|u_1 - [\nabla_{y^2} g(x_1, y_{x_1})]^{-1}\nabla_y f(x_1, y_{x_1})\|^2 \leq \Delta_u$.

Furthermore, we have $A_1 =\leq \frac{\sigma^2}{b_1 M}$, $C_1 \leq \frac{\sigma^2}{b_1 M}$, $H_1 \leq \frac{\sigma^2}{b_1 M}$ where b_1 be the size of the first batch. Then, we divide both sides by $\eta\alpha_T T/2$ to have:

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\big[\|\nabla h(\bar{x}_t)\|^2\big] \leq \big(\frac{2\Delta}{\eta} + \frac{27b\sigma^2}{32b_1\alpha_1} + \frac{36\tilde{L}^2\Delta_y}{\mu\gamma} + \frac{36L^2\Delta_u}{\mu\tau} + \frac{b^2M^2C_{\sigma,\zeta}\ln(T)}{8\tilde{L}}\big)\frac{1}{T\alpha_T}$$

Note that we have:

$$\frac{1}{\alpha_t t} = \frac{(u+\sigma^2 t)^{1/3}}{\delta t} \leq \frac{u^{1/3}}{\delta t} + \frac{\sigma^{2/3}}{\delta t^{2/3}}$$

where the inequality uses the fact that $(x+y)^{1/3} \leq x^{1/3} + y^{1/3}$. In particular, when $t = 1$,

we have

$$\frac{1}{\alpha_1} \leq \frac{u^{1/3} + \sigma^{2/3}}{\delta} = \frac{(16\tilde{L})^{1/3}((bM)^{2/3}\bar{u}^{1/3} + 1)}{(bM)^{2/3}} \tag{4.30}$$

when $t = T$, we have:

$$\frac{1}{\alpha_T T} \leq \frac{u^{1/3}}{\delta T} + \frac{\sigma^{2/3}}{\delta T^{2/3}} = (16\tilde{L})^{1/3}\left(\frac{\bar{u}^{1/3}}{T} + \frac{1}{(bMT)^{2/3}}\right) \tag{4.31}$$

In summary, we have:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\big[\|\nabla h(\bar{x}_t)\|^2\big] \\
&\leq \big(\frac{2\Delta}{\eta} + \frac{27b\sigma^2}{32b_1\alpha_1} + \frac{36\tilde{L}^2\Delta_y}{\mu\gamma} + \frac{36L^2\Delta_u}{\mu\tau} + \frac{b^2M^2C_{\sigma,\zeta}\ln(T)}{8\tilde{L}}\big)\big(\frac{(16\tilde{L}\bar{u})^{1/3}}{T} + \frac{(16\tilde{L})^{1/3}}{(bMT)^{2/3}}\big)
\end{aligned}$$

Recall that $\tilde{L} = O(\kappa)$, $\bar{L} = O(\kappa^3)$, therefore we have $c_\nu = \Theta((bM)^{-1})$, $c_\omega = \Theta(\kappa^2(bM)^{-1})$, $c_u = \Theta(\kappa^2(bM)^{-1})$ $\bar{u} = \Theta(I^3\kappa^3)$, then for η, γ, τ, we have:

$$\gamma < \min\big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, \frac{\tilde{L}^2}{c_\nu}, \frac{\tilde{L}^2}{c_\omega}, \frac{\tilde{L}^2}{c_u}, \frac{1}{2L}, 1\big)$$

$$\tau < \min\big(\frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, \frac{\tilde{L}^2}{c_\nu}, \frac{\tilde{L}^2}{c_u}, \frac{1}{2L}, \frac{1}{2}\big)$$

$$\eta < \min\big(\frac{\mu\gamma}{36\kappa\tilde{L}}, \frac{1}{8C_1^{1/2}}, \frac{\tilde{L}}{4C_1^{1/2}}, \frac{\tilde{L}^2}{c_\nu}, \frac{\tilde{L}^2}{c_\omega}, \frac{\tilde{L}^2}{c_u}, \frac{1}{2\bar{L}}, 1\big)$$

where $C_1 = O(\kappa^2)$, so we have $\gamma^{-1} = O(\kappa)$, $\eta^{-1} = O(\kappa^3)$, $\tau^{-1} = O(\kappa)$, furthermore,

$\alpha_1^{-1} = O(I\kappa^{4/3})$, $C_{\sigma,\zeta} = O(\kappa^6(bM)^{-2})$, assume we choose the size of the first batch to be $b_1 = Ib$.

Combine everything together, we have:

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] = O\Big(\frac{\kappa^{19/3}I}{T} + \frac{\kappa^{16/3}}{(bMT)^{2/3}}\Big)$$

To reach an ϵ-stationary point, we need $T = O(\kappa^8(bM)^{-1}\epsilon^{-1.5})$, $I = O(\kappa^{5/3}(bM)^{-1}\epsilon^{-0.5})$. The communication cost is $E = T/I \geq \kappa^{19/3}\epsilon^{-1}$, the sample complexity is $Gc(f,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$, $Gc(g,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$, $Jv(g,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$,
$Hv(g,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$ □

4.7.2 Proof for the FedBiO Algorithm

Algorithm 8 follows Eq. 4.6, and we discuss its convergence property in this subsection.

4.7.2.1 Lower Problem Solution Error and Hyper-gradient Estimation Error

Lemma 81. *When $\gamma < \frac{1}{2L}$, we have:*

$$\begin{aligned}\mathbb{E}\Big\|\bar{y}_t - y_{\bar{x}_t}\Big\|^2 \leq (1-\frac{\mu\gamma}{4})\mathbb{E}\Big\|\bar{y}_{t-1} - y_{\bar{x}_{t-1}}\Big\|^2 + \frac{9\kappa^2\eta^2}{2\mu\gamma}\mathbb{E}\Big\|\bar{\nu}_{t-1}\Big\|^2 - \frac{\gamma^2}{4}\mathbb{E}\|\bar{\omega}_{t-1}\|^2 \\ + \frac{9L^2\gamma}{2\mu M}\sum_{m=1}^{M}\mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2 + \|y_{t-1}^{(m)} - \bar{y}_{t-1})\|^2\Big] + \frac{4\gamma\sigma^2}{\mu b_y M}\end{aligned}$$

Algorithm 8 Federated Bilevel Optimization (**FedBiO**)

Input: Initial states x_1, y_1 and u_1; learning rates $\{\gamma_t, \eta_t, \tau_t\}_{t=1}^T$
Initialization: Set $x_1^{(m)} = x_1$, $y_1^{(m)} = y_1$, $u_1^{(m)} = u_1$;
for $t = 1$ **to** T **do**
  Randomly sample mutually independent minibatch of samples $\mathcal{B}_y$ and $\mathcal{B}_x = \{\mathcal{B}_{g,1}, \mathcal{B}_{g,2}, \mathcal{B}_{f,1}, \mathcal{B}_{f,2}\}$ of size b;
  $\omega_t^{(m)} = \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y)$
  $\nu_t^{(m)} = \nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,1}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,1}) u_t^{(m)}$;
  $\hat{y}_{t+1}^{(m)} = y_t^{(m)} - \gamma_t \omega_t^{(m)}$, $\hat{x}_{t+1}^{(m)} = x_t^{(m)} - \eta_t \nu_t^{(m)}$;
  if t mod I $= 0$ **then**
    $y_{t+1}^{(m)} = \frac{1}{M}\sum_{j=1}^M \hat{y}_{t+1}^{(j)}$; $x_{t+1}^{(m)} = \frac{1}{M}\sum_{j=1}^M \hat{x}_{t+1}^{(j)}$
  else
    $y_{t+1}^{(m)} = \hat{y}_{t+1}^{(m)}$, $x_{t+1}^{(m)} = \hat{x}_{t+1}^{(m)}$
  end if
  $\hat{u}_{t+1}^{(m)} = \mathcal{P}_r(\tau_t \nabla_y f^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{f,2}) + (I - \tau_t \nabla_{y^2} g^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_{g,2})) u_t^{(m)})$;
  if t mod I $= 0$ **then**
    $u_{t+1}^{(m)} = \frac{1}{M}\sum_{j=1}^M \hat{u}_{t+1}^{(j)}$
  else
    $u_{t+1}^{(m)} = \hat{u}_{t+1}^{(m)}$
  end if
end for

Lemma 82. *Suppose we choose $\tau < \frac{1}{L}$, then we have:*

$$\begin{aligned}
&\mathbb{E}\left\|\bar{u}_{t+1} - u_{\bar{x}_{t+1}}\right\|^2 \\
&\leq (1 - \frac{\mu\tau}{4})\mathbb{E}\left\|\bar{u}_t - u_{\bar{x}_t}\right\|^2 + \frac{5\tau^2\sigma^2}{4b_x M} + \frac{5\eta^2\bar{L}^2}{\mu\tau}\mathbb{E}\left\|\bar{\nu}_t\right\|^2 \\
&\quad + \frac{5}{4}\Big(\frac{3\tau L^2}{\mu M} + \frac{\tau L_{y^2}^2 C_f^2}{2\mu^3 M}\Big)\sum_{m=1}^M \mathbb{E}\Big[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + 2\left\|\bar{y}_t - y_t^{(m)}\right\|^2 + 2\left\|y_{\bar{x}_t} - \bar{y}_t\right\|^2\Big]
\end{aligned}$$

We provide the proof for Lemma 82 here and Lemma 81 can be derived similarly.

Proof. First, by proposition 114 (set $\alpha = 1$) and choose $\gamma < \frac{1}{2L}$, we have:

$$\mathbb{E}\|\bar{y}_t - y_{\bar{x}_{t-1}}\|^2 \leq (1 - \frac{\mu\gamma}{2})\mathbb{E}\|\bar{y}_{t-1} - y_{\bar{x}_{t-1}}\|^2 - \frac{\gamma^2}{4}\mathbb{E}\|\bar{\omega}_{t-1}\|^2$$

$$
\begin{aligned}
&+ \frac{4\gamma}{\mu}\mathbb{E}\|\nabla_y g(\bar{x}_{t-1}, \bar{y}_{t-1}) - \frac{1}{M}\sum_{m=1}^{M} \nabla_y g(x_{t-1}^{(m)}, y_{t-1}^{(m)})\|^2 + \frac{4\gamma\sigma^2}{\mu b_y M} \\
&\leq (1 - \frac{\mu\gamma}{2})\mathbb{E}\|\bar{y}_{t-1} - y_{\bar{x}_{t-1}}\|^2 - \frac{\gamma^2}{4}\mathbb{E}\|\bar{\omega}_{t-1}\|^2 \\
&\quad + \frac{4\gamma}{\mu M}\sum_{m=1}^{M} \mathbb{E}\|\nabla_y^{(m)} g(\bar{x}_{t-1}, \bar{y}_{t-1}) - \nabla_y g(x_{t-1}^{(m)}, y_{t-1}^{(m)})\|^2 + \frac{4\gamma\sigma^2}{\mu b_y M} \\
&\leq (1 - \frac{\mu\gamma}{2})\mathbb{E}\|\bar{y}_{t-1} - y_{\bar{x}_{t-1}}\|^2 - \frac{\gamma^2}{4}\mathbb{E}\|\bar{\omega}_{t-1}\|^2 \\
&\quad + \frac{4L^2\gamma}{\mu M}\sum_{m=1}^{M} \mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2 + \|y_{t-1}^{(m)} - \bar{y}_{t-1})\|^2\Big] + \frac{4\gamma\sigma^2}{\mu b_y M}
\end{aligned}
$$

Furthermore, by the generalized triangle inequality, we have:

$$
\begin{aligned}
\mathbb{E}\Big\|\bar{y}_t - y_{\bar{x}_t}\Big\|^2 &\leq (1 - \frac{\mu\gamma}{4})\mathbb{E}\Big\|\bar{y}_{t-1} - y_{\bar{x}_{t-1}}\Big\|^2 + (1 + \frac{4}{\mu\gamma})\mathbb{E}\Big\|y_{\bar{x}_t} - y_{\bar{x}_{t-1}}\Big\|^2 - (1 + \frac{\mu\gamma}{4})\frac{\gamma^2}{4}\mathbb{E}\|\bar{\omega}_{t-1}\|^2 \\
&\quad + (1 + \frac{\mu\gamma}{4})\frac{4L^2\gamma}{\mu M}\sum_{m=1}^{M} \mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2 + \|y_{t-1}^{(m)} - \bar{y}_{t-1})\|^2\Big] + \frac{4\gamma\sigma^2}{\mu b_y M} \\
&\leq (1 - \frac{\mu\gamma}{4})\mathbb{E}\Big\|\bar{y}_{t-1} - y_{\bar{x}_{t-1}}\Big\|^2 + \frac{9\kappa^2\eta^2}{2\mu\gamma}\mathbb{E}\Big\|\bar{\nu}_{t-1}\Big\|^2 - \frac{\gamma^2}{4}\mathbb{E}\|\bar{\omega}_{t-1}\|^2 \\
&\quad + \frac{9L^2\gamma}{2\mu M}\sum_{m=1}^{M} \mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2 + \|y_{t-1}^{(m)} - \bar{y}_{t-1})\|^2\Big] + \frac{4\gamma\sigma^2}{\mu b_y M}
\end{aligned}
$$

where the second inequality is due to $\gamma < 1/2L$. This completes the proof. □

4.7.2.2 Local Variable Drift

Lemma 83. *For any $t \neq \bar{t}_s, s \in [S]$, we have:*

$$
\|x_t^{(m)} - \bar{x}_t\|^2 \leq I\eta^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big\|\nu_\ell^{(m)} - \bar{\nu}_\ell\Big\|^2,\ \|y_t^{(m)} - \bar{y}_t\|^2 \leq I\gamma^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big\|\omega_\ell^{(m)} - \bar{\omega}_\ell\Big\|^2
$$

Proof. Note from Algorithm and the definition of $\bar{t}_s$ that at $t = \bar{t}_s$ with $s \in [S]$, $x_t^{(m)} = \bar{x}_t$, for all k. For $t \neq \bar{t}_s$, with $s \in [S]$, we have: $x_t^{(m)} = x_{t-1}^{(m)} - \eta\nu_{t-1}^{(m)}$, this implies that: $x_t^{(m)} = x_{\bar{t}_{s-1}}^{(m)} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\nu_\ell^{(m)}$ and $\bar{x}_t = \bar{x}_{\bar{t}_{s-1}} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\bar{\nu}_\ell$. So for $t \neq \bar{t}_s$, with $s \in [S]$ we have:

$$\begin{aligned}\|x_t^{(m)} - \bar{x}_t\|^2 &= \Big\|x_{\bar{t}_{s-1}}^{(m)} - \bar{x}_{\bar{t}_{s-1}} - \Big(\sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\nu_\ell^{(m)} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\bar{\nu}_\ell\Big)\Big\|^2 = \Big\|\sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\Big(\nu_\ell^{(m)} - \bar{\nu}_\ell\Big)\Big\|^2 \\ &\leq I\eta^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big\|\nu_\ell^{(m)} - \bar{\nu}_\ell\Big\|^2\end{aligned}$$

We can derive the bound for $\|y_t^{(m)} - \bar{y}_t\|^2$ similarly. This completes the proof. □

Lemma 84. *For any $t \in [T]$, we have:*

$$\begin{aligned}\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|\Big(\nu_t^{(m)} - \bar{\nu}_t\Big)\Big\|^2 \leq \frac{4L^2}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|u_t^{(m)} - \bar{u}_t\Big\|^2 + 4\zeta_f^2 + \frac{8C_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{2\sigma^2}{b_x} \\ + \Big(\frac{16L^2}{M} + \frac{32L_{xy}^2C_f^2}{\mu^2 M}\Big)\sum_{m=1}^{M} \mathbb{E}\Big[\Big\|x_t^{(m)} - \bar{x}_t\Big\|^2 + \Big\|y_t^{(m)} - \bar{y}_t\Big\|^2\Big]\end{aligned}$$

Lemma 85. *For $t \in T$, we have:*

$$\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|\Big(\omega_t^{(m)} - \bar{\omega}_t\Big)\Big\|^2 \leq \frac{2L^2}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|x_t^{(m)} - \bar{x}_t\Big\|^2 + \frac{2L^2}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|y_t^{(m)} - \bar{y}_t\Big\|^2 + \frac{2\sigma^2}{b_y} + 2\zeta_g^2$$

Lemma 86. *For $t \in [T]$, we have:*

$$\begin{aligned}&\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|\Big(u_{t+1}^{(m)} - \bar{u}_{t+1}\Big)\Big\|^2 \\ &\leq (1 + \frac{1}{I})\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\Big\|u_t^{(m)} - \bar{u}_t\Big\|^2 + \frac{64I\tau^2C_f^2\zeta_{g,yy}^2}{\mu^2} + 32I\tau^2\zeta_f^2 + \frac{2\tau^2\sigma^2}{b_x}\end{aligned}$$

$$+\Big(\frac{128IL^2\tau^2}{M}+\frac{256I\tau^2L_{y^2}^2C_f^2}{\mu^2M}\Big)\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)}-\bar{x}_t\Big\|^2+\Big\|y_t^{(m)}-\bar{y}_t\Big\|^2\Big]$$

Lemma 84-Lemma 86 bounds the local drift of $\nu_t^{(m)}$, $\omega_t^{(m)}$ and $u_{t+1}^{(m)}$. We provide the proof for Lemma 84 here and the other two bounds can be derived similarly.

Proof. We have:

$$\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\Big\|\Big(\nu_t^{(m)}-\bar{\nu}_t\Big)\Big\|^2\\
&\le\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_x f^{(m)}(x_t^{(m)},y_t^{(m)})-\nabla_{xy}g^{(m)}(x_t^{(m)},y_t^{(m)})u_t^{(m)}\\
&\quad-\frac{1}{M}\sum_{j=1}^{M}\nabla_x f^{(j)}(x_t^{(j)},y_t^{(j)})-\nabla_{xy}g^{(j)}(x_t^{(j)},y_t^{(j)})u_t^{(j)}\Big\|^2+\frac{2\sigma^2}{b_x}\\
&\le\underbrace{\frac{2}{M}\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_x f^{(m)}(x_t^{(m)},y_t^{(m)})-\frac{1}{M}\sum_{j=1}^{M}\nabla_x f^{(j)}(x_t^{(j)},y_t^{(j)})\Big\|^2}_{T_1}\\
&\quad+\underbrace{\frac{2}{M}\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_{xy}g^{(m)}(x_t^{(m)},y_t^{(m)})u_t^{(m)}-\frac{1}{M}\sum_{j=1}^{M}\nabla_{xy}g^{(j)}(x_t^{(j)},y_t^{(j)})u_t^{(j)}\Big\|^2}_{T_2}+\frac{2\sigma^2}{b_x}
\end{aligned}$$

For the term T_1, we have:

$$\begin{aligned}
T_1&\le\frac{16}{M}\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_x f^{(m)}(x_t^{(m)},y_t^{(m)})-\nabla_x f^{(m)}(\bar{x}_t,\bar{y}_t)\Big\|^2\\
&\quad+\frac{4}{M}\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla_x f^{(m)}(\bar{x}_t,\bar{y}_t)-\nabla_x f(\bar{x}_t,\bar{y}_t)\Big\|^2\\
&\le\frac{16L^2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)}-\bar{x}_t\Big\|^2+\Big\|y_t^{(m)}-\bar{y}_t\Big\|^2\Big]+4\zeta_f^2
\end{aligned}$$

Next for the term T_2, we have:

$$T_2 \leq \frac{4L^2}{M}\sum_{m=1}^{M}\mathbb{E}\left\|u_t^{(m)}-\bar{u}_t\right\|^2+\frac{4C_f^2}{\mu^2M^2}\sum_{m=1}^{M}\sum_{j=1}^{M}\mathbb{E}\left\|\nabla_{xy}g^{(m)}(x_t^{(m)},y_t^{(m)})-\nabla_{xy}g^{(j)}(x_t^{(j)},y_t^{(j)})\right\|^2$$
$$\leq \frac{4L^2}{M}\sum_{m=1}^{M}\mathbb{E}\left\|u_t^{(m)}-\bar{u}_t\right\|^2+\frac{32L_{xy}^2C_f^2}{\mu^2M}\sum_{m=1}^{M}\mathbb{E}\left[\left\|x_t^{(m)}-\bar{x}_t\right\|^2+\left\|y_t^{(m)}-\bar{y}_t\right\|^2\right]+\frac{8C_f^2\zeta_{g,xy}^2}{\mu^2}$$

Combine everything together, we get the claim in the lemma. □

Lemma 84-Lemma 86 have recursive dependence of each other. Next, we provide an un-intertwined bound for each of them. For ease of notation, we denote $D_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\nu_t^{(m)}-\bar{\nu}_t\|^2$, $B_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\omega_t^{(m)}-\bar{\omega}_t\|^2$, $A_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|u_t^{(m)}-\bar{u}_t\|^2$ and $C_t=\mathbb{E}\left\|\bar{y}_t-y_{\bar{x}_t}\right\|^2$.

Lemma 87. *For $\gamma\leq\frac{1}{8I\tilde{L}_1}$ and $\eta<\frac{1}{8I\tilde{L}_2}$, $\tau<\frac{1}{128I\tilde{L}_2}$, where $\tilde{L}_1^2=\left(L^2+\frac{2L_{xy}^2C_f^2}{\mu^2}\right)$ and $\tilde{L}_2^2=\left(L^2+\frac{2L_{y^2}^2C_f^2}{\mu^2}\right)$ are constants, then we have:*

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t\leq 96I\zeta_f^2+16I\zeta_g^2+\frac{16IC_f^2\zeta_{g,yy}^2}{\mu^2}+\frac{32IC_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{16I\sigma^2}{b_y}+\frac{20I\sigma^2}{b_x}$$
$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t\leq 24I\zeta_f^2+8I\zeta_g^2+\frac{4IC_f^2\zeta_{g,yy}^2}{\mu^2}+\frac{8IC_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{8I\sigma^2}{b_y}+\frac{5I\sigma^2}{b_x}$$

Proof. Based on Lemma 84, and sum from $\bar{t}_{s-1}+1$ to $\bar{t}_s-1$, we have:

$$\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}D_t\leq 16I\eta^2\tilde{L}_1^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}\sum_{\ell=\bar{t}_{s-1}}^{t-1}D_\ell+16I\gamma^2\tilde{L}_1^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}\sum_{\ell=\bar{t}_{s-1}}^{t-1}B_\ell$$
$$+4L^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}A_t+4(I-1)\zeta_f^2+\frac{8(I-1)C_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{2(I-1)\sigma^2}{b_x}$$
$$\leq 16I^2\eta^2\tilde{L}_1^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}D_t+16I^2\gamma^2\tilde{L}_1^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}B_t$$
$$+4L^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}A_t+4(I-1)\zeta_f^2+\frac{8(I-1)C_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{2(I-1)\sigma^2}{b_x}$$

where we denote $\tilde{L}_1^2 = (L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2})$. Combine with the case of $t = \bar{t}_s$ in Lemma 84, we have:

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} D_t \leq 16I^2\tilde{L}_1^2\eta^2 \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} D_t + 16I^2\tilde{L}_1^2\gamma^2 \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} B_t + 4L^2 \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} A_t + 4I\zeta_f^2 + \frac{8IC_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{2I\sigma^2}{b_x} \tag{4.32}$$

Based on Lemma 85, and sum from $\bar{t}_{s-1}+1$ to $\bar{t}_s - 1$, we have:

$$\begin{aligned}\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1} B_t &\leq 2I\eta^2L^2 \sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1} \sum_{\ell=\bar{t}_{s-1}}^{t-1} D_\ell + 2I\gamma^2L^2 \sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1} \sum_{\ell=\bar{t}_{s-1}}^{t-1} B_\ell + 2(I-1)\sigma^2 + 2(I-1)\zeta_g^2 \\ &\leq 2I^2\eta^2L^2 \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} D_\ell + 2I^2\gamma^2L^2 \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} B_\ell + \frac{2(I-1)\sigma^2}{b_y} + 2(I-1)\zeta_g^2\end{aligned}$$

Combine with the case of $t = \bar{t}_s$ in Lemma 85, we have:

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} B_t \leq 2I^2\eta^2L^2 \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} D_\ell + 2I^2\gamma^2L^2 \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} B_\ell + \frac{2I\sigma^2}{b_y} + 2I\zeta_g^2 \tag{4.33}$$

Apply Lemma 86 recursively, we have:

$$\begin{aligned}A_t &\leq \sum_{\ell=\bar{t}_{s-1}}^{t-1} (1+\frac{1}{I})^{t-\ell}\Big(\frac{64I\tau^2C_f^2\zeta_{g,yy}^2}{\mu^2} + 32I\tau^2\zeta_f^2 + \frac{2\tau^2\sigma^2}{b_x} \\ &\quad + 128I^2\eta^2\tau^2\tilde{L}_2^2 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} D_{\bar{\ell}} + 128I^2\gamma^2\tau^2\tilde{L}_2^2 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} B_{\bar{\ell}}\Big) \\ &\leq \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big(\frac{192I\tau^2C_f^2\zeta_{g,yy}^2}{\mu^2} + 96I\tau^2\zeta_f^2 + \frac{6\tau^2\sigma^2}{b_x} \\ &\quad + 384I^2\eta^2\tau^2\tilde{L}_2^2 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} D_{\bar{\ell}} + 384I^2\gamma^2\tau^2\tilde{L}_2^2 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} B_{\bar{\ell}}\Big)\end{aligned}$$

where we denote $\tilde{L}_2^2 = (L^2 + \frac{2L_{y_2}^2 C_f^2}{\mu^2})$, and the second inequality uses the fact that $t - l \leq I$ and the inequality $log(1 + a/x) \leq a/x$ for $x > -a$, so we have $(1 + a/x)^x \leq e^a$, Then we choose $a = 1$ and $x = I$. Finally, we use the fact that $e^1 \leq 3$. Next, we sum from $\bar{t}_{s-1}$ to $\bar{t}_s - 1$ to have:

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} A_t \leq \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} \Big(\frac{48I^2\tau^2 C_f^2 \zeta_{g,yy}^2}{\mu^2} + 96I^2\tau^2\zeta_f^2 + \frac{6I\tau^2\sigma^2}{b_x} \tag{4.34}$$

$$+ 24I^3\eta^2\tau^2\tilde{L}_2^2 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} D_{\bar{\ell}} + 24I^3\gamma^2\tau^2\tilde{L}_2^2 \sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell-1} B_{\bar{\ell}} \Big)$$

$$\leq \frac{192I^3\tau^2 C_f^2 \zeta_{g,yy}^2}{\mu^2} + 96I^3\tau^2\zeta_f^2 + \frac{6I^2\tau^2\sigma^2}{b_x} \tag{4.35}$$

$$+ 384I^4\eta^2\tau^2\tilde{L}_2^2 \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} D_\ell + 384I^4\gamma^2\tau^2\tilde{L}_2^2 \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} B_\ell \tag{4.36}$$

Combine Eq. 4.32, Eq. 4.33 and Eq. 4.34, and we choose η, γ and τ such that $I^2\gamma^2 L^2 < \frac{1}{4}$, $I^2\gamma^2\tilde{L}_1^2 < \frac{1}{64}$, $I^2\eta^2\tilde{L}_1^2 < \frac{1}{32}$, $I^2\eta^2 L^2 < \frac{1}{16}$, $I^2\tau^2\tilde{L}_2^2 < \frac{1}{128^2}$, $I^2\tau^2 L^2 < \frac{1}{48}$, we get the claim in the lemma, by using the fact that $\tilde{L}_1 > L$ and $\tilde{L}_2 > L$, we get the simplified condition in the lemma.

□

4.7.2.3 Descent Lemma

Lemma 88. *For all $t \in [\bar{t}_{s-1}, \bar{t}_s - 1]$, the iterates generated satisfy:*

$$\mathbb{E}\Big\|\nabla h(\bar{x}_t) - \mathbb{E}_\xi[\bar{\nu}_t]\Big\|^2 \leq \frac{2\tilde{L}_1^2}{M} \sum_{m=1}^{M} \mathbb{E}\Big[\Big\|\bar{x}_t - x_t^{(m)}\Big\|^2$$

$$+ 2\Big\|\bar{y}_t - y_t^{(m)}\Big\|^2 + 2\Big\|y_{\bar{x}_t} - \bar{y}_t\Big\|^2\Big] + 4L^2\mathbb{E}\Big\|u_{\bar{x}_t} - \bar{u}_t\Big\|^2$$

where we denote $u_{\bar{x}_t} = [\nabla_{y^2} g(\bar{x}_t, y_{\bar{x}_t})]^{-1} \nabla_y f(\bar{x}_t, y_{\bar{x}_t})$ *and* $\tilde{L}_1^2 = \left(L^2 + \frac{2L_{xy}^2 C_f^2}{\mu^2}\right)$ *is a constant.*

Proof. By $\nabla h(\bar{x}_t) = \Phi(\bar{x}, y_{\bar{x}})$, we have:

$$\begin{aligned}
\mathbb{E}\left\|\nabla h(\bar{x}_t) - \mathbb{E}_\xi[\bar{\nu}_t]\right\|^2 \leq \mathbb{E}\Big\|\nabla_x f(\bar{x}_t, y_{\bar{x}_t}) - \nabla_{xy} g(\bar{x}_t, y_{\bar{x}_t}) \times [\nabla_{y^2} g(\bar{x}_t, y_{\bar{x}_t})]^{-1} \nabla_y f(\bar{x}_t, y_{\bar{x}_t}) \\
- \frac{1}{M} \sum_{m=1}^{M} \left(\nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}) u_t^{(m)}\right)\Big\|^2 \\
\leq 2\mathbb{E}\Big\|\nabla_x f(\bar{x}_t, y_{\bar{x}_t}) - \frac{1}{M} \sum_{m=1}^{M} \nabla_x f^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2 \\
+ 2\mathbb{E}\Big\|\nabla_{xy} g(\bar{x}_t, y_{\bar{x}_t}) \times [\nabla_{y^2} g(\bar{x}_t, y_{\bar{x}_t})]^{-1} \nabla_y f(\bar{x}_t, y_{\bar{x}_t}) \\
- \frac{1}{M} \sum_{m=1}^{M} \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)}) u_t^{(m)}\Big\|^2
\end{aligned}$$

We denote the two terms above as T_1, T_2 respectively. For the first term T_1, we have:

$$\begin{aligned}
T_1 &\leq \frac{2L^2}{M} \sum_{m=1}^{M} \mathbb{E}\left[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + \left\|y_{\bar{x}_t} - y_t^{(m)}\right\|^2\right] \\
&\leq \frac{2L^2}{M} \sum_{m=1}^{M} \mathbb{E}\left[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + 2\left\|\bar{y}_t - y_t^{(m)}\right\|^2 + 2\left\|y_{\bar{x}_t} - \bar{y}_t\right\|^2\right]
\end{aligned}$$

For the second term T_2, we have:

$$\begin{aligned}
T_2 \leq \frac{4C_f^2}{\mu^2 M} \sum_{m=1}^{M} \mathbb{E}\Big\|\nabla_{xy} g^{(m)}(\bar{x}_t, y_{\bar{x}_t}) - \nabla_{xy} g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2 \\
+ 4L^2 \mathbb{E}\Big\|[\nabla_{y^2} g(\bar{x}_t, y_{\bar{x}})]^{-1} \nabla_y f(\bar{x}_t, y_{\bar{x}_t}) - \bar{u}_t\Big\|^2
\end{aligned}$$

We denote the first term above as $T_{2,1}$. For the term $T_{2,1}$, we have:

$$T_{2,1} \leq \frac{4L_{xy}^2 C_f^2}{\mu^2 M} \sum_{m=1}^{M} \mathbb{E}\left[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + 2\left\|\bar{y}_t - y_t^{(m)}\right\|^2 + 2\left\|y_{\bar{x}_t} - \bar{y}_t\right\|^2\right]$$

Combine everything together, we get the claim in the lemma. □

Lemma 89. *For all $t \in [\bar{t}_{s-1}+1, \bar{t}_s - 1]$ and $s \in [S]$, suppose $\eta < \frac{1}{2\bar{L}}$, the iterates generated satisfy:*

$$\mathbb{E}[h(\bar{x}_{t+1})] \leq \mathbb{E}[h(\bar{x}_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}\|\mathbb{E}_\xi[\nu_t^{(m)}]\|^2 + \frac{\eta^2\bar{L}\sigma^2}{2b_x M} + 2\eta L^2\mathbb{E}\left\|u_{\bar{x}_t} - \bar{u}_t\right\|^2 + \frac{\eta\tilde{L}_1^2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + 2\left\|\bar{y}_t - y_t^{(m)}\right\|^2 + 2\left\|y_{\bar{x}_t} - \bar{y}_t\right\|^2\Big]$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Using the smoothness of f we have:

$$\begin{aligned}
\mathbb{E}[h(\bar{x}_{t+1})] &\leq \mathbb{E}[h(\bar{x}_t)] + \mathbb{E}\langle\nabla h(\bar{x}_t), \bar{x}_{t+1} - \bar{x}_t\rangle + \frac{\bar{L}}{2}\mathbb{E}\|\bar{x}_{t+1} - \bar{x}_t\|^2 \\
&= \mathbb{E}[h(\bar{x}_t)] - \eta\mathbb{E}\langle\nabla h(\bar{x}_t), \mathbb{E}_\xi[\bar{\nu}_t]\rangle + \frac{\eta^2\bar{L}}{2}\mathbb{E}\|\mathbb{E}_\xi[\bar{\nu}_t]\|^2 + \frac{\eta^2\bar{L}\sigma^2}{2b_x M} \\
&\overset{(a)}{=} \mathbb{E}[h(\bar{x}_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t) - \mathbb{E}_\xi[\bar{\nu}_t]\|^2 \\
&\quad - \left(\frac{\eta}{2} - \frac{\eta^2\bar{L}}{2}\right)\|\mathbb{E}_\xi[\bar{\nu}_t]\|^2 + \frac{\eta^2\bar{L}\sigma^2}{2b_x M} \\
&\overset{(b)}{\leq} \mathbb{E}[h(\bar{x}_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}\|\mathbb{E}_\xi[\nu_t^{(m)}]\|^2 + \frac{\eta^2\bar{L}\sigma^2}{2b_x M} \\
&\quad + \frac{\eta\tilde{L}_1^2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\left\|\bar{x}_t - x_t^{(m)}\right\|^2 + 2\left\|\bar{y}_t - y_t^{(m)}\right\|^2 + 2\left\|y_{\bar{x}_t} - \bar{y}_t\right\|^2\Big] + 2\eta L^2\mathbb{E}\left\|u_{\bar{x}_t} - \bar{u}_t\right\|^2
\end{aligned}$$

where equality (a) uses $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a-b\|^2]$; (b) follows the assumption that $\eta < 1/2\bar{L}$ and Lemma 88. □

4.7.2.4 Proof of Convergence Theorem

We first denote the following potential function $\mathcal{G}(t)$:

$$\mathcal{G}_t = \mathbb{E}[h(\bar{x}_t)] + \frac{9\eta\tilde{L}_1^2}{\mu\gamma}\mathbb{E}\left\|\bar{y}_t - y_{\bar{x}_t}\right\|^2 + \frac{9\eta L^2}{\mu\tau}\mathbb{E}\left\|\bar{u}_t - u_{\bar{x}_t}\right\|^2$$

Theorem 90. *Suppose we choose* $\tau = \min(\frac{1}{128I\tilde{L}_2}, \frac{1}{144\kappa L})$, *then denote* $\bar{\gamma} = \min(\frac{1}{8I\tilde{L}_2}, \frac{\tau}{36\kappa\bar{L}}, \frac{1}{4\bar{L}}, \frac{1}{8I\tilde{L}_1})$, *if we choose* $\eta = \frac{\mu\gamma}{36\kappa\tilde{L}_1}$, *and* $\gamma = \min\left(\bar{\gamma}, \left(\frac{\Delta'}{C'_\gamma T}\right)^{1/3}\right)$ *and* $r = \frac{C_f}{\mu}$ *where* Δ' *and* C'_γ *are constants denoted in Eq. 4.38, then we have:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 = O\left(\frac{\kappa^8}{T} + \left(\frac{\kappa^{12}}{T^2}\right)^{1/3} + \frac{\kappa^4\sigma^2}{b_y M} + \frac{\sigma^2}{b_x M}\right)$$

and to reach an ϵ *stationary point, we choose the inner batch size* $b_y = O(M^{-1}\kappa^4\epsilon^{-1})$, *upper batch size* $b_x = O(M^{-1}\epsilon^{-1})$, *and* $T = O(\kappa^6\epsilon^{-1.5})$ *number of iterations.*

Proof. Similar to Lemma 87, we denote $D_t = \frac{1}{M}\sum_{m=1}^{M}\|\nu_t^{(m)} - \bar{\nu}_t\|^2$, $B_t = \frac{1}{M}\sum_{m=1}^{M}\|\omega_t^{(m)} - \bar{\omega}_t\|^2$, $C_t = \mathbb{E}\|\bar{y}_t - y_{\bar{x}_t}\|^2$ and $A_t = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|u_t^{(m)} - \bar{u}_t\|^2$, additionally, we denote $E_t = \|\mathbb{E}_\xi[\bar{\nu}_t]\|^2$ and $F_t = \mathbb{E}\|\bar{u}_t - u_{\bar{x}_t}\|^2$. Combine Lemma 81, Lemma 82 and the definition of the potential function we have:

$$\begin{aligned}\mathcal{G}_{\bar{t}_s} - \mathcal{G}_{\bar{t}_{s-1}} \leq & -\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta L^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}F_t - \frac{\eta\tilde{L}_1^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}C_t \\ & - \frac{\eta}{4}\left(1 - \frac{162\kappa^2\eta^2\tilde{L}_1^2}{\mu^2\gamma^2} - \frac{180\kappa^2\eta^2\bar{L}^2}{\tau^2}\right)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}E_t + \Big(\tilde{L}_1^2 + \frac{81\kappa^2\tilde{L}_1^2}{2} \\ & + \frac{45\kappa^2\tilde{L}_2^2}{16}\Big)I^2\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t\end{aligned}$$

$$+\Big(2\tilde{L}_1^2+\frac{81\kappa^2\tilde{L}_1^2}{2}+\frac{45\kappa^2\tilde{L}_2^2}{16}\Big)I^2\gamma^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t+\frac{81I\kappa^2\tilde{L}_1^2\eta^3\sigma^2}{2\mu^2\gamma^2b_xM}+\frac{36I\tilde{L}_1^2\eta\sigma^2}{\mu^2b_yM}$$
$$+\frac{45I\kappa^2\bar{L}^2\eta^3\sigma^2}{\tau^2b_xM}+\frac{45I\kappa L\tau\eta\sigma^2}{4b_xM}+\frac{\eta^2I\bar{L}\sigma^2}{2b_xM}$$

to bound the coefficients above, we choose $\eta\leq\min\Big(\frac{\mu\gamma}{36\kappa\tilde{L}_1},\frac{\tau}{36\kappa\bar{L}},\frac{1}{4\bar{L}}\Big)$, $\tau<\frac{1}{144\kappa L}$ and we denote $C_1=\Big(2\tilde{L}_1^2+\frac{81\kappa^2\tilde{L}_1^2}{2}+\frac{45\kappa^2\tilde{L}_2^2}{16}\Big)I^2$. Then we have:

$$\mathcal{G}_{\bar{t}_s}-\mathcal{G}_{\bar{t}_{s-1}}\leq-\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2-\frac{\eta L^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}F_t-\frac{\eta\tilde{L}_1^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}C_t-\frac{\eta}{8}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}E_t$$
$$+C_1\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t+C_1\gamma^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t+\frac{I\eta\sigma^2}{2b_xM}+\frac{36I\tilde{L}_1^2\eta\sigma^2}{\mu^2b_yM}$$

Next, we combine with lemma 87 to have:

$$\mathcal{G}_{\bar{t}_s}-\mathcal{G}_{\bar{t}_{s-1}}\leq-\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2-\frac{\eta L^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}F_t-\frac{\eta\tilde{L}_1^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}C_t$$
$$-\frac{\eta}{8}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}E_t+\frac{I\eta\sigma^2}{2b_xM}+\frac{36I\tilde{L}_1^2\eta\sigma^2}{\mu^2b_yM}$$
$$+C_1\eta^3\Big(96I\zeta_f^2+16I\zeta_g^2+\frac{16IC_f^2\zeta_{g,yy}^2}{\mu^2}+\frac{32IC_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{16I\sigma^2}{b_y}+\frac{20I\sigma^2}{b_x}\Big)$$
$$+C_1\gamma^2\eta\Big(24I\zeta_f^2+8I\zeta_g^2+\frac{4IC_f^2\zeta_{g,yy}^2}{\mu^2}+\frac{8IC_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{8I\sigma^2}{b_y}+\frac{5I\sigma^2}{b_x}\Big)$$

Sum over all $s\in[S]$ (assume $T=SI+1$ without loss of generality) to obtain:

$$\frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2$$
$$\leq\mathcal{G}_1-\mathcal{G}_T+\frac{T\eta\sigma^2}{2b_xM}+\frac{36T\tilde{L}_1^2\eta\sigma^2}{\mu^2b_yM}$$
$$+C_1\eta^3\Big(96T\zeta_f^2+16T\zeta_g^2+\frac{16TC_f^2\zeta_{g,yy}^2}{\mu^2}+\frac{32TC_f^2\zeta_{g,xy}^2}{\mu^2}+\frac{16T\sigma^2}{b_y}+\frac{20T\sigma^2}{b_x}\Big)$$

$$
\begin{aligned}
&+ C_1\gamma^2\eta\Big(24T\zeta_f^2 + 8T\zeta_g^2 + \frac{4TC_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{8TC_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{8T\sigma^2}{b_y} + \frac{5T\sigma^2}{b_x}\Big) \\
\leq\ & \Delta + \frac{9\eta\tilde{L}_1^2\Delta_y}{\mu\gamma} + \frac{9\eta L^2\Delta_u}{\mu\tau} + \frac{T\eta\sigma^2}{2b_xM} + \frac{36T\tilde{L}_1^2\eta\sigma^2}{\mu^2 b_yM} \\
&+ C_1\eta^3\Big(96T\zeta_f^2 + 16T\zeta_g^2 + \frac{16TC_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{32TC_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{16T\sigma^2}{b_y} + \frac{20T\sigma^2}{b_x}\Big) \\
&+ C_1\gamma^2\eta\Big(24T\zeta_f^2 + 8T\zeta_g^2 + \frac{4TC_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{8TC_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{8T\sigma^2}{b_y} + \frac{5T\sigma^2}{b_x}\Big)
\end{aligned}
$$

we define $\Delta = h(x_1) - h^*$ as the initial sub-optimality of the function, $\Delta_y = \|y_1 - y_{x_1}\|^2$ as the initial sub-optimality of the inner variable estimation, $\Delta_u = \|u_1 - u_{x_1}\|^2$ as the initial sub-optimality of the hyper-gradient estimation. Then we divide by $\eta T/2$ on both sides and have:

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 \leq\ & \frac{2\Delta}{\eta T} + \frac{18\tilde{L}_1^2\Delta_y}{\mu\gamma T} + \frac{18L^2\Delta_u}{\mu\tau T} + \frac{\sigma^2}{b_xM} + \frac{72\tilde{L}_1^2\sigma^2}{\mu^2 b_yM} \\
&+ 2C_1\eta^2\Big(96\zeta_f^2 + 16\zeta_g^2 + \frac{16C_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{32C_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{16\sigma^2}{b_y} + \frac{20\sigma^2}{b_x}\Big) \\
&+ 2C_1\gamma^2\Big(24\zeta_f^2 + 8\zeta_g^2 + \frac{4C_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{8C_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{8\sigma^2}{b_y} + \frac{5\sigma^2}{b_x}\Big)
\end{aligned}
$$

For ease of notation, we denote constants $C_\eta = 2C_1(96\zeta_f^2 + 16\zeta_g^2 + \frac{16C_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{32C_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{16\sigma^2}{b_y} + \frac{20\sigma^2}{b_x})$ and $C_\gamma = 2C_1(24\zeta_f^2 + 8\zeta_g^2 + \frac{4C_f^2\zeta_{g,yy}^2}{\mu^2} + \frac{8C_f^2\zeta_{g,xy}^2}{\mu^2} + \frac{8\sigma^2}{b_y} + \frac{5\sigma^2}{b_x})$, we have:

$$
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 \leq \frac{2\Delta}{\eta T} + \frac{18\tilde{L}_1^2\Delta_y}{\mu\gamma T} + \frac{18L^2\Delta_u}{\mu\tau T} + \frac{\sigma^2}{b_xM} + \frac{72\tilde{L}_1^2\sigma^2}{\mu^2 b_yM} + C_\eta\eta^2 + C_\gamma\gamma^2 \quad (4.37)
$$

Recall that, we have the condition that $\eta \leq \min\Big(\frac{1}{8I\tilde{L}_2}, \frac{\mu\gamma}{36\kappa\tilde{L}_1}, \frac{\tau}{36\kappa L}, \frac{1}{4L}\Big)$, $\gamma \leq \frac{1}{8I\tilde{L}_1}$, $\tau \leq$

$\min(\frac{1}{128I\tilde{L}_2}, \frac{1}{144\kappa L})$. Suppose we choose $\tau = \min(\frac{1}{128I\tilde{L}_2}, \frac{1}{144\kappa L})$, then denote

$$\bar{\gamma} = \min(\frac{1}{8I\tilde{L}_2}, \frac{\tau}{36\kappa\bar{L}}, \frac{1}{4\bar{L}}, \frac{1}{8I\tilde{L}_1}),$$

and let $\gamma \leq \bar{\gamma}$, and $\eta = \frac{\mu\gamma}{36\kappa\tilde{L}_1}$, then we have:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 \leq \frac{72\kappa\tilde{L}_1\Delta + 18\tilde{L}_1^2\Delta_y}{\mu\gamma T} + \Big(\frac{C_\eta\mu^2}{36^2\kappa^2\tilde{L}_1^2} + C_\gamma\Big)\gamma^2 + \frac{18L^2\Delta_u}{\mu\tau T} + \frac{\sigma^2}{b_x M} + \frac{72\tilde{L}_1^2\sigma^2}{\mu^2 b_y M}$$

We denote

$$C_\gamma^{'} = \Big(\frac{C_\eta\mu^2}{36^2\kappa^2\tilde{L}_1^2} + C_\gamma\Big),\ \Delta^{'} = \frac{72\kappa\tilde{L}_1\Delta + 18\tilde{L}_1^2\Delta_y}{\mu}, \tag{4.38}$$

then we choose γ as:

$$\gamma = \min\Big(\bar{\gamma}, \Big(\frac{\Delta^{'}}{C_\gamma^{'}T}\Big)^{1/3}\Big)$$

and obtain:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 \leq \frac{\Delta^{'}}{\bar{\gamma}T} + \Big(\frac{C_\gamma^{'}(\Delta^{'})^2}{T^2}\Big)^{1/3} + \frac{18L^2\Delta_u}{\mu\tau T} + \frac{\sigma^2}{b_x M} + \frac{72\tilde{L}_1^2\sigma^2}{\mu^2 b_y M}$$

Finally, since $\tilde{L}_1 = O(\kappa)$, $\bar{L} = O(\kappa^3)$, suppose we choose $I = O(1)$, then we have $\tau^{-1} = O(\kappa)$, $\bar{\gamma}^{-1} = O(\kappa^5)$, $\Delta^{'} = O(\kappa^3)$, $C_1 = O(\kappa^4)$, $C_\eta = O(\kappa^6)$, $C_\gamma = O(\kappa^6)$, $C_\gamma^{'} = O(\kappa^6)$ then we have:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 = O\left(\frac{\kappa^8}{T} + \Big(\frac{\kappa^{12}}{T^2}\Big)^{1/3} + \frac{\kappa^4\sigma^2}{b_y M} + \frac{\sigma^2}{b_x M}\right)$$

and to reach an ϵ stationary point, we choose the inner batch size $b_y = O(M^{-1}\kappa^4\epsilon^{-1})$, upper batch size $b_x = O(M^{-1}\epsilon^{-1})$, and $T = O(\kappa^6\epsilon^{-1.5})$ number of iterations. $\square$

Algorithm 9 FedBiO- Local Lower Level Problem

Input: Initial states x_1, y_1; learning rates $\{\gamma_t, \eta_t\}_{t=1}^T$
Initialization: Set $x_1^{(m)} = x_1$, $y_1^{(m)} = y_1$;
for $t = 1$ **to** T **do**
  Randomly sample mutually independent minibatch of samples $\mathcal{B}_y$ and $\mathcal{B}_x$ of size b;
  $\omega_t^{(m)} = \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y)$ and $\nu_t^{(m)} = \Phi^{(m)}(x^{(m)}, y^{(m)}; \mathcal{B}_x)$;
  $\hat{y}_{t+1}^{(m)} = y_t^{(m)} - \gamma_t \omega_t^{(m)}$, $\hat{x}_{t+1}^{(m)} = x_t^{(m)} - \eta_t \nu_t^{(m)}$;
  if t mod I $= 0$ **then**
    $y_{t+1}^{(m)} = \hat{y}_{t+1}^{(m)}$, $x_{t+1}^{(m)} = \frac{1}{M}\sum_{j=1}^M \hat{x}_{t+1}^{(j)}$
  else
    $y_{t+1}^{(m)} = \hat{y}_{t+1}^{(m)}$, $x_{t+1}^{(m)} = \hat{x}_{t+1}^{(m)}$
  end if
end for

4.8 Proof for Local Lower Level Problem

The FedBiOAcc-Local and FedBiO-Local are presented in Algorithm 10 and Algorithm 9, respectively. Then in this section, we discuss the convergence rate of the two algorithms. Please see Theorem 102 and Theorem 109 for the convergence rates.

For Eq. (4.10), we also assume Assumptions 55 -58, with a slightly different assumption to the heterogeneity as follows:

Assumption 91. *For any $m, j \in [M]$ and $z = (x, y)$, we have: $\|\nabla f^{(m)}(z) - \nabla f^{(j)}(z)\| \leq \zeta_f$, $\|\nabla_{xy} g^{(m)}(z) - \nabla_{xy} g^{(j)}(z)\| \leq \zeta_{g,xy}$, $\|\nabla_{y^2} g^{(m)}(z) - \nabla_{y^2} g^{(j)}(z)\| \leq \zeta_{g,yy}$, $\|y_x^{(m)} - y_x^{(j)}\| \leq \zeta_{g^*}$, where ζ_f, $\zeta_{g,xy}$, $\zeta_{g,yy}$, ζ_{g^*} are constants.*

Note that we remove the requirement of gradient dissimilarity ζ_g in Assumption 59 and add the dissimilarity bound ζ_{g^*} for the minimizer of the lower level problem. Note that Assumption 91 is a sufficient condition such that the dissimilarity of local hyper-gradient is bounded by some constant ζ.

Proposition 92. *(Lemma 4 and 7 in [32]) Suppose Assumptions 56, 57 and 58 hold and*

Algorithm 10 FedBiOAcc - Local Lower Level Problem

Input: Constants c_ω, c_ν, γ, η; learning rate schedule $\{\alpha_t\}$, $t \in [T]$, initial state (x_1, y_1);
Initialization: Set $y_1^{(m)} = y_1$, $x_1^{(m)} = x_1$, $\omega_1^{(m)} = \nabla_y g^{(m)}(x_1, y_1, \mathcal{B}_y)$, $\nu_1^{(m)} = \Phi^{(m)}(x_1, y_1; \mathcal{B}_x)$ for $m \in [M]$
for $t = 1$ **to** T **do**
  $\hat{y}_{t+1}^{(m)} = y_t^{(m)} - \gamma\alpha_t\omega_t^{(m)}$, $\hat{x}_{t+1}^{(m)} = x_t^{(m)} - \eta\alpha_t\nu_t^{(m)}$, $\hat{u}_{t+1}^{(m)} = u_t^{(m)} - \tau\alpha_t q_t^{(m)}$
  if t mod I $= 0$ **then**
    $y_{t+1}^{(m)} = \hat{y}_{t+1}^{(m)}$, $x_{t+1}^{(m)} = \frac{1}{M}\sum_{j=1}^M \hat{x}_{t+1}^{(j)}$
  else
    $y_{t+1}^{(m)} = \hat{y}_{t+1}^{(m)}$, $x_{t+1}^{(m)} = \hat{x}_{t+1}^{(m)}$,
  end if
  Randomly sample minibatches $\mathcal{B}_y$ and $\mathcal{B}_x$
  $\hat{\omega}_{t+1}^{(m)} = \nabla_y g^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}, \mathcal{B}_y) + (1 - c_\omega\alpha_t^2)(\omega_t^{(m)} - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)}, \mathcal{B}_y))$
  $\hat{\nu}_{t+1}^{(m)} = \Phi^{(m)}(x_{t+1}^{(m)}, y_{t+1}^{(m)}; \mathcal{B}_x) + (1 - c_\nu\alpha_t^2)(\nu_t^{(m)} - \Phi^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_x))$
  if t mod I $= 0$ **then**
    $\omega_{t+1}^{(m)} = \hat{\omega}_{t+1}^{(m)}$, $\nu_{t+1}^{(m)} = \frac{1}{M}\sum_{j=1}^M \hat{\nu}_{t+1}^{(j)}$,
  else
    $\omega_{t+1}^{(m)} = \hat{\omega}_{t+1}^{(m)}$, $\nu_{t+1}^{(m)} = \hat{\nu}_{t+1}^{(m)}$
  end if
end for

$\tau < \frac{1}{L}$, the hypergradient estimator $\Phi(x, y; \mathcal{B}_x)$ w.r.t. x based on a minibatch $\mathcal{B}_x$ has bounded variance and bias:

a) $\|\mathbb{E}[\Phi^{(m)}(x, y; \mathcal{B}_x)] - \Phi^{(m)}(x, y)\| \leq G_1$, *where* $G_1 = \kappa(1 - \tau\mu)^{Q+1}C_f$

b) $\mathbb{E}\|\Phi^{(m)}(x, y; \mathcal{B}_x) - \mathbb{E}[\Phi^{(m)}(x, y; \mathcal{B}_x)]\|^2 \leq G_2^2$, *where* $G_2^2 = (2C_f^2 + 12C_f^2L^2\tau^2(Q+1)^2 + 4C_f^2L^2(Q+2)(Q+1)^2\tau^4\sigma^2)/b_x$

Proposition 93. *Suppose Assumptions 56 and 57 hold, the following statements hold:*

a) $y_x^{(m)}$ *is Lipschitz continuous in* x *with constant* $\rho = \kappa$, *where* $\kappa = \frac{L}{\mu}$ *is the condition number of* $g^{(m)}(x, y)$.

b) $\|\Phi^{(m)}(x_1; y_1) - \Phi^{(m)}(x_2; y_2)\|^2 \leq \hat{L}^2(\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2)$, *where* $\hat{L} = O(\kappa^2)$.

d) $h^{(m)}(x)$ *is Lipschitz continuous in* x *with constant* $\bar{L}$ *i.e., for any given* $x_1, x_2 \in X$, *we have* $\|\nabla h^{(m)}(x_2) - \nabla h^{(m)}(x_1)\| \leq \bar{L}\|x_2 - x_1\|$ *where* $\bar{L} = O(\kappa^3)$.

This is a standard results in bilevel optimization and we omit the proof here.

Proposition 94. *In Eq. 4.10, suppose Assumption 55, 56, 57, 91 hold, we have:*

$$\|\nabla h^{(m)}(x) - \nabla h^{(j)}(x)\| \leq (1+\kappa)\zeta_f + \frac{C_f}{\mu}\zeta_{g,xy} + \frac{\kappa C_f}{\mu}\zeta_{g,yy} + \Big((1+\kappa)L + \frac{C_f L_{xy}}{\mu} + \frac{\kappa C_f L_{y^2}}{\mu}\Big)\zeta_{g^*} := \zeta$$

Proof. For $h^{(m)}(x) = f^{(m)}(x, y_x^{(m)}), m \in [M]$ in Eq. 4.10, we have:

$$\begin{aligned}
&\|\nabla h^{(m)}(x) - \nabla h^{(j)}(x)\| \\
&= \|\nabla_x f^{(m)}(x, y_x^{(m)}) - \nabla_{xy} g^{(m)}(x, y_x^{(m)})[\nabla_{y^2} g^{(m)}(x, y_x^{(m)})]^{-1}\nabla_y f^{(m)}(x, y_x^{(m)}) \\
&\quad - \Big(\nabla_x f^{(j)}(x, y_x^{(j)}) - \nabla_{xy} g^{(j)}(x, y_x^{(j)})[\nabla_{y^2} g^{(j)}(x, y_x^{(j)})]^{-1}\nabla_y f^{(j)}(x, y_x^{(j)})\Big)\| \\
&\leq \Big\|\nabla_x f^{(m)}(x, y_x^{(m)}) - \nabla_x f^{(j)}(x, y_x^{(j)})\Big\| + \Big\|\nabla_{xy} g^{(m)}(x, y_x^{(m)}) \\
&\quad - \nabla_{xy} g^{(j)}(x, y_x^{(j)})\Big\|\Big\|\Big(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\Big)^{-1}\nabla_y f^{(m)}(x, y_x^{(m)})\Big\| \\
&\quad + \Big\|\nabla_{xy} g^{(j)}(x, y_x^{(j)})\Big\|\Big\|\Big(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\Big)^{-1}\nabla_y f^{(m)}(x, y_x^{(m)}) \\
&\quad - \Big(\nabla_{yy} g^{(j)}(x, y_x^{(j)})\Big)^{-1}\nabla_y f^{(j)}(x, y_x^{(j)})\Big\|
\end{aligned}$$

Next we bound the three terms separately. For the first term:

$$\Big\|\nabla_x f^{(m)}(x, y_x^{(m)}) - \nabla_x f^{(j)}(x, y_x^{(j)})\Big\| \leq \Big\|\nabla_x f^{(m)}(x, y_x^{(m)}) - \nabla_x f^{(j)}(x, y_x^{(m)})\Big\| + \Big\|\nabla_x f^{(j)}(x, y_x^{(m)}) - \nabla_x f^{(j)}(x, y_x^{(j)})\Big\|$$

$$\leq \zeta_f + L\left\|y_x^{(m)} - y_x^{(j)}\right\| \leq \zeta_f + L\zeta_{g^*} \tag{4.39}$$

where the second inequality is due to Assumption 56 and Assumption 91. The last inequality also follows the Assumption 91. Next, for the second term, we have:

$$\begin{aligned}
&\left\|\nabla_{xy} g^{(m)}(x, y_x^{(m)}) - \nabla_{xy} g^{(j)}(x, y_x^{(j)})\right\| \left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1} \nabla_y f^{(m)}(x, y_x^{(m)})\right\| \\
&\leq \frac{C_f}{\mu}\left\|\nabla_{xy} g^{(m)}(x, y_x^{(m)}) - \nabla_{xy} g^{(j)}(x, y_x^{(j)})\right\| \\
&\leq \frac{C_f}{\mu}\left\|\nabla_{xy} g^{(m)}(x, y_x^{(m)}) - \nabla_{xy} g^{(j)}(x, y_x^{(m)})\right\| + \frac{C_f}{\mu}\left\|\nabla_{xy} g^{(j)}(x, y_x^{(m)}) - \nabla_{xy} g^{(j)}(x, y_x^{(j)})\right\| \\
&\leq \frac{C_f \zeta_{g,xy}}{\mu} + \frac{C_f L_{xy}}{\mu}\left\|y_x^{(m)} - y_x^{(j)})\right\| \leq \frac{C_f \zeta_{g,xy}}{\mu} + \frac{C_f L_{xy} \zeta_{g^*}}{\mu}
\end{aligned}$$

where the first inequality follows from the Assumption 55, 56; the third inequality follows from Assumption 91, 57, the last inequality follows from Assumption 91. Next, for the third term, we have:

$$\begin{aligned}
&\left\|\nabla_{xy} g^{(j)}(x, y_x^{(j)})\right\| \left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1} \nabla_y f^{(m)}(x, y_x^{(m)}) - \left(\nabla_{yy} g^{(j)}(x, y_x^{(j)})\right)^{-1} \nabla_y f^{(j)}(x, y_x^{(j)})\right\| \\
&\leq L\left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1} \nabla_y f^{(m)}(x, y_x^{(m)}) - \left(\nabla_{yy} g^{(j)}(x, y_x^{(j)})\right)^{-1} \nabla_y f^{(j)}(x, y_x^{(j)})\right\| \\
&\leq L\left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1}\right\| \left\|\nabla_y f^{(m)}(x, y_x^{(m)}) - \nabla_y f^{(j)}(x, y_x^{(j)})\right\| \\
&\quad + L\left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1} - \left(\nabla_{yy} g^{(j)}(x, y_x^{(j)})\right)^{-1}\right\| \left\|\nabla_y f^{(j)}(x, y_x^{(j)})\right\| \\
&\leq \frac{L}{\mu}\left\|\nabla_y f^{(m)}(x, y_x^{(m)}) - \nabla_y f^{(j)}(x, y_x^{(j)})\right\| \\
&\quad + C_f L\left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1} - \left(\nabla_{yy} g^{(j)}(x, y_x^{(j)})\right)^{-1}\right\| \\
&\leq \frac{L(\zeta_f + L\zeta_{g^*})}{\mu} + C_f L\left\|\left(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\right)^{-1}\right\| \times
\end{aligned}$$

$$\left\|\nabla_{yy} g^{(m)}(x, y_x^{(m)}) - \nabla_{yy} g^{(j)}(x, y_x^{(j)})\right\| \left\|\left(\nabla_{yy} g^{(j)}(x, y_x^{(j)})\right)^{-1}\right\|$$
$$\leq \frac{L(\zeta_f + L\zeta_{g^*})}{\mu} + \frac{C_f L(\zeta_{g,yy} + L_{y^2}\zeta_{g^*})}{\mu^2}$$

where the first inequality is by Assumption 57; the third inequality is by Assumption 57, 56; the fourth inequality is by Cauchy Schwartz inequality; the last inequality is by Assumption 55, 57 and the result in Eq. 4.39. Combine everything together, we have:

$$\left\|\nabla_x f^{(m)}(x, y_x^{(m)}) - \nabla_x f^{(j)}(x, y_x^{(j)})\right\| \leq \zeta_f + L\zeta_{g^*} + \frac{C_f \zeta_{g,xy}}{\mu} + \frac{C_f L_{xy}\zeta_{g^*}}{\mu} + \frac{L(\zeta_f + L\zeta_{g^*})}{\mu} + \frac{C_f L(\zeta_{g,yy} + L_{y^2}\zeta_{g^*})}{\mu^2}$$

which completes the proof. □

4.8.1 Proof for the FedBiOAcc-Local Algorithm

4.8.1.1 Hyper-Gradient Bias and Inner-Gradient Bias

Lemma 95. *Suppose we have $c_\nu \alpha_t^2 < 1$, then:*

$$\mathbb{E}\left[\left\|\bar{\nu}_t - \mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}]\right\|^2\right] \leq (1 - c_\nu\alpha_{t-1}^2)\mathbb{E}\left[\left\|\bar{\nu}_{t-1} - \mathbb{E}_\xi[\bar{\mu}_{t-1,\mathcal{B}_x}]\right\|^2\right] + \frac{2c_\nu^2\alpha_{t-1}^4}{b_x M}G_2^2 + \frac{2\hat{L}^2}{b_x M^2}\sum_{m=1}^{M}\mathbb{E}\left[\left\|x_t^{(m)} - x_{t-1}^{(m)}\right\|^2 + \left\|y_t^{(m)} - y_{t-1}^{(m)}\right\|^2\right]$$

where $\mu_{t,\xi}^{(m)} = \Phi^{(m)}(x_t^{(m)}, y_t^{(m)}; \mathcal{B}_x)$ and the expectation outside is w.r.t *all the stochasity of the algorithm.*

Proof. For ease of notation, we denote $\mu_{t,\xi}^{(m)} = \Phi^{(m)}(x_t^{(m)}, y_t^{(m)}; \xi_x)$, and $\mu_t^{(m)} = \Phi^{(m)}(x_t^{(m)}, y_t^{(m)})$,

then by the definition of $\bar{\nu}_t$ we have:

$$
\begin{aligned}
&\mathbb{E}\Big[\Big\|\bar{\nu}_t - \mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}]\Big\|^2\Big] = \mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{m=1}^{M}\Big(\hat{\nu}_t^{(m)} - \mathbb{E}_\xi[\mu_{t,\mathcal{B}_x}^{(m)}]\Big)\Big\|^2\Big] \\
&= \mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{m=1}^{M}\Big(\mu_{t,\mathcal{B}_x}^{(m)} + (1-c_\nu\alpha_{t-1}^2)(\nu_{t-1}^{(m)} - \mu_{t-1,\mathcal{B}_x}^{(m)}) - \mathbb{E}_\xi[\mu_{t,\mathcal{B}_x}^{(m)}]\Big)\Big\|^2\Big] \\
&= \mathbb{E}\Big[\Big\|(1-c_\nu\alpha_{t-1}^2)\Big(\bar{\nu}_{t-1} - \mathbb{E}_\xi[\bar{\mu}_{t-1,\mathcal{B}_x}]\Big) \\
&\qquad + \Big(\bar{\mu}_{t,\mathcal{B}_x} - \mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}] + (1-c_\nu\alpha_{t-1}^2)(\mathbb{E}_\xi[\bar{\mu}_{t-1,\mathcal{B}_x}] - \bar{\mu}_{t-1,\mathcal{B}_x})\Big)\Big\|^2\Big] \\
&\leq (1-c_\nu\alpha_{t-1}^2)\mathbb{E}\Big[\Big\|\bar{\nu}_{t-1} - \mathbb{E}_\xi[\bar{\mu}_{t-1,\mathcal{B}_x}]\Big\|^2\Big] \\
&\qquad + \frac{1}{b_x^2M^2}\sum_{m=1}^{M}\sum_{\xi_x\in\mathcal{B}_x}\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mathbb{E}_\xi[\mu_{t,\xi_x}^{(m)}] + (1-c_\nu\alpha_{t-1}^2)(\mathbb{E}_\xi[\mu_{t-1,\xi_x}^{(m)}] - \mu_{t-1,\xi_x}^{(m)})\Big\|^2\Big]
\end{aligned}
$$

where inequality (a) uses the fact that the cross product term is zero in expectation, the condition that $c_\nu\alpha_t^2 < 1$ and the fact that clients independently choose samples.

We denote the second term above as T_1, then we have:

$$
\begin{aligned}
T_1 &\overset{(a)}{\leq} 2(c_\nu\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mathbb{E}_\xi[\mu_{t,\xi_x}^{(m)}]\Big\|^2\Big] \\
&\qquad + 2(1-c_\nu\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mu_{t-1,\xi_x}^{(m)} - (\mathbb{E}_\xi[\mu_t^{(m)}] - \mathbb{E}_\xi[\mu_{t-1}^{(m)}])\Big\|^2\Big] \\
&\overset{(b)}{\leq} 2(c_\nu\alpha_{t-1}^2)^2\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mathbb{E}_\xi[\mu_t^{(m)}]\Big\|^2\Big] + 2\mathbb{E}\Big[\Big\|\mu_{t,\xi_x}^{(m)} - \mu_{t-1,\xi_x}^{(m)}\Big\|^2\Big] \\
&\overset{(c)}{\leq} 2(c_\nu\alpha_{t-1}^2)^2G_2^2 + 2\hat{L}^2\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}
$$

where inequality (a) follows the generalized triangle inequality; (b) follows Proposition 111 due to the definition of $\mu_t^{(m)}$; (c) follows the smoothness property of $\hat{L}$ and the bounded variance assumption; This completes the proof. □

Lemma 96. *Suppose we have $c_\omega \alpha_{t-1}^2 < 1$, then we have:*

$$\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\omega_t^{(m)} - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2\Big] \\
&\leq \frac{(1-c_\omega\alpha_{t-1}^2)}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\omega_{t-1}^{(m)} - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)})\Big\|^2\Big] + \frac{2(c_\omega\alpha_{t-1}^2)^2\sigma^2}{b_y} \\
&\quad + \frac{2L^2}{b_y M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|x_t^{(m)} - x_{t-1}^{(m)}\Big\|^2 + \Big\|y_t^{(m)} - y_{t-1}^{(m)}\Big\|^2\Big]
\end{aligned}$$

where the expectation is w.r.t the stochasticity of the algorithm.

The proof of Lemma 96 can be derived similar as Lemma 95

4.8.1.2 Lower Problem Solution Error

Lemma 97. *Suppose we choose $\gamma \leq \frac{1}{2L}$ and $\alpha_t < 1$. Then for $t \neq \bar{t}_s$, we have:*

$$\begin{aligned}
\mathbb{E}\Big[\Big\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\Big\|^2\Big] \leq (1 - \frac{\mu\gamma\alpha_{t-1}}{4})\mathbb{E}\Big[\Big\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\Big\|^2\Big] - \frac{\gamma^2\alpha_{t-1}}{4}\mathbb{E}\Big[\Big\|\omega_{t-1}^{(m)}\Big\|^2\Big] \\
+ \frac{9\gamma\alpha_{t-1}}{2\mu}\mathbb{E}\Big[\Big\|\omega_{t-1}^{(m)} - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)})\Big\|^2\Big] + \frac{9\kappa^2\eta^2\alpha_{t-1}}{2\mu\gamma}\mathbb{E}\Big[\|\nu_{t-1}^{(m)}\|^2\Big]
\end{aligned}$$

for $t = \bar{t}_s$, we have:

$$\begin{aligned}
\mathbb{E}\Big[\Big\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\Big\|^2\Big] \leq (1 - \frac{\mu\gamma\alpha_{t-1}}{4})\mathbb{E}\Big[\Big\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\Big\|^2\Big] - \frac{\gamma^2\alpha_{t-1}}{4}\mathbb{E}\Big[\Big\|\omega_{t-1}^{(m)}\Big\|^2\Big] \\
+ \frac{9\gamma\alpha_{t-1}}{2\mu}\mathbb{E}\Big[\Big\|\omega_{t-1}^{(m)} - \nabla_y g^{(m)}(x_{t-1}^{(m)}, y_{t-1}^{(m)})\Big\|^2\Big] \\
+ \frac{9\kappa^2\eta^2\alpha_{t-1}}{\mu\gamma}\mathbb{E}\Big[\|\nu_{t-1}^{(m)}\|^2\Big] + \frac{9\kappa^2}{\mu\gamma\alpha_{t-1}}\mathbb{E}\Big[\|\hat{x}_t^{(m)}) - \bar{x}_t\|^2\Big]
\end{aligned}$$

Proof. First, we exploit Proposition 114, and choose the function $g^{(m)}(x_t^{(m)}, \cdot)$, by assump-

tion it is L smooth and μ strongly convex, and we choose $\gamma < \frac{1}{2L}$ and $\alpha_t < 1$, thus:

$$\|y_{t+1}^{(m)} - y_{x_t^{(m)}}^{(m)}\|^2 \leq (1 - \frac{\mu\gamma\alpha_t}{2})\|y_{x_t^{(m)}}^{(m)} - y_t^{(m)}\|^2 - \frac{\gamma^2\alpha_t}{4}\|\omega_t^{(m)}\|^2 + \frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(x_t^{(m)}, y_t^{(m)}) - w_t^{(m)}\|^2. \tag{4.40}$$

Next, we decompose the term $\|y_{t+1}^{(m)} - y_{x_{t+1}^{(m)}}^{(m)}\|^2$ as follows:

$$\begin{aligned}\|y_{t+1}^{(m)} - y_{x_{t+1}^{(m)}}^{(m)}\|^2 &\leq (1 + \frac{\mu\gamma\alpha_t}{4})\|y_{t+1}^{(m)} - y_{x_t^{(m)}}^{(m)}\|^2 + (1 + \frac{4}{\mu\gamma\alpha_t})\|y_{x_t^{(m)}}^{(m)} - y_{x_{t+1}^{(m)}}^{(m)})\|^2 \\ &\leq (1 + \frac{\mu\gamma\alpha_t}{4})\|y_{t+1}^{(m)} - y_{x_t^{(m)}}^{(m)}\|^2 + (1 + \frac{4}{\mu\gamma\alpha_t})\kappa^2\|x_t^{(m)} - x_{t+1}^{(m)}\|^2 \end{aligned} \tag{4.41}$$

where the first inequality holds by the generalized triangle inequality, and the second inequality is due to case a) of Proposition 3.9. Combining the above inequalities 4.40 and 4.41, we have

$$\begin{aligned}\|y_{t+1}^{(m)} - y_{\hat{x}_{t+1}^{(m)}}^{(m)})\|^2 \leq (1 + \frac{\mu\gamma\alpha_t}{4})(1 - \frac{\mu\gamma\alpha_t}{2})\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)})\|^2 - (1 + \frac{\mu\gamma\alpha_t}{4})\frac{\gamma^2\alpha_t}{4}\|\omega_t^{(m)}\|^2 \\ + (1 + \frac{\mu\gamma\alpha_t}{4})\frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(x_t^{(m)}, y_t^{(m)}) - w_t^{(m)}\|^2 + (1 + \frac{4}{\mu\gamma\alpha_t})\kappa^2\|x_t^{(m)} - x_{t+1}^{(m)}\|^2\end{aligned}$$

Since we choose $\gamma \leq \frac{1}{2L}$, $\alpha_t < 1$, we have:

$$(1 + \frac{\mu\gamma\alpha_t}{4})(1 - \frac{\mu\gamma\alpha_t}{2}) = 1 - \frac{\mu\gamma\alpha_t}{4} - \frac{\mu^2\gamma^2\alpha_t^2}{8} \leq 1 - \frac{\mu\gamma\alpha_t}{4}$$

and $-(1 + \frac{\mu\gamma\alpha_t}{4}) \leq -1, (1 + \frac{\mu\gamma\alpha_t}{4}) \leq \frac{9}{8}$, $\mu\gamma\alpha_t < \frac{1}{2}$. Thus, we have

$$\|y_{t+1}^{(m)} - y_{\hat{x}_{t+1}^{(m)}}^{(m)})\|^2 \leq \left(1 - \frac{\mu\gamma\alpha_t}{4}\right)\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)})\|^2 - \frac{\gamma^2\alpha_t}{4}\|\omega_t^{(m)}\|^2$$

$$+\frac{9\gamma\alpha_t}{2\mu}\|\nabla_y g(x_t^{(m)},y_t^{(m)})-w_t\|^2+\frac{9\kappa^2}{2\mu\gamma\alpha_t}\underbrace{\|x_t^{(m)}-x_{t+1}^{(m)}\|^2}_{T_1}$$

Note for the term T_1 we have $T_1=\left\|\eta\alpha_t\nu_t^{(m)}\right\|^2$ for $t+1\neq\bar{t}_s$ and $T_1=\left\|\bar{x}_{t+1}-x_t^{(m)}\right\|^2\leq 2\left\|\hat{x}_{t+1}^{(m)}-\bar{x}_{t+1}\right\|^2+2\left\|\eta\alpha_t\nu_t^{(m)}\right\|^2$ for $t+1=\bar{t}_s$. This completes the proof. □

4.8.1.3 Upper Variable Drift

Lemma 98. *For $t\in[\bar{t}_{s-1}+1,\bar{t}_s]$, with $s\in[S]$ we have:*

$$\|\hat{x}_t^{(m)}-\bar{x}_t\|^2\leq\sum_{\ell=\bar{t}_{s-1}}^{t-1}I\eta^2\alpha_l^2\left\|\left(\nu_\ell^{(m)}-\bar{\nu}_\ell\right)\right\|^2$$

Proof. Since we have $\hat{x}_t^{(m)}=x_{t-1}^{(m)}-\eta\alpha_{t-1}\nu_{t-1}^{(m)}$, this implies that:

$$\hat{x}_t^{(m)}=x_{\bar{t}_{s-1}}^{(m)}-\sum_{\ell=\bar{t}_{s-1}}^{t-1}\eta\alpha_\ell\nu_\ell^{(m)}\quad\text{and}\quad\bar{x}_t=\bar{x}_{\bar{t}_{s-1}}-\sum_{\ell=\bar{t}_{s-1}}^{t-1}\eta\alpha_\ell\bar{\nu}_\ell.$$

So for $t\in[\bar{t}_{s-1}+1,\bar{t}_s]$, with $s\in[S]$ we have:

$$\begin{aligned}\|\hat{x}_t^{(m)}-\bar{x}_t\|^2&=\left\|x_{\bar{t}_{s-1}}^{(m)}-\bar{x}_{\bar{t}_{s-1}}-\Big(\sum_{\ell=\bar{t}_{s-1}}^{t-1}\eta\alpha_\ell\nu_\ell^{(m)}-\sum_{\ell=\bar{t}_{s-1}}^{t-1}\eta\alpha_\ell\bar{\nu}_\ell\Big)\right\|^2\\&\overset{(a)}{=}\Big\|\sum_{\ell=\bar{t}_{s-1}}^{t-1}\eta\alpha_\ell\Big(\nu_\ell^{(m)}-\bar{\nu}_\ell\Big)\Big\|^2\overset{(b)}{\leq}\sum_{\ell=\bar{t}_{s-1}}^{t-1}I\eta^2\alpha_l^2\left\|\left(\nu_\ell^{(m)}-\bar{\nu}_\ell\right)\right\|^2\end{aligned}$$

where the equality (a) follows from the fact that $x_{\bar{t}_{s-1}}^{(m)}=\bar{x}_{\bar{t}_{s-1}}$; inequality (b) is due to $t-\bar{t}_{s-1}\leq I$ and the generalized triangle inequality. □

Lemma 99. *Suppose $\alpha_t < \frac{1}{16I\hat{L}}$, $\eta < 1$, then for $t \neq \bar{t}_s, s \in [S]$, we have:*

$$\begin{aligned}\sum_{m=1}^{M}\mathbb{E}\|\nu_t^{(m)}-\bar{\nu}_t\|^2 \leq \Big(1+\frac{33}{32I}\Big)\sum_{m=1}^{M}\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2 + 4I\hat{L}^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[2\|\eta\bar{\nu}_{t-1}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big]\\ + 8IM(c_\nu\alpha_{t-1}^2)^2G_1^2 + \frac{8IM(c_\nu\alpha_{t-1}^2)^2G_2^2}{b_x} + 16IM(c_\nu\alpha_{t-1}^2)^2\zeta^2\\ + 128I\hat{L}^2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|y_{t-1}^{(m)}-y_{x_{t-1}^{(m)}}^{(m)}\|^2\Big]\\ + 128I\bar{L}^2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\sum_{\ell=\bar{t}_{s-1}}^{t-2}I\eta^2\alpha_l^2\mathbb{E}\Big\|\Big(\nu_\ell^{(m)}-\bar{\nu}_\ell\Big)\Big\|^2\end{aligned}$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. By the update step in Line 7 of Algorithm 7, for $t \neq \bar{t}_s$, we have:

$$\begin{aligned}&\mathbb{E}\|\hat{\nu}_t^{(m)}-\bar{\nu}_t\|^2 \qquad (4.42)\\ &= \mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)} + (1-c_\nu\alpha_{t-1}^2)\Big(\nu_{t-1}^{(m)}-\mu_{t-1,\mathcal{B}_x}^{(m)}\Big) - \Big(\bar{\mu}_{t,\mathcal{B}_x}+(1-c_\nu\alpha_{t-1}^2)\Big(\bar{\nu}_{t-1}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big)\Big\|^2\\ &= \mathbb{E}\Big\|(1-c_\nu\alpha_{t-1}^2)\Big(\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\Big)+\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2\\ &\overset{(a)}{\leq} (1+\frac{1}{I})(1-c_\nu\alpha_{t-1}^2)^2\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2\\ &\quad + \Big(1+I\Big)\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2\\ &\leq \Big(1+\frac{1}{I}\Big)\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2\\ &\quad + \Big(1+I\Big)\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2 \qquad (4.43)\end{aligned}$$

where (a) follows from the the generalized triangle inequality.

Next we bound the second term of the above inequality (denoted as T_1):

$$\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-(1-c_\nu\alpha_{t-1}^2)\Big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big)\Big\|^2$$

$$\leq 2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t,\mathcal{B}_x}-\big(\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\big)\Big\|^2+2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big\|^2$$
$$\leq 2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\mu_{t-1,\mathcal{B}_x}^{(m)}\Big\|^2+2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big\|^2$$

where the second inequality follows Proposition 111. We bound the two terms separately, for the first term, we have:

$$\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t,\mathcal{B}_x}^{(m)}-\mu_{t-1,\mathcal{B}_x}^{(m)}\Big\|^2\leq\hat{L}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_t^{(m)}-x_{t-1}^{(m)}\|^2+\|y_t^{(m)}-y_{t-1}^{(m)}\|^2\Big]$$
$$\leq\hat{L}^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|\eta\nu_{t-1}^{(m)}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big] \tag{4.44}$$

where the inequalities follow Proposition 93.b) and the fact that $\hat{x}_t^{(m)}=x_t^{(m)}$ when $t\neq\bar{t}_s$;

Next for the second term, we have:

$$\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\bar{\mu}_{t-1,\mathcal{B}_x}\Big\|^2=\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}-\big(\bar{\mu}_{t-1,\mathcal{B}_x}-\bar{\mu}_{t-1}\big)+\mu_{t-1}^{(m)}-\bar{\mu}_{t-1}\Big\|^2$$
$$\overset{(a)}{\leq}2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}-\big(\bar{\mu}_{t-1,\mathcal{B}_x}-\bar{\mu}_{t-1}\big)\Big\|^2+2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1}^{(m)}-\bar{\mu}_{t-1}\Big\|^2$$
$$\overset{(b)}{\leq}\underbrace{2\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}\Big\|^2}_{T_1}+\underbrace{4\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla h^{(m)}(\bar{x}_{t-1})-\nabla h(\bar{x}_{t-1})\Big\|^2}_{T_2}$$
$$+\underbrace{4\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1}^{(m)}-\nabla h^{(m)}(\bar{x}_{t-1})+\nabla h(\bar{x}_{t-1})-\bar{\mu}_{t-1}\Big\|^2}_{T_3} \tag{4.45}$$

Note for the term T_1 of Eq. 4.45, we have $\mathbb{E}\Big\|\mu_{t-1,\mathcal{B}_x}^{(m)}-\mu_{t-1}^{(m)}\Big\|^2\leq G_1^2+\frac{G_2^2}{b_x}$; For the term T_2

of Eq. 4.45, by the bounded intra-node heterogeneity assumption we have:

$$T_2 \leq 4\sum_{m=1}^{M}\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\|\nabla h^{(m)}(\bar{x}_{t-1}) - \nabla h^{(j)}(\bar{x}_{t-1})\|^2 \leq 4M\zeta^2$$

Finally, For the term T_3 of Eq. 4.45

$$\begin{aligned}
T_3 &\leq 8\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1}^{(m)} - \nabla h^{(m)}(\bar{x}_{t-1})\Big\|^2 + 8\sum_{m=1}^{M}\mathbb{E}\Big\|\nabla h(\bar{x}_{t-1}) - \bar{\mu}_{t-1}\Big\|^2 \\
&\leq 16\sum_{m=1}^{M}\mathbb{E}\Big\|\mu_{t-1}^{(m)} - \nabla h^{(m)}(\bar{x}_{t-1})\Big\|^2 \\
&\leq 32\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\mu_{t-1}^{(m)} - \nabla h^{(m)}(x_{t-1}^{(m)})\Big\|^2 + \Big\|\nabla h^{(m)}(x_{t-1}^{(m)}) - \nabla h^{(m)}(\bar{x}_{t-1})\Big\|^2\Big] \\
&\overset{(a)}{\leq} 32\bar{L}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2\Big] + 32\hat{L}^2\sum_{m=1}^{M}\mathbb{E}\Big[\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\|^2\Big]
\end{aligned}$$

where inequality (b) follows Proposition 93.c) and d).

Combine Eq. 4.42, Eq. 4.44 and Eq. 4.45, use the fact that $I \geq 1$, we have:

$$\begin{aligned}
&\sum_{m=1}^{M}\mathbb{E}\|\nu_t^{(m)} - \bar{\nu}_t\|^2 \\
&\leq \Big(1 + \frac{1}{I}\Big)\sum_{m=1}^{M}\mathbb{E}\|\nu_{t-1}^{(m)} - \bar{\nu}_{t-1}\|^2 + 4I\hat{L}^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[\underbrace{\|\eta\nu_{t-1}^{(m)}\|^2}_{T_1} + \|\gamma\omega_{t-1}^{(m)}\|^2\Big] \\
&\quad + 8IM(c_\nu\alpha_{t-1}^2)^2G_1^2 + \frac{8IM(c_\nu\alpha_{t-1}^2)^2G_2^2}{b_x} + 16IM(c_\nu\alpha_{t-1}^2)^2\zeta^2 \\
&\quad + 128I\bar{L}^2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_{t-1}^{(m)} - \bar{x}_{t-1}\|^2\Big] + 128I\hat{L}^2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\|^2\Big]
\end{aligned}$$

We separate the term T_1 with triangle inequality to get:

$$\sum_{m=1}^{M}\mathbb{E}\|\nu_t^{(m)} - \bar{\nu}_t\|^2$$

$$\leq \Big(1+\frac{1}{I}+8I\hat{L}^2\eta^2\alpha_{t-1}^2\Big)\sum_{m=1}^{M}\mathbb{E}\|\nu_{t-1}^{(m)}-\bar{\nu}_{t-1}\|^2+4I\hat{L}^2\alpha_{t-1}^2\sum_{m=1}^{M}\mathbb{E}\Big[2\|\eta\bar{\nu}_{t-1}\|^2+\|\gamma\omega_{t-1}^{(m)}\|^2\Big]$$
$$+8IM(c_\nu\alpha_{t-1}^2)^2G_1^2+\frac{8IM(c_\nu\alpha_{t-1}^2)^2G_2^2}{b_x}+16IM(c_\nu\alpha_{t-1}^2)^2\zeta^2$$
$$+\underbrace{128I\bar{L}^2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|x_{t-1}^{(m)}-\bar{x}_{t-1}\|^2\Big]}_{T_1}+128I\hat{L}^2(c_\nu\alpha_{t-1}^2)^2\sum_{m=1}^{M}\mathbb{E}\Big[\|y_{t-1}^{(m)}-y_{x_{t-1}^{(m)}}^{(m)}\|^2\Big]$$

Finally, choose $\eta\alpha_t<\frac{1}{16\hat{L}I}$ and combine with Lemma 98 to bound the term T_1, we get the bound in the lemma. This completes the proof. □

Next, to simply the notation, we denote $A_t=\mathbb{E}\|\bar{\nu}_t-\mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}]\|^2$, $B_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|y_t^{(m)}-y_{x_t^{(m)}}^{(m)}\|^2$, $C_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\omega_t^{(m)}-\nabla_y g^{(m)}(x_t^{(m)},y_t^{(m)})\|^2$, $D_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\nu_t^{(m)}-\bar{\nu}_t\|^2$, $E_t=\mathbb{E}\|\bar{\nu}_t\|^2$, $F_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\omega_t^{(m)}\|^2$.

Lemma 100. *For $\alpha_t<\frac{1}{16\hat{L}I}$, we have:*

$$\Big(1-\frac{3\kappa^2\eta^2c_\nu^2}{4*16^3I^5\hat{L}^4}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_tD_t\leq\frac{3c_\nu^2}{2*16^2I^4\hat{L}^2}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell B_\ell+\frac{3\eta^2}{32I}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell E_\ell+\frac{3\gamma^2}{64I}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell F_\ell$$
$$+\Big(\frac{3c_\nu^2G_1^2}{32I\hat{L}^2}+\frac{3c_\nu^2G_2^2}{32Ib_x\hat{L}^2}+\frac{3c_\nu^2\zeta^2}{16I\hat{L}^2}\Big)\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell^3$$

where the terms D_t, E_t and F_t are denoted above.

Proof. Based on Lemma 99, for $t\neq\bar{t}_s$, we have:

$$D_t\leq\Big(1+\frac{33}{32I}\Big)D_{t-1}+128I\hat{L}^2c_\nu^2\alpha_{t-1}^4B_{t-1}+8I\hat{L}^2\alpha_{t-1}^2\eta^2E_{t-1}+4I\hat{L}^2\alpha_{t-1}^2\gamma^2F_{t-1}$$
$$+8Ic_\nu^2\alpha_{t-1}^4G_1^2+\frac{8Ic_\nu^2\alpha_{t-1}^4G_2^2}{b_x}+16Ic_\nu^2\alpha_{t-1}^4\zeta^2+128I^2\bar{L}^2\eta^2c_\nu^2\alpha_{t-1}^4\sum_{\ell=\bar{t}_{s-1}}^{t-2}\alpha_l^2D_l$$

while for $t=\bar{t}_s$, we have $D_{\bar{t}_s}=1/M\sum_{m=1}^{M}\mathbb{E}\|\nu_{\bar{t}_s}^{(m)}-\bar{\nu}_{\bar{t}_s}\|^2=0$. Apply the above equation

recursively from $\bar{t}_{s-1}+1$ to t. so we have:

$$
\begin{aligned}
D_t \leq & \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big(1+\frac{33}{32I}\Big)^{t-\ell}\Big(128I\hat{L}^2c_\nu^2\alpha_\ell^4B_\ell + 8I\hat{L}^2\eta^2\alpha_\ell^2E_\ell + 4I\hat{L}^2\gamma^2\alpha_\ell^2F_\ell + 8Ic_\nu^2G_1^2\alpha_\ell^4 \\
& + \frac{8Ic_\nu^2G_2^2\alpha_\ell^4}{b_x} + 16Ic_\nu^2\zeta^2\alpha_\ell^4 + 128I^2\bar{L}^2\eta^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell}\alpha_{\bar{\ell}}^2D_{\bar{\ell}}\Big) \\
\leq & \sum_{\ell=\bar{t}_{s-1}}^{t-1} \Big(384I\hat{L}^2c_\nu^2\alpha_\ell^4B_\ell + 24I\hat{L}^2\eta^2\alpha_\ell^2E_\ell + 12I\hat{L}^2\gamma^2\alpha_\ell^2F_\ell + 24Ic_\nu^2G_1^2\alpha_\ell^4 + \frac{24Ic_\nu^2G_2^2\alpha_\ell^4}{b_x} \\
& + 16Ic_\nu^2\zeta^2\alpha_\ell^4 + 384I^2\bar{L}^2\eta^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell}\alpha_{\bar{\ell}}^2D_{\bar{\ell}}\Big)
\end{aligned}
$$

The second inequality uses the fact that $t-l \leq I$ and the inequality $log(1+a/x) \leq a/x$ for $x > -a$, so we have $(1+a/x)^x \leq e^a$, Then we choose $a = 33/32$ and $x = I$. Finally, we use the fact that $e^{33/32} \leq 3$.

Next we multiply α_t over both sides and take sum from $\bar{t}_{s-1}+1$ to $\bar{t}_s$, we have:

$$
\begin{aligned}
\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s}\alpha_tD_t \leq & \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\Big(384I\hat{L}^2c_\nu^2\alpha_\ell^4B_\ell + 24I\hat{L}^2\eta^2\alpha_\ell^2E_\ell + 12I\hat{L}^2\gamma^2\alpha_\ell^2F_\ell \\
& + \left(24Ic_\nu^2G_1^2 + \frac{24Ic_\nu^2G_1^2}{b_x} + 48Ic_\nu^2\zeta^2\right)\alpha_\ell^4 + 384I^2\bar{L}^2\eta^2c_\nu^2\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell}\alpha_{\bar{\ell}}^2D_{\bar{\ell}}\Big) \\
\overset{(a)}{\leq} & \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(24I^{1/2}\hat{L}c_\nu^2\alpha_\ell^4B_\ell + \frac{3I^{1/2}\hat{L}\eta^2}{2}\alpha_\ell^2E_\ell + \frac{3I^{1/2}\hat{L}\gamma^2}{4}\alpha_\ell^2F_\ell \\
& + \left(\frac{3I^{1/2}c_\nu^2G_1^2}{2\hat{L}} + \frac{3I^{1/2}c_\nu^2G_2^2}{2b_x\hat{L}} + \frac{3I^{1/2}c_\nu^2\zeta^2}{\hat{L}}\right)\alpha_\ell^4 + \frac{24I^{3/2}\bar{L}^2\eta^2c_\nu^2}{\hat{L}}\alpha_\ell^4\sum_{\bar{\ell}=\bar{t}_{s-1}}^{\ell}\alpha_{\bar{\ell}}^2D_{\bar{\ell}}\Big) \\
\overset{(b)}{\leq} & \frac{3c_\nu^2}{2*16^2I^4\hat{L}^2}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell B_\ell + \frac{3\eta^2}{32I}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell E_\ell + \frac{3\gamma^2}{64I}\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell F_\ell \\
& + \left(\frac{3c_\nu^2G_1^2}{32I\hat{L}^2} + \frac{3c_\nu^2G_2^2}{32Ib_x\hat{L}^2} + \frac{3c_\nu^2\zeta^2}{16I\hat{L}^2}\right)\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_\ell^3 + \frac{3\kappa^2\eta^2c_\nu^2}{4*16^3I^5\hat{L}^4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_tD_t
\end{aligned}
$$

In inequalities (a) and (b), we use $\alpha_t < \frac{1}{16\hat{L}I^{3/2}}$. Note that $\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s}\alpha_tD_t = \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_tD_t$

as $D_{\bar{t}_s} = D_{\bar{t}_{s-1}} = 0$, so we have:

$$\Big(1 - \frac{3\kappa^2\eta^2 c_\nu^2}{4 * 16^3 I^5 \hat{L}^4}\Big) \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_t D_t \leq \frac{3c_\nu^2}{2 * 16^2 I^4 \hat{L}^2} \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_\ell B_\ell + \frac{3\eta^2}{32I} \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_\ell E_\ell + \frac{3\gamma^2}{64I} \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_\ell F_\ell + \left(\frac{3c_\nu^2 G_1^2}{32I\hat{L}^2} + \frac{3c_\nu^2 G_2^2}{32Ib_x\hat{L}^2} + \frac{3c_\nu^2\zeta^2}{16I\hat{L}^2}\right) \sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1} \alpha_\ell^3$$

This completes the proof. □

4.8.1.4 Descent Lemma

Lemma 101. *Suppose $\eta < \frac{1}{2\bar{L}}$, $\alpha_t < 1$, for all $t \in [\bar{t}_{s-1}, \bar{t}_s - 1]$ and $s \in [S]$, the iterates generated satisfy:*

$$\mathbb{E}[h(\bar{x}_{t+1})] \leq \mathbb{E}[h(\bar{x}_t)] - \frac{\eta\alpha_t}{4}\mathbb{E}[\|\bar{\nu}_t\|^2] - \frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2] + 2\eta\alpha_t\mathbb{E}[\|\mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}] - \bar{\nu}_t\|^2] + 4\eta\alpha_t G_1^2 + \frac{\bar{L}^2 I\eta^3\alpha_t}{M} \sum_{\ell=\bar{t}_{s-1}}^{t-1} \alpha_l^2 \sum_{m=1}^{M} \mathbb{E}\|\Big(\nu_\ell^{(m)} - \bar{\nu}_\ell\Big)\|^2 + \frac{4\hat{L}^2\eta\alpha_t}{M} \sum_{m=1}^{M} \mathbb{E}\|y_{x_t^{(m)}}^{(m)} - y_t^{(m)}\|^2$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. By the smoothness of $h(x)$ we have:

$$\begin{aligned}
\mathbb{E}[h(\bar{x}_{t+1})] &\leq \mathbb{E}\Big[h(\bar{x}_t) + \langle\nabla h(\bar{x}_t), \bar{x}_{t+1} - \bar{x}_t\rangle + \frac{\bar{L}}{2}\|\bar{x}_{t+1} - \bar{x}_t\|^2\Big] \\
&\overset{(a)}{=} \mathbb{E}\Big[h(\bar{x}_t) - \eta\alpha_t\langle\nabla h(\bar{x}_t), \bar{\nu}_t\rangle + \frac{\eta^2\alpha_t^2\bar{L}}{2}\|\bar{\nu}_t\|^2\Big] \\
&\overset{(b)}{=} \mathbb{E}\Big[h(\bar{x}_t) - \frac{\eta\alpha_t}{2}\Big\|\bar{\nu}_t\Big\|^2 - \frac{\eta\alpha_t}{2}\|\nabla h(\bar{x}_t)\|^2 + \frac{\eta\alpha_t}{2}\Big\|\nabla h(\bar{x}_t) - \bar{\nu}_t\Big\|^2 + \frac{\eta\alpha_t^2\bar{L}}{2}\Big\|\bar{\nu}_t\Big\|^2\Big] \\
&= \mathbb{E}\Big[h(\bar{x}_t) - \frac{\eta\alpha_t}{4}\Big\|\bar{\nu}_t\Big\|^2 - \frac{\eta\alpha_t}{2}\|\nabla h(\bar{x}_t)\|^2 + \frac{\eta\alpha_t}{2}\underbrace{\Big\|\nabla h(\bar{x}_t) - \bar{\nu}_t\Big\|^2}_{T_1}\Big]
\end{aligned}$$

where equality (a) follows from the iterate update given in Algorithm 7; (b) uses $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a-b\|^2]$ and $\eta\alpha_t < \frac{1}{2\bar{L}}$; For the term T_1, we have:

$$\mathbb{E}\Big[\Big\|\nabla h(\bar{x}_t) - \bar{\nu}_t\Big\|^2\Big] \leq 2\mathbb{E}\Big[\Big\|\nabla h(\bar{x}_t) - \frac{1}{M}\sum_{m=1}^{M}\nabla h(x_t^{(m)})\Big\|^2\Big] + 4\mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{m=1}^{M}\nabla h(x_t^{(m)}) - \mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}]\Big\|^2\Big] + 4\mathbb{E}\Big[\Big\|\mathbb{E}_\xi[\bar{\mu}_{t,\mathcal{B}_x}] - \bar{\nu}_t\Big\|^2\Big]$$

For the first term, we have:

$$\begin{aligned}
&2\mathbb{E}\Big[\Big\|\nabla h(\bar{x}_t) - \frac{1}{M}\sum_{m=1}^{M}\nabla h(x_t^{(m)})\Big\|^2\Big] \\
&\leq \frac{2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\nabla h(\bar{x}_t) - \nabla h(x_t^{(m)})\Big\|^2\Big] \leq \frac{2\bar{L}^2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\bar{x}_t - x_t^{(m)}\Big\|^2\Big] \\
&\leq \frac{2\bar{L}^2 I\eta^2}{M}\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2\sum_{m=1}^{M}\mathbb{E}\Big\|\Big(\nu_\ell^{(m)} - \bar{\nu}_\ell\Big)\Big\|^2
\end{aligned}$$

where the last inequality uses Lemma 98. For the second term, we have:

$$\begin{aligned}
&4\mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{m=1}^{M}\nabla h(x_t^{(m)}) - \mathbb{E}_\xi[\bar{\mu}_{t,\xi}]\Big\|^2\Big] \\
&\leq \frac{8}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\nabla h(x_t^{(m)}) - \mu_t^{(m)}\Big\|^2\Big] + \frac{8}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|\mu_t^{(m)} - \mathbb{E}_\xi[\mu_{t,\mathcal{B}_x}^{(m)}]\Big\|^2\Big] \\
&\leq \frac{8\hat{L}^2}{M}\sum_{m=1}^{M}\mathbb{E}\Big[\Big\|y_{x_t^{(m)}}^{(m)} - y_t^{(m)}\Big\|^2\Big] + 8G_1^2
\end{aligned}$$

Plug the bound for term T_1 back gets the claim in the lemma. □

4.8.1.5 Proof of Convergence Theorem

We first denote the following potential function $\mathcal{G}(t)$:

$$\mathcal{G}_t = h(\bar{x}_t) + \frac{9bM\eta}{16\alpha_t}\Big\|\bar{\nu}_t - \frac{1}{M}\sum_{m=1}^{M}\nabla h(x_t^{(m)})\Big\|^2 + \frac{18\eta\hat{L}^2}{\mu\gamma}\times\frac{1}{M}\sum_{m=1}^{M}\Big\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\Big\|^2 + \frac{9bM\hat{L}^2\eta}{16L^2\alpha_t}\times\frac{1}{M}\sum_{m=1}^{M}\Big\|\omega_t^{(m)} - \nabla_y g^{(m)}(x_t^{(m)}, y_t^{(m)})\Big\|^2$$

Theorem 102. *Suppose* $\gamma \le \frac{1}{2L}$, $\eta < \min\Big(\frac{\mu\gamma}{144\kappa\hat{L}}, \frac{\hat{L}^2}{C_1^{1/2}c_\nu}, \frac{1}{(C_1I)^{1/2}}, \frac{\hat{L}}{(C_1I)^{1/2}}, \frac{I\hat{L}^2}{\kappa c_\nu}, 1\Big)$, $c_\nu = \frac{32}{9bM} + \frac{\hat{L}}{24Ib^2M^2}$, $c_\omega = \frac{144L^2}{bM\mu^2} + \frac{\hat{L}}{24Ib^2M^2}$, $u = (bM\sigma)^2\bar{u}$, *where* $\bar{u} = \max\Big(2, 16^3I^{9/2}\hat{L}, c_\nu^{3/2}, c_\omega^{3/2}\Big)$, $\delta = \frac{(bM\sigma)^{2/3}}{\hat{L}^{2/3}}$, *then we have:*

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] = O\Big(\frac{\kappa^{19/3}I^{3/2}}{T} + \frac{\kappa^{16/3}}{(bMT)^{2/3}} + \kappa^3b^2M^2I^{9/2}G_1^2\Big)$$

To reach an ϵ*-stationary point, we need* $T = O(\kappa^8(bM)^{-1}\epsilon^{-1.5})$, $I = O(\kappa^{10/9}(bM)^{-2/3}\epsilon^{-1/3})$ *and* $Q = O(\kappa\log\Big(\frac{\kappa}{bM\epsilon}\Big))$.

Proof. By the condition that $u \ge c_\nu^{3/2}\delta^3$, it is straightforward to verify that $c_\nu\alpha_t^2 < 1$. By Lemma 95 (in new notation), when $t \ne \bar{t}_s$, we have:

$$\frac{A_t}{\alpha_{t-1}} - \frac{A_{t-1}}{\alpha_{t-2}} \le \Big(\alpha_{t-1}^{-1} - \alpha_{t-2}^{-1} - c_\nu\alpha_{t-1}\Big)A_{t-1} + \frac{2c_\nu^2\alpha_{t-1}^3G_2^2}{bM} + \frac{4\hat{L}^2\eta^2\alpha_{t-1}}{bM}(D_{t-1} + E_{t-1}) + \frac{2\hat{L}^2\gamma^2\alpha_{t-1}F_{t-1}}{bM}$$

Note we choose $b_x = b$. For $\alpha_{t-1}^{-1} - \alpha_{t-2}^{-1}$, we have:

$$\alpha_t^{-1} - \alpha_{t-1}^{-1} = \frac{(u+\sigma^2 t)^{1/3}}{\delta} - \frac{(u+\sigma^2(t-1))^{1/3}}{\delta} \overset{(a)}{\leq} \frac{\sigma^2}{3\delta(u+\sigma^2(t-1))^{2/3}}$$
$$\overset{(b)}{\leq} \frac{2^{2/3}\sigma^2\delta^2}{3\delta^3(u+\sigma^2 t)^{2/3}} \overset{(c)}{=} \frac{2^{2/3}\sigma^2}{3\delta^3}\alpha_t^2 \leq \frac{2\hat{L}^2}{3M^2}\alpha_t^2 \leq \frac{\hat{L}}{24Ib^2M^2}\alpha_t$$

where inequality (a) results from the concavity of $x^{1/3}$ as: $(x+y)^{1/3} - x^{1/3} \leq y/3x^{2/3}$, inequality (b) used the fact that $u_t \geq 2\sigma^2$, inequality (c) uses the definition of α_t. By choosing $c_\nu = \frac{32}{9bM} + \frac{\hat{L}}{24Ib^2M^2}$, we have:

$$\frac{A_t}{\alpha_{t-1}} - \frac{A_{t-1}}{\alpha_{t-2}} \leq -\frac{32}{9bM}\alpha_{t-1}A_{t-1} + \frac{2c_\nu^2\alpha_{t-1}^3 G_2^2}{bM} + \frac{4\hat{L}^2\eta^2\alpha_{t-1}}{bM}(D_{t-1}+E_{t-1}) + \frac{2\hat{L}^2\gamma^2\alpha_{t-1}F_{t-1}}{bM}$$

When $t = \bar{t}_s$, by Lemma 95 and Lemma 98, we have:

$$\frac{A_t}{\alpha_{t-1}} - \frac{A_{t-1}}{\alpha_{t-2}} \leq -\frac{32}{9bM}\alpha_{t-1}A_{t-1} + \frac{2c_\nu^2\alpha_{t-1}^3 G_2^2}{bM} + \frac{8\hat{L}^2\eta^2\alpha_{t-1}}{bM}(D_{t-1}+E_{t-1})$$
$$+ \frac{2\hat{L}^2\gamma^2\alpha_{t-1}}{bM}F_{t-1} + \frac{8\hat{L}^2}{bM}\sum_{\ell=\bar{t}_{s-1}}^{t-1} I\eta^2\alpha_\ell D_\ell$$

Note we use the fact $\alpha_t/\alpha_{\bar{t}_s-1} < 2$ in the last term, which is due to:

$$\frac{\alpha_t}{\alpha_{\bar{t}_s-1}} = \frac{(u_{\bar{t}_s-1}+\sigma^2(\bar{t}_s-1))^{1/3}}{(u_t+\sigma^2 t)^{1/3}} = \Big(1 + \frac{u_{\bar{t}_s-1} - u_t + \sigma^2(\bar{t}_s-1-t)}{u_t+\sigma^2 t}\Big)^{1/3}$$
$$\leq \Big(1 + \frac{(I-1)\sigma^2}{u_t+\sigma^2 t}\Big)^{1/3} \leq 1 + \frac{(I-1)}{3(t+I+1)} \leq 2$$

where we use the condition $u_t \geq (I+1)\sigma^2$. Next, we telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$:

$$\Big(\frac{A_{\bar{t}_s}}{\alpha_{\bar{t}_s-1}} - \frac{A_{\bar{t}_{s-1}}}{\alpha_{\bar{t}_{s-1}-1}}\Big) \leq -\frac{32}{9bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t A_t + \frac{2c_\nu^2 G_2^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3 + \frac{16I\hat{L}^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t + \frac{8\hat{L}^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t + \frac{2\hat{L}^2\gamma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t \tag{4.46}$$

Next, we follow similar derivation as $A_t/\alpha_{t-1} - A_{t-1}/\alpha_{t-2}$. By Lemma 96, For $t \neq \bar{t}_s$, we choose $c_\omega = \frac{144L^2}{bM\mu^2} + \frac{\hat{L}}{24Ib^2M^2}$, to obtain:

$$\frac{C_t}{\alpha_{t-1}} - \frac{C_{t-1}}{\alpha_{t-2}} \leq -\frac{144L^2\alpha_{t-1}}{bM\mu^2}C_{t-1} + \frac{2c_\omega^2\alpha_{t-1}^3\sigma^2}{bM} + \frac{4L^2\eta^2\alpha_{t-1}}{bM}(D_{t-1}+E_{t-1}) + \frac{2L^2\gamma^2}{bM}\alpha_{t-1}F_{t-1}$$

Note we choose $b_y = bM$. When $t = \bar{t}_s$, by Lemma 95 and Lemma 98, we have:

$$\frac{C_t}{\alpha_{t-1}} - \frac{C_{t-1}}{\alpha_{t-2}} \leq -\frac{144L^2\alpha_{t-1}}{bM\mu^2}C_{t-1} + \frac{2c_\omega^2\alpha_{t-1}^3\sigma^2}{bM} + \frac{8L^2\eta^2\alpha_{t-1}}{bM}(D_{t-1}+E_{t-1}) + \frac{2L^2\gamma^2\alpha_{t-1}}{bM}F_{t-1} + \frac{4L^2}{bM}\sum_{\ell=\bar{t}_{s-1}}^{t-1} I\eta^2\alpha_\ell D_\ell$$

Divide $\hat{c}_\omega$ for both sides and then telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$, we have:

$$\Big(\frac{C_{\bar{t}_s}}{\alpha_{\bar{t}_s-1}} - \frac{C_{\bar{t}_{s-1}}}{\alpha_{\bar{t}_{s-1}-1}}\Big) \leq -\frac{144L^2}{2bM\mu^2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t C_t + \frac{2c_\omega^2\sigma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3 + \frac{16IL^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t + \frac{8L^2\eta^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t + \frac{2L^2\gamma^2}{bM}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t. \tag{4.47}$$

Next from Lemma 97, for $t \neq \bar{t}_s$, we have:

$$B_t - B_{t-1} \leq -\frac{\mu\gamma\alpha_{t-1}B_{t-1}}{4} - \frac{\gamma^2\alpha_{t-1}F_{t-1}}{4} + \frac{9\gamma\alpha_{t-1}C_{t-1}}{2\mu} + \frac{9\kappa^2\eta^2\alpha_{t-1}D_{t-1}}{\mu\gamma} + \frac{9\kappa^2\eta^2\alpha_{t-1}E_{t-1}}{\mu\gamma}$$

When $t = \bar{t}_s$, we have:

$$B_t - B_{t-1} \leq -\frac{\mu\gamma\alpha_{t-1}B_{t-1}}{4} - \frac{\gamma^2\alpha_{t-1}F_{t-1}}{4} + \frac{9\gamma\alpha_{t-1}C_{t-1}}{2\mu} + \frac{18\kappa^2\eta^2\alpha_{t-1}D_{t-1}}{\mu\gamma} + \frac{18\kappa^2\eta^2\alpha_{t-1}E_{t-1}}{\mu\gamma} + \frac{9\kappa^2 I\eta^2\alpha_{t-1}}{\mu\gamma}\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_\ell D_\ell$$

For the coefficient of the last term, we use $\alpha_t/\alpha_{\bar{t}_s-1} < 2$. We telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$ and have:

$$B_{\bar{t}_s} - B_{\bar{t}_{s-1}} \leq -\frac{\mu\gamma}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t - \frac{\gamma^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t + \frac{9\gamma}{2\mu}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t C_t + \frac{36I\kappa^2\eta^2}{\mu\gamma}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t + \frac{18\kappa^2\eta^2}{\mu\gamma}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t \tag{4.48}$$

Next, by Lemma 101, we have:

$$\mathbb{E}[h(\bar{x}_{t+1})] \leq \mathbb{E}[h(\bar{x}_t)] - \frac{\eta\alpha_t}{4}E_t - \frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2] + \bar{L}^2 I\eta^3\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2 D_l + 2\eta\alpha_t A_t + 4\hat{L}^2\eta\alpha_t B_t + 4\eta\alpha_t G_1^2$$

We telescope from $\bar{t}_{s-1}$ to $\bar{t}_s$ to have:

$$\begin{aligned}\mathbb{E}[h(\bar{x}_{\bar{t}_s}) - h(\bar{x}_{\bar{t}_{s-1}})] &\leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{4}E_t - \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2] + 4\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_1^2 \\ &\quad + \bar{L}^2 I\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t\sum_{\ell=\bar{t}_{s-1}}^{t-1}\alpha_l^2 D_l + \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}2\eta\alpha_t A_t + \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}4\hat{L}^2\eta\alpha_t B_t \\ &\leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{4}E_t - \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2] + 4\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_1^2\end{aligned}$$

$$+\frac{\kappa^2\eta^3}{64}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t+\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}2\eta\alpha_t A_t+\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}4\hat{L}^2\eta\alpha_t B_t \tag{4.49}$$

In the last inequality, we use the fact that $\bar{t}_s-\bar{t}_{s-1}\le I$, $\alpha_t<\frac{1}{16\hat{L}I}$ and $\bar{L}/\hat{L}=\kappa+1\le 2\kappa$

Combine Eq. (4.46),Eq. (4.47), Eq. (4.48) and Eq. (4.49) and we have:

$$\begin{aligned}
&\mathbb{E}[\mathcal{G}_{\bar{t}_s}]-\mathbb{E}[\mathcal{G}_{\bar{t}_{s-1}}]\\
&\le-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}[\|\nabla h(\bar{x}_t)\|^2]+\Big(\frac{9\hat{L}^2\eta c_\omega^2\sigma^2}{8L^2}+\frac{9\eta c_\nu^2 G_2^2}{8}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3-\frac{\hat{L}^2\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t\\
&\quad-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\frac{9\eta\gamma^2\hat{L}^2}{2}-\frac{9\eta\gamma^2\hat{L}^2}{8}-\frac{9\eta\gamma\hat{L}^2}{8\mu}\Big)\alpha_t F_t\\
&\quad-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\Big(\frac{1}{4}-\frac{324\kappa^2\hat{L}^2\eta^2}{\mu^2\gamma^2}-\frac{9\hat{L}^2\eta^2}{2}-\frac{9\hat{L}^2\eta^2}{2}\Big)\eta\alpha_t E_t\\
&\quad+\Big(\frac{\kappa^2}{64}+\frac{648I\kappa^2\hat{L}^2}{\mu^2\gamma^2}+9I\hat{L}^2+9I\hat{L}^2\Big)\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t+4\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_1^2
\end{aligned}$$

By the condition that $\eta<\frac{\mu\gamma}{144\kappa\hat{L}}$. So we have:

$$\begin{aligned}
&\mathbb{E}[\mathcal{G}_{\bar{t}_s}]-\mathbb{E}[\mathcal{G}_{\bar{t}_{s-1}}]\\
&\le-\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big]+\Big(\frac{9\eta c_\omega^2\sigma^2}{8\mu\gamma}+\frac{9\eta c_\nu^2 G_2^2}{8\mu\gamma}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3+4\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_1^2\\
&\quad-\frac{9\eta\gamma^2\hat{L}^2}{4}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t-\frac{3\eta}{16}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t-\frac{\hat{L}^2\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t+C_1 I\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t
\end{aligned}\tag{4.50}$$

where we denote $C_1=\Big(\frac{\kappa^2}{64}+\frac{648\kappa^2\hat{L}^2}{\mu^2\gamma^2}+9\hat{L}^2+9\hat{L}^2\Big)$. By Lemma 100 and choose $\eta<\frac{\hat{L}^2}{\kappa c_\nu}$, we have:

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t D_t\le\frac{c_\nu^2}{128I^4\hat{L}^2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t B_t+\frac{\eta^2}{8I}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t E_t+\frac{\gamma^2}{16I}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t F_t$$

$$+\Big(\frac{c_\nu^2 G_1^2}{8I\hat{L}^2}+\frac{c_\nu^2 G_2^2}{8Ib\hat{L}^2}+\frac{c_\nu^2\zeta^2}{4I\hat{L}^2}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3 \tag{4.51}$$

Combine Eq. (4.50) and Eq. (4.51), and use the condition that $\eta < \min\Big(\frac{\hat{L}^2}{C_1^{1/2}c_\nu}, \frac{1}{C_1^{1/2}}, \frac{\hat{L}}{C_1^{1/2}}, 1\Big)$, the fact that $I \geq 1$, we have:

$$\begin{aligned}\mathbb{E}[\mathcal{G}_{\bar{t}_s}]-\mathbb{E}[\mathcal{G}_{\bar{t}_{s-1}}] \leq & -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big]+4\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t G_1^2 \\ & +\eta\Big(\frac{9\hat{L}^2c_\omega^2\sigma^2}{8L^2}+\frac{9c_\nu^2G_2^2}{8}+\frac{c_\nu^2G_1^2}{8}+\frac{c_\nu^2G_2^2}{8b}+\frac{c_\nu^2\zeta^2}{4}\Big)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\alpha_t^3\end{aligned}$$

Sum over all $s \in [S]$ (assume $T = SI + 1$ without loss of generality), we have:

$$\mathbb{E}[\mathcal{G}_T]-\mathbb{E}[\mathcal{G}_1] \leq -\sum_{t=1}^{T-1}\frac{\eta\alpha_t}{2}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big]+\Big(\eta C_{\sigma,\zeta}+\frac{4\eta G_1^2}{\alpha_T^2}\Big)\sum_{t=1}^{T-1}\alpha_t^3$$

For ease of notation, we denote $C_{\sigma,\zeta} = \Big(\frac{9\hat{L}^2c_\omega^2\sigma^2}{8L^2}+\frac{9c_\nu^2G_2^2}{8}+\frac{c_\nu^2G_1^2}{8}+\frac{c_\nu^2G_2^2}{8b}+\frac{c_\nu^2\zeta^2}{4}\Big)$. Rearranging the terms and use the fact that α_t is non-increasing, we have:

$$\begin{aligned}&\frac{\eta\alpha_T}{2}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] \\ &\leq \mathbb{E}[\mathcal{G}_1]-\mathbb{E}[\mathcal{G}_T]+\Big(\eta C_{\sigma,\zeta}+\frac{4\eta G_1^2}{\alpha_T^2}\Big)\sum_{t=1}^{T-1}\alpha_t^3 \\ &\leq h(x_1)-h^*+\frac{9bM\eta A_1}{16\alpha_1}+\frac{18\eta\hat{L}^2B_1}{\mu\gamma}+\frac{9bM\eta C_1}{16\alpha_1}+\Big(\eta C_{\sigma,\zeta}+\frac{4\eta G_1^2}{\alpha_T^2}\Big)\sum_{t=1}^{T-1}\alpha_t^3\end{aligned}$$

where we use $\mathcal{G}_T \geq h^*$ (h^* is the optimal value of h), and for the last term, we use the

following fact:

$$\sum_{t=1}^{T}\alpha_t^3=\sum_{t=1}^{T}\frac{\delta^3}{u+\sigma^2 t}\leq\sum_{t=1}^{T}\frac{\delta^3}{\sigma^2+\sigma^2 t}=\frac{\delta^3}{\sigma^2}\sum_{t=1}^{T}\frac{1}{1+t}\leq\frac{\delta^3}{\sigma^2}\ln(T+1)=\frac{b^2M^2}{\hat{L}^2}\ln(T+1)$$

the first inequality follows $u_t>\sigma^2$, the last inequality follows Proposition 112. Next, we denote the initial sub-optimality as $\Delta=h(\bar{x}_1)-h^*$, and initial inner variable estimation error *i.e.* $B_1=\frac{1}{M}\sum_{m=1}^{M}\|y_1^{(m)}-y_{x_1^{(m)}}^{(m)}\|^2\leq\Delta_y$, and we assume $\Delta_y=O(\kappa^{-1})$, furthermore, we have:

$$A_1=\mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{m=1}^{M}\Big(\Phi^{(m)}(x_1^{(m)},y_1^{(m)};\mathcal{B}_x)-\Phi^{(m)}(x_1^{(m)},y_1^{(m)})\Big)\Big\|^2\Big]\leq\frac{\sigma^2}{b_1M}$$

and

$$C_1=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|\omega_1^{(m)}-\nabla_y g^{(m)}(x_1^{(m)},y_1^{(m)})\|^2\leq\frac{\sigma^2}{b_1}$$

where we choose the size of the first minibatch to be $b_x=b_1$ and $b_y=b_1M$. Then, we divide both sides by $\eta\alpha_T T/2$ to have:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big]\leq\frac{2\Delta}{\eta\alpha_T T}+\frac{9b\sigma^2}{8b_1T\alpha_1\alpha_T}+\frac{36\hat{L}^2\Delta_y}{\kappa\mu\gamma T\alpha_T}+\frac{9b\sigma^2}{8b_1T\alpha_1\alpha_T}\\+\frac{2b^2M^2C_{\sigma,\zeta}\ln(T)}{\hat{L}^2T\alpha_T}+\frac{8b^2M^2G_1^2\ln(T)}{\hat{L}^2T\alpha_T^3}\end{aligned}$$

Note that we have:

$$\frac{1}{\alpha_t t}=\frac{(u+\sigma^2 t)^{1/3}}{\delta t}\leq\frac{u^{1/3}}{\delta t}+\frac{\sigma^{2/3}}{\delta t^{2/3}}$$

where the inequality uses the fact that $(x+y)^{1/3} \leq x^{1/3} + y^{1/3}$. In particular, when $t = 1$, we have

$$\frac{1}{\alpha_1} \leq \frac{u^{1/3} + \sigma^{2/3}}{\delta} = \frac{\hat{L}^{2/3}(\bar{u}^{1/3}(bM)^{2/3} + 1)}{(bM)^{2/3}}$$

when $t = T$, we have:

$$\frac{1}{\alpha_T T} \leq \frac{u^{1/3}}{\delta T} + \frac{\sigma^{2/3}}{\delta T^{2/3}} = \frac{\hat{L}^{2/3}\bar{u}^{1/3}}{T} + \frac{\hat{L}^{2/3}}{(bMT)^{2/3}}$$

In summary, we have:

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] \leq \Big(\frac{2\Delta}{\eta} + \frac{9b\sigma^2}{8b_1\alpha_1} + \frac{36\hat{L}^2\Delta_y}{\kappa\mu\gamma} + \frac{9b\sigma^2}{8b_1\alpha_1} + 2\ln(T)\Big(\frac{b^2M^2C_{\sigma,\zeta}}{\hat{L}^2} + \frac{4b^2M^2G_1^2}{\hat{L}^2\alpha_T^2}\Big)\Big)\Big(\frac{\hat{L}^{2/3}\bar{u}^{1/3}}{T} + \frac{\hat{L}^{2/3}}{(bMT)^{2/3}}\Big)$$

Note that $\hat{L} = O(\kappa^2)$, $\bar{L} = O(\kappa^3)$, $c_\nu = O((bM)^{-1}\kappa^2)$ and $c_\omega = O((bM)^{-1}\kappa^2)$, $\bar{u} = O(I^{9/2}\kappa^2 + (bM)^{-3/2}\kappa^3)$, $\alpha_1^{-1} = O(I^{3/2}\kappa^2 + (bM)^{-1/2}\kappa^{7/3})$, then for η, we have:

$$\eta \leq \min\Big(\frac{\mu\gamma}{144\kappa\hat{L}}, \frac{\hat{L}^2}{\kappa c_\nu}, \frac{\hat{L}^2}{C_1^{1/2}c_\nu}, \frac{1}{C_1^{1/2}}, \frac{\hat{L}}{C_1^{1/2}}, \frac{1}{2\bar{L}}, 1\Big)$$

Recall that $C_1 = \Big(\frac{\kappa^2}{64} + \frac{648\kappa^2\hat{L}^2}{\mu^2\gamma^2} + 9\hat{L}^2 + 9\hat{L}^2\Big)$, suppose we choose $\gamma = \frac{1}{2L}$, then $C_1 = O(\kappa^8)$ and $\eta^{-1} = O(\kappa^4)$, $\mu\gamma = O(\kappa^{-1})$. recall that $C_{\sigma,\zeta} = \Big(\frac{9\hat{L}^2c_\omega^2\sigma^2}{8L^2} + \frac{9c_\nu^2G_2^2}{8} + \frac{c_\nu^2G_1^2}{8} + \frac{c_\nu^2G_2^2}{8b} + \frac{c_\nu^2\zeta^2}{4}\Big)$, so we have $C_{\sigma,\zeta} = O((bM)^{-2}\kappa^8)$, suppose we choose $b_1 = O(I^{3/2})$. Finally, for the coefficient

of the hyper-gradient bias term G_1^2, we have:

$$\frac{8b^2M^2\ln(T)}{\hat{L}^{1/3}\alpha_T^2}\Big(\frac{\bar{u}^{1/3}}{T}+\frac{1}{(bMT)^{2/3}}\Big) \leq \frac{16b^2M^2\bar{u}}{T}+16\left(\frac{b^2M^2\bar{u}}{T}\right)^{2/3}+16\left(\frac{b^2M^2\bar{u}}{T}\right)^{1/3}+16$$
$$= O(\kappa^3 b^2 M^2 I^{9/2})$$

Then, we have:

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\Big[\|\nabla h(\bar{x}_t)\|^2\Big] = O\Big(\frac{\kappa^{19/3}I^{3/2}}{T}+\frac{\kappa^{16/3}}{(bMT)^{2/3}}+\kappa^3 b^2 M^2 I^{9/2} G_1^2\Big)$$

To reach an ϵ-stationary point, we need $T = O(\kappa^8(bM)^{-1}\epsilon^{-1.5})$, $I = O(\kappa^{10/9}(bM)^{-2/3}\epsilon^{-1/3})$ and $Q = O(\kappa\log\Big(\frac{\kappa}{bM\epsilon}\Big))$. The communication cost is $E = T/I \geq \kappa^{62/9}(bM)^{-1/3}\epsilon^{-7/6}$, the sample complexity is $Gc(f,\epsilon) = O(M^{-1}\kappa^8\epsilon^{-1.5})$, $Gc(g,\epsilon) = O(\kappa^8\epsilon^{-1.5})$, $Jv(g,\epsilon) = O(\kappa^8\epsilon^{-1.5})$, $Hv(g,\epsilon) = O(\kappa^9\epsilon^{-1.5})$

Suppose we choose $b = O(\epsilon^{-0.5})$, we have $T = O(\kappa^8 M^{-1}\epsilon^{-1}))$, $I = \kappa^{10/9}M^{-2/3}$, $Q = O(\kappa\log\Big(\frac{\kappa}{M\epsilon}\Big))$ and $E = \kappa^{62/9}M^{-1/3}\epsilon^{-1}$. If we instead choose $b = O(1)$, we have $T = O(\kappa^8 M^{-1}\epsilon^{-1.5})$, $I = O(\kappa^{10/9}M^{-2/3}\epsilon^{-1/3})$ and $Q = O(\kappa\log\Big(\frac{\kappa}{M\epsilon}\Big))$. The communication cost is $E = O(\kappa^{62/9}M^{-1/3}\epsilon^{-7/6})$. □

4.8.2 Proof for the FedBiO-Local Algorithm

In this section, we investigate the convergence rate for the FedBiO-Local algorithm (Algorithm 9).

4.8.2.1 Lower Problem Solution Error

Lemma 103. *When $\gamma < \frac{1}{L}$, when $t \neq \bar{t}_s$, we have:*

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2 \leq (1-\frac{\mu\gamma}{2})\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + \frac{5\kappa^2\eta^2}{\mu\gamma M}\sum_{m=1}^{M}\mathbb{E}\left\|\nu_{t-1}^{(m)}\right\|^2 + \frac{3\gamma^2\sigma^2}{b_y}$$

when $t = \bar{t}_s$, we have:

$$\begin{aligned}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2 \leq (1-\frac{\mu\gamma}{2})\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + \frac{10\kappa^2\eta^2}{\mu\gamma M}\sum_{m=1}^{M}\mathbb{E}\left\|\nu_{t-1}^{(m)}\right\|^2 \\ + \frac{10\kappa^2}{\mu\gamma M}\sum_{m=1}^{M}\mathbb{E}\left\|x_t^{(m)} - \bar{x}_t\right\|^2 + \frac{3\gamma^2\sigma^2}{b_y}\end{aligned}$$

Proof. First, we have:

$$\begin{aligned}\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2 &\leq (1+\frac{\mu\gamma}{2})\mathbb{E}\left\|y_t^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + (1+\frac{2}{\mu\gamma})\mathbb{E}\left\|y_{x_t^{(m)}}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 \\ &\leq (1+\frac{\mu\gamma}{2})(1-\mu\gamma)\mathbb{E}\left\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + (1+\frac{2}{\mu\gamma})\mathbb{E}\left\|y_{x_t^{(m)}}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 \\ &\quad + (1+\frac{\mu\gamma}{2})\frac{2\gamma^2\sigma^2}{b_y} \\ &\leq (1-\frac{\mu\gamma}{2})\mathbb{E}\left\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + \frac{3\kappa^2}{\mu\gamma}\mathbb{E}\left\|x_t^{(m)} - x_{t-1}^{(m)}\right\|^2 + \frac{3\gamma^2\sigma^2}{b_y}\end{aligned}$$

where the second inequality is due to Proposition 113 where we choose $\gamma < 1/L$; in the last inequality, we use $\gamma < 1/(L)$ and $\mu \leq L$, For the last term, when $t \neq \bar{t}_s$, we have:

$$\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2 \leq (1-\frac{\mu\gamma}{2})\mathbb{E}\left\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + \frac{5\kappa^2\eta^2}{\mu\gamma}\mathbb{E}\left\|v_{t-1}^{(m)}\right\|^2 + \frac{3\gamma^2\sigma^2}{b_y}$$

Then when $t = \bar{t}_s$, we have

$$\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2 \leq (1 - \frac{\mu\gamma}{2})\mathbb{E}\left\|y_{t-1}^{(m)} - y_{x_{t-1}^{(m)}}^{(m)}\right\|^2 + \frac{10\kappa^2\eta^2}{\mu\gamma}\mathbb{E}\left\|v_{t-1}^{(m)}\right\|^2 + \frac{10\kappa^2}{\mu\gamma}\mathbb{E}\|x_t^{(m)} - \bar{x}_t\|^2 + \frac{3\gamma^2\sigma^2}{b_y}$$

Average over all clients, we get the claim in the lemma. □

4.8.2.2 Upper Variable Drift

Lemma 104. *For any $t \neq \bar{t}_s, s \in [S]$, we have:*

$$\|x_t^{(m)} - \bar{x}_t\|^2 \leq I\eta^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \left\|\nu_\ell^{(m)} - \bar{\nu}_\ell\right\|^2$$

Proof. Note from Algorithm and the definition of $\bar{t}_s$ that at $t = \bar{t}_s$ with $s \in [S]$, $x_t^{(m)} = \bar{x}_t$, for all k. For $t \neq \bar{t}_s$, with $s \in [S]$, we have: $x_t^{(m)} = x_{t-1}^{(m)} - \eta\nu_{t-1}^{(m)}$, this implies that: $x_t^{(m)} = x_{\bar{t}_{s-1}}^{(m)} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\nu_\ell^{(m)}$ and $\bar{x}_t = \bar{x}_{\bar{t}_{s-1}} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\bar{\nu}_\ell$. So for $t \neq \bar{t}_s$, with $s \in [S]$ we have:

$$\|x_t^{(m)} - \bar{x}_t\|^2 = \left\|x_{\bar{t}_{s-1}}^{(m)} - \bar{x}_{\bar{t}_{s-1}} - \Big(\sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\nu_\ell^{(m)} - \sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\bar{\nu}_\ell\Big)\right\|^2 \overset{(a)}{=} \left\|\sum_{\ell=\bar{t}_{s-1}}^{t-1} \eta\Big(\nu_\ell^{(m)} - \bar{\nu}_\ell\Big)\right\|^2$$
$$\leq I\eta^2 \sum_{\ell=\bar{t}_{s-1}}^{t-1} \left\|\nu_\ell^{(m)} - \bar{\nu}_\ell\right\|^2$$

This completes the proof. □

Lemma 105. *For $t \neq \bar{t}_s, s \in [S]$, we have:*

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|\left(\nu_t^{(m)}-\bar{\nu}_t\right)\right\|^2 \leq \frac{4\hat{L}^2}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_t^{(m)}-y_{x_t^{(m)}}^{(m)}\right\|^2+\frac{12\bar{L}^2I\eta^2}{M}\sum_{m=1}^{M}\sum_{\ell=\bar{t}_{s-1}}^{t-1}\mathbb{E}\left\|\nu_\ell^{(m)}-\bar{\nu}_\ell\right\|^2 + 8\zeta^2+4G_1^2+\frac{4G_2^2}{b_x}$$

for $t = \bar{t}_s, s \in [S]$, we have:

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|\left(\nu_t^{(m)}-\bar{\nu}_t\right)\right\|^2 \leq \frac{4\hat{L}^2}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_t^{(m)}-y_{x_t^{(m)}}^{(m)}\right\|^2+8\zeta^2+4G_1^2+\frac{4G_2^2}{b_x}$$

Proof. For $t \in [T]$, we have:

$$\begin{aligned}
&\mathbb{E}\left\|\left(\nu_t^{(m)}-\bar{\nu}_t\right)\right\|^2\\
&\overset{(a)}{\leq} 2\mathbb{E}\left\|\left(\nu_t^{(m)}-\nabla h^{(m)}(x_t^{(m)})\right)-\left(\bar{\nu}_t-\frac{1}{M}\sum_{m=1}^{M}\nabla h^{(j)}(x_t^{(j)})\right)\right\|^2\\
&\qquad\qquad + 2\mathbb{E}\left\|\left(\nabla h^{(m)}(x_t^{(m)})-\frac{1}{M}\sum_{j=1}^{M}\nabla h^{(j)}(x_t^{(j)})\right)\right\|^2\\
&\overset{(b)}{\leq} 2\mathbb{E}\left\|\nu_t^{(m)}-\nabla h^{(m)}(x_t^{(m)})\right\|^2+2\mathbb{E}\left\|\nabla h^{(m)}(x_t^{(m)})-\frac{1}{M}\sum_{j=1}^{M}\nabla h^{(j)}(x_t^{(j)})\right\|^2\\
&\leq 4\hat{L}^2\mathbb{E}\left\|y_t^{(m)}-y_{x_t^{(m)}}^{(m)}\right\|^2+4G_1^2+\frac{4G_2^2}{b_x}+2\mathbb{E}\left\|\nabla h^{(m)}(x_t^{(m)})-\frac{1}{M}\sum_{j=1}^{M}\nabla h^{(j)}(x_t^{(j)})\right\|^2 \qquad (4.52)
\end{aligned}$$

where the equality (a) uses triangle inequality and (b) follows from the application of Proposition 111. Next, for the second term of 4.52 we have:

$$\begin{aligned}
&\sum_{m=1}^{M}\left\|\nabla h^{(m)}(x_t^{(m)})-\frac{1}{M}\sum_{j=1}^{M}\nabla h^{(j)}(x_t^{(j)})\right\|^2\\
&\overset{(a)}{\leq} 2\sum_{m=1}^{M}\left\|\nabla h^{(m)}(x_t^{(m)})-\nabla h^{(m)}(\bar{x}_t)\right\|^2+4M\left\|\nabla h(\bar{x}_t)-\frac{1}{M}\sum_{j=1}^{M}\nabla h^{(j)}(x_t^{(j)})\right\|^2
\end{aligned}$$

$$+4\sum_{m=1}^{M}\left\|\nabla h^{(m)}(\bar{x}_t)-\nabla h(\bar{x}_t)\right\|^2 \overset{(b)}{\leq} 6\bar{L}^2\sum_{m=1}^{M}\left\|x_t^{(m)}-\bar{x}_t\right\|^2+4M\zeta^2 \tag{4.53}$$

where (a) follows the generalized triangle inequality; (b) utilizes the heterogeneity Assumption 59. Next for the first term, it is 0 when $t=\bar{t}_s$ and when $t\neq\bar{t}_s$, we use Lemma 104. Substituting 4.53 back to 4.52, we get the results in the lemma. □

Lemma 106. *For $s\in[S]$, we have:*

$$(1-12\bar{L}^2I^2\eta^2)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t\leq 4\hat{L}^2\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t+8I\zeta^2+4IG_1^2+\frac{4IG_2^2}{b_x}$$

Proof. For ease of notation, we denote $D_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|(\nu_t^{(m)}-\bar{\nu}_t)\|^2$ and $B_t=\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\|y_t^{(m)}-y_{x_t^{(m)}}^{(m)}\|^2$. Based on Lemma 105, we have:

$$D_t\leq 4\hat{L}^2B_t+12\bar{L}^2I\eta^2\sum_{\ell=\bar{t}_{s-1}}^{t-1}D_\ell+8\zeta^2+4G_1^2+\frac{4G_2^2}{b_x}$$

Next, we sum from $\bar{t}_{s-1}+1$ to $\bar{t}_s-1$, we have:

$$\begin{aligned}\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}D_t&\leq 4\hat{L}^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}B_t+12\bar{L}^2I\eta^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}\sum_{\ell=\bar{t}_{s-1}}^{t-1}D_\ell+8(I-1)\zeta^2+4(I-1)G^2\\&\leq 4\hat{L}^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}B_t+12\bar{L}^2I^2\eta^2\sum_{\ell=\bar{t}_{s-1}}^{\bar{t}_s-1}D_\ell+8(I-1)\zeta^2+4(I-1)G_1^2+\frac{4(I-1)G_2^2}{b_x}\end{aligned}$$

In the second inequality, we use $t-1\leq\bar{t}_s-1$, combine with the case when $t=\bar{t}_{s-1}$ in lemma 104, we have:

$$(1-12\bar{L}^2I^2\eta^2)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t\leq 4\hat{L}^2\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t+8I\zeta^2+4IG_1^2+\frac{4IG_2^2}{b_x}$$

This completes the proof. □

4.8.2.3 Descent Lemma

Lemma 107. *For all $t \in [\bar{t}_{s-1}, \bar{t}_s - 1]$, the iterates generated satisfy:*

$$\mathbb{E}\left\|\nabla h(\bar{x}_t) - \mathbb{E}_\xi[\bar{\nu}_t]\right\|^2 \leq \frac{2\hat{L}^2}{M}\sum_{m=1}^{M}\left(4\kappa^2\mathbb{E}\left\|x_t^{(m)} - \bar{x}_t\right\|^2 + 2\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2\right) + 2G_1^2$$

Proof. By definition of $\bar{\nu}_t$ and $\nabla h(\bar{x}_t)$, we have:

$$\begin{aligned}
\mathbb{E}\left\|\nabla h(\bar{x}_t) - \mathbb{E}_\xi[\bar{\nu}_t]\right\|^2 &\overset{(a)}{\leq} \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|\mathbb{E}_\xi[\nu_t^{(m)}] - \nabla h^{(m)}(\bar{x}_t)\right\|^2 \\
&\leq \frac{2}{M}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\mathbb{E}_\xi[\nu_t^{(m)}] - \mu_t^{(m)}\right\|^2 + \left\|\mu_t^{(m)} - \nabla h^{(m)}(\bar{x}_t)\right\|^2\right] \\
&\overset{(b)}{\leq} \frac{\hat{L}^2}{M}\sum_{m=1}^{M}\left(\mathbb{E}\left\|x_t^{(m)} - \bar{x}_t\right\|^2 + \mathbb{E}\left\|y_t^{(m)} - y_{\bar{x}_t}^{(m)}\right\|^2\right) + 2G_1^2 \\
&\leq \frac{2\hat{L}^2}{M}\sum_{m=1}^{M}\left(\mathbb{E}\left\|x_t^{(m)} - \bar{x}_t\right\|^2 + \mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)} + y_{x_t^{(m)}}^{(m)} - y_{\bar{x}_t}^{(m)}\right\|^2\right) + 2G_1^2 \\
&\leq \frac{2\hat{L}^2}{M}\sum_{m=1}^{M}\left(\left(1+2\kappa^2\right)\mathbb{E}\left\|x_t^{(m)} - \bar{x}_t\right\|^2 + 2\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2\right) + +2G_1^2
\end{aligned}$$

where inequality (a) follows the generalized triangle inequality; inequality (b) follows the Proposition 93 and Proposition 92.

□

Lemma 108. *For $t \neq \bar{t}_s$, the iterates generated satisfy:*

$$\begin{aligned}
\mathbb{E}[h(\bar{x}_{t+1})] \leq \mathbb{E}[h(\bar{x}_t)] &- \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\nu}_t]\right\|^2 + \frac{\eta^2\bar{L}G_2^2}{2b_xM} + \eta G_1^2 \\
&+ \frac{\eta\hat{L}^2}{M}\sum_{m=1}^{M}\left(4\kappa^2 I\eta^2\sum_{\ell=\bar{t}_{s-1}}^{t-1}\mathbb{E}\left\|\nu_\ell^{(m)} - \bar{\nu}_\ell\right\|^2 + 2\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2\right)
\end{aligned}$$

for $t = \bar{t}_s$, we have:

$$\mathbb{E}[h(\bar{x}_{t+1})] \leq \mathbb{E}[h(\bar{x}_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\nu}_t]\right\|^2 + \frac{\eta^2 \bar{L} G_2^2}{2b_x M} + \eta G_1^2 + \frac{2\eta \hat{L}^2}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Using the smoothness of f we have:

$$\begin{aligned}
\mathbb{E}[h(\bar{x}_{t+1})] &\leq \mathbb{E}[h(\bar{x}_t)] + \mathbb{E}\langle \nabla h(\bar{x}_t), \bar{x}_{t+1} - \bar{x}_t\rangle + \frac{\bar{L}}{2}\mathbb{E}\|\bar{x}_{t+1} - \bar{x}_t\|^2 \\
&\overset{(a)}{=} \mathbb{E}[h(\bar{x}_t)] - \eta\mathbb{E}\langle \nabla h(\bar{x}_t), \mathbb{E}_\xi[\bar{\nu}_t]\rangle + \frac{\eta^2\bar{L}}{2}\mathbb{E}\|\mathbb{E}_\xi[\bar{\nu}_t]\|^2 + \frac{\eta^2\bar{L}G_2^2}{2b_x M} \\
&\overset{(b)}{=} \mathbb{E}[h(\bar{x}_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t) - \mathbb{E}_\xi[\bar{\nu}_t]\|^2 \\
&\quad - \left(\frac{\eta}{2} - \frac{\eta^2\bar{L}}{2}\right)\mathbb{E}\|\mathbb{E}_\xi[\bar{\nu}_t]\|^2 + \frac{\eta^2\bar{L}G_2^2}{2b_x M} \\
&\overset{(c)}{\leq} \mathbb{E}[h(\bar{x}_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}\mathbb{E}\left\|\mathbb{E}_\xi[\nu_t^{(m)}]\right\|^2 + \frac{\eta^2\bar{L}G_2^2}{2b_x M} + \eta G_1^2 \\
&\quad + \frac{\eta\hat{L}^2}{M}\sum_{m=1}^{M}\Big(\underbrace{4\kappa^2\mathbb{E}\left\|x_t^{(m)} - \bar{x}_t\right\|^2}_{T_1} + 2\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2\Big)
\end{aligned}$$

where equality (a) follows from the iterate update given in Step 6 of Algorithm 8; (b) uses $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a-b\|^2]$; (c) follows the assumption that $\eta < 1/2\bar{L}$. Finally, use lemma 107 to bound T_1 when $t \neq \bar{t}_s$ finishes the proof. □

4.8.2.4 Proof of Convergence Theorem

We first denote the following potential function $\mathcal{G}(t)$:

$$\mathcal{G}_t = \mathbb{E}[h(\bar{x}_t)] + \frac{9\eta\hat{L}^2}{\mu\gamma} \times \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\right\|^2$$

Theorem 109. *Suppose we have constant* $\bar{\eta} = \min\left(\frac{1}{2C_1^{1/2}}, \frac{\mu\gamma}{12\kappa\hat{L}}, \frac{1}{2\bar{L}}, \frac{1}{6I\hat{L}}\right)$*, if we choose* $\eta = \min\left(\bar{\eta}, \left(\frac{2\Delta}{C_\eta T}\right)^{1/3}\right)$ *and* $\gamma = \frac{1}{2L}$*, we have:*

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla h(\bar{x}_t)\|^2 = O\left(\frac{\kappa^5}{T} + \left(\frac{\kappa^{16}}{T^2}\right)^{1/3} + \frac{\kappa^5\sigma^2}{b_y} + \frac{G_2^2}{b_x M} + G_1^2\right)$$

To reach an ϵ *stationary point, we choose the inner batch size* $b_y = O(\kappa^5\epsilon^{-1})$*, upper batch size* $b_x = O(M^{-1}\epsilon^{-1})$ *and* $Q = O(\kappa\log\left(\frac{\kappa}{\epsilon}\right))$ *in Eq. 4.11, and* $T = O(\kappa^8\epsilon^{-1.5})$ *number of iterations.*

Proof. Similar to Lemma 106, we denote $D_t = \frac{1}{M}\sum_{m=1}^{M}\|\left(\nu_t^{(m)} - \bar{\nu}_t\right)\|^2$, $B_t = \frac{1}{M}\sum_{m=1}^{M}\|y_t^{(m)} - y_{x_t^{(m)}}^{(m)}\|^2$, additionally, we denote $E_t = \|\mathbb{E}_\xi[\bar{\nu}_t]\|^2$. First, by Lemma 103, when $t \neq \bar{t}_s$, by the triangle inequality, we have:

$$B_t - B_{t-1} \leq -\frac{\mu\gamma}{2}B_{t-1} + \frac{10\kappa^2\eta^2}{\mu\gamma}D_{t-1} + \frac{10\kappa^2\eta^2}{\mu\gamma}E_{t-1} + \frac{10\kappa^2\eta^2G_2^2}{\mu\gamma b_x M} + \frac{3\gamma^2\sigma^2}{b_y}$$

When $t = \bar{t}_s$, we have:

$$B_t - B_{t-1} \leq -\frac{\mu\gamma}{2}B_{t-1} + \frac{20\kappa^2\eta^2}{\mu\gamma}D_{t-1} + \frac{20\kappa^2\eta^2}{\mu\gamma}E_{t-1} + \frac{10\kappa^2 I\eta^2}{\mu\gamma}\sum_{\ell=\bar{t}_{s-1}}^{t-1}D_\ell + \frac{10\kappa^2\eta^2G_2^2}{\mu\gamma b_x M} + \frac{3\gamma^2\sigma^2}{b_y}$$

We telescope from $\bar{t}_{s-1}+1$ to $\bar{t}_s$ and have:

$$B_{\bar{t}_s} - B_{\bar{t}_{s-1}} \leq -\frac{\mu\gamma}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} B_t + \frac{40\kappa^2 I\eta^2}{\mu\gamma}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} D_t + \frac{20\kappa^2\eta^2}{\mu\gamma}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} E_t + \frac{10I\kappa^2\eta^2 G_2^2}{\mu\gamma b_x M} + \frac{3I\gamma^2\sigma^2}{b_y} \tag{4.54}$$

Next, by Lemma 108, when $t \neq \bar{t}_s$, we have:

$$\begin{aligned}
&\mathbb{E}[h(\bar{x}_{t+1})] - \mathbb{E}[h(\bar{x}_t)] \\
&\leq -\frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}E_t + 4\kappa^2\hat{L}^2 I\eta^3\sum_{\ell=\bar{t}_{s-1}}^{t-1} D_l + 2\eta\hat{L}^2 B_t + \frac{\eta^2\bar{L}G_2^2}{2b_x M} + \eta G_1^2
\end{aligned}$$

and when $t = \bar{t}_s$, we have:

$$\mathbb{E}[h(\bar{x}_{t+1})] - \mathbb{E}[h(\bar{x}_t)] \leq -\frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{\eta}{4}E_t + 2\eta\hat{L}^2 B_t + \frac{\eta^2\bar{L}G_2^2}{2b_x M} + \eta G_1^2$$

We telescope from $\bar{t}_{s-1}$ to $\bar{t}_s$ to have:

$$\begin{aligned}
\mathbb{E}[h(\bar{x}_{\bar{t}_s})] - \mathbb{E}[h(\bar{x}_{\bar{t}_{s-1}})] &\leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta}{4}E_t + 4\kappa^2\hat{L}^2 I\eta^3\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s-1}\sum_{\ell=\bar{t}_{s-1}}^{t-1} D_l \\
&\quad + \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} 2\hat{L}^2\eta B_t + \frac{I\eta^2\bar{L}G_2^2}{2b_x M} + I\eta G_1^2 \\
&\leq -\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\frac{\eta}{4}E_t + 4\kappa^2\hat{L}^2 I^2\eta^3\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} D_l \\
&\quad + 2\hat{L}^2\eta\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1} B_t + \frac{I\eta^2\bar{L}G_2^2}{2b_x M} + I\eta G_1^2
\end{aligned} \tag{4.55}$$

In the last inequality, we use the fact that $\bar{t}_s - \bar{t}_{s-1} \leq I$.

Next, by the definition of the potential function and combine with Eq. 4.54 and

Eq. 4.55, we have:

$$\begin{aligned}\mathcal{G}_{\bar{t}_s} - \mathcal{G}_{\bar{t}_{s-1}} \leq & -\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{5\eta\hat{L}^2}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t - \frac{\eta}{4}\left(1 - \frac{720\kappa^2\eta^2\hat{L}^2}{\mu^2\gamma^2}\right)\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}E_t \\ & + \left(\frac{360}{\mu^2\gamma^2} + 4I\right)\eta^3\kappa^2\hat{L}^2 I\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t + \frac{27I\hat{L}^2\gamma\eta\sigma^2}{b_y\mu} \\ & + \frac{90I\kappa^2\hat{L}^2\eta^3G_2^2}{\mu^2\gamma^2 b_x M} + \frac{I\eta^2\bar{L}G_2^2}{2b_x M} + I\eta G_1^2\end{aligned}$$

to bound the coefficients above, we choose $\eta \leq \frac{\mu\gamma}{48\kappa\hat{L}}$. Then we have:

$$\begin{aligned}\mathcal{G}_{\bar{t}_s} - \mathcal{G}_{\bar{t}_{s-1}} \leq & -\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{5\eta\hat{L}^2}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t - \frac{\eta}{8}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}E_t \\ & + \left(\frac{360}{\mu^2\gamma^2} + 4I\right)\eta^3\kappa^2\hat{L}^2 I\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t + \frac{27I\hat{L}^2\gamma\eta\sigma^2}{b_y\mu} \\ & + \frac{90I\kappa^2\hat{L}^2\eta^3G_2^2}{\mu^2\gamma^2 b_x M} + \frac{I\eta^2\bar{L}G_2^2}{2b_x M} + I\eta G_1^2\end{aligned}$$

By lemma 106, and choosing $\eta < \frac{1}{6I\bar{L}}$, we have:

$$\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}D_t \leq 6\hat{L}^2\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t + 18I\zeta^2 + 6IG_1^2 + \frac{6IG_2^2}{b_x}$$

Next, we denote $C_1 = \left(\frac{360}{\mu^2\gamma^2} + 4I\right)\kappa^2\hat{L}^2$, and choose $\eta < \min(\frac{1}{2C_1^{1/2}}, \frac{\mu\gamma}{12\kappa\hat{L}}, \frac{1}{2\bar{L}})$ then we have:

$$\begin{aligned}&\mathcal{G}_{\bar{t}_s} - \mathcal{G}_{\bar{t}_{s-1}} \\ &\leq -\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 - \frac{5\eta\hat{L}^2}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}B_t - \frac{\eta}{8}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}E_t \\ &\quad + C_1 I\eta^3\left(6\hat{L}^2\sum_{t=\bar{t}_{s-1}+1}^{\bar{t}_s}B_t + 18I\zeta^2 + 6IG_1^2 + \frac{6IG_2^2}{b_x}\right)\end{aligned}$$

$$
\begin{aligned}
&+\frac{27I\hat{L}^2\gamma\eta\sigma^2}{b_y\mu}+\frac{5I\eta G_2^2}{8b_xM}+\frac{I\eta^2\bar{L}G_2^2}{2b_x}+I\eta G_1^2\\
\leq&-\frac{\eta}{2}\sum_{t=\bar{t}_{s-1}}^{\bar{t}_s-1}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2+18C_1I\hat{L}^2\eta^3\zeta^2+6C_1I\hat{L}^2\eta^3G_1^2+\frac{6C_1I\hat{L}^2\eta^3G_2^2}{b_x}+\frac{5I\eta G_2^2}{8b_xM}\\
&+\frac{27I\hat{L}^2\gamma\eta\sigma^2}{b_y\mu}+\frac{I\eta^2\bar{L}G_2^2}{2b_xM}+I\eta G_1^2
\end{aligned}
$$

Sum over all $s\in[S]$ (assume $T=SI+1$ without loss of generality) to obtain:

$$
\begin{aligned}
\frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2&\leq\mathcal{G}_1-\mathcal{G}_T+18C_1T\hat{L}^2\eta^3\zeta^2+6C_1T\hat{L}^2\eta^3G_1^2+\frac{6C_1T\hat{L}^2\eta^3G_2^2}{b_x}\\
&\quad+\frac{5T\eta G_2^2}{8b_xM}+\frac{27T\hat{L}^2\gamma\eta\sigma^2}{b_y\mu}+\frac{T\eta^2\bar{L}G_2^2}{2b_xM}+T\eta G_1^2\\
&\leq\Delta+\frac{9\eta\hat{L}^2\Delta_y}{\mu\gamma}+18C_1T\hat{L}^2\eta^3\zeta^2+6C_1T\hat{L}^2\eta^3G_1^2+\frac{6C_1T\hat{L}^2\eta^3G_2^2}{b_x}\\
&\quad+\frac{5T\eta G_2^2}{8b_xM}+\frac{27T\hat{L}^2\gamma\eta\sigma^2}{b_y\mu}+\frac{T\eta^2\bar{L}G_2^2}{2b_xM}+T\eta G_1^2
\end{aligned}
$$

we define $\Delta=h(x_1)-h^*$ as the initial sub-optimality of the function and $\Delta_y=\frac{1}{M}\sum_{m=1}^{M}\left\|y_1^{(m)}-y_{x_1}^{(m)}\right\|^2$ as the initial sub-optimality of the inner variable estimation, then we divide by $\eta T/2$ on both sides and have:

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2\leq&\underbrace{\frac{2\Delta}{\eta T}+\frac{\eta\bar{L}G_2^2}{2b_xM}+\left(36C_1\hat{L}^2\zeta^2+12C_1\hat{L}^2G_1^2+\frac{12C_1\hat{L}^2G_2^2}{b_x}\right)\eta^2}_{T_1}\\
&+\underbrace{\frac{18\hat{L}^2\Delta_y}{\mu\gamma T}+\frac{54\hat{L}^2\gamma\sigma^2}{b_y\mu}}_{T_2}+\underbrace{\frac{5G_2^2}{4b_xM}+G_1^2}_{T_3}
\end{aligned}
$$

As shown in the inequality, we break the bound into three parts. The T_1 part has a structure similar to that for the single level federated learning problems. Then the T_2 part includes the optimization error of the lower problem, and the statistical error of sampling. Finally,

the T_3 part includes the bias and variance of the hyper-gradient estimate.

Next, by $\eta < \frac{1}{2\bar{L}}$, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 \leq \frac{2\Delta}{\eta T} + \left(36C_1\hat{L}^2\zeta^2 + 12C_1\hat{L}^2G_1^2 + \frac{12C_1\hat{L}^2G_2^2}{b_x}\right)\eta^2 + \frac{18\hat{L}^2\Delta_y}{\mu\gamma T} + \frac{54\hat{L}^2\gamma\sigma^2}{b_y\mu} + \frac{G_2^2}{4b_xM} + \frac{5G_2^2}{4b_xM} + G_1^2 \tag{4.56}$$

Next, we denote constant $\bar{\eta} = \min\left(\frac{1}{2C_1^{1/2}}, \frac{\mu\gamma}{12\kappa\hat{L}}, \frac{1}{2\bar{L}}, \frac{1}{6I\hat{L}}\right)$ and

$$C_\eta = \left(36C_1\hat{L}^2\zeta^2 + 12C_1\hat{L}^2G_1^2 + \frac{12C_1\hat{L}^2G_2^2}{b_x}\right)$$

we choose

$$\eta = \min\left(\bar{\eta}, \left(\frac{2\Delta}{C_\eta T}\right)^{1/3}\right)$$

and $\gamma = \frac{1}{2L}$, and obtain:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla h(\bar{x}_t)\|^2 \leq \frac{2\Delta}{\bar{\eta}T} + \frac{36\kappa\hat{L}^2\Delta_y}{T} + \left(\frac{4C_\eta\Delta^2}{T^2}\right)^{1/3} + \frac{27\hat{L}^2\sigma^2}{\mu b_y L} + \frac{3G_2^2}{2b_xM} + 2G_1^2$$

Finally, since $\hat{L} = O(\kappa^2)$, $\bar{L} = O(\kappa^3)$ and and $\mu\gamma = O(\kappa^{-1})$. Suppose we choose $I = O(1)$, then $\bar{\eta} = O(\kappa^{-4})$ and $C_1 = O(\kappa^8)$, $\zeta = O(\kappa^2)$, $C_\eta = O(\kappa^{16})$, thus, we have

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla h(\bar{x}_t)\|^2 = O\left(\frac{\kappa^5}{T} + \left(\frac{\kappa^{16}}{T^2}\right)^{1/3} + \frac{\kappa^5\sigma^2}{b_y} + \frac{G_2^2}{b_xM} + G_1^2\right)$$

and to reach an ϵ stationary point, we choose the inner batch size $b_y = O(\kappa^5\epsilon^{-1})$, upper batch size $b_x = O(M^{-1}\epsilon^{-1})$ and $Q = O(\kappa\log\left(\frac{\kappa}{\epsilon}\right))$ in Eq. 4.11, and $T = O(\kappa^8\epsilon^{-1.5})$ number

of iterations. □

4.9 Useful Propositions

In this section, we state some propositions useful in the proof:

Proposition 110 (Lemma 3 of [55]). *(generalized triangle inequality) Let* $\{x_k\}, k \in K$ *be* K *vectors. Then the following are true:*

1. $||x_i + x_j||^2 \leq (1+a)||x_i||^2 + (1+\frac{1}{a})||x_j||^2$ *for any* $a > 0$, *and*
2. $||\sum_{k=1}^K x_k||^2 \leq K \sum_{k=1}^K ||x_k||^2$

Proposition 111 (Lemma C.1 of [13]). *For a finite sequence* $x^{(k)} \in \mathbb{R}^d$ *for* $k \in [K]$ *define* $\bar{x} \coloneqq \frac{1}{K}\sum_{k=1}^K x^{(k)}$, *we then have* $\sum_{k=1}^K \|x^{(k)} - \bar{x}\|^2 \leq \sum_{k=1}^K \|x^{(k)}\|^2$.

Proposition 112 (Lemma C.2 of [13]). *Let* $a_0 > 0$ *and* $a_1, a_2, \ldots, a_T \geq 0$. *We have*

$$\sum_{t=1}^{T} \frac{a_t}{a_0 + \sum_{i=t}^t a_i} \leq \ln\Big(1 + \frac{\sum_{i=1}^t a_i}{a_0}\Big).$$

Proposition 113. *Suppose we have function* $g(y)$, *which is* L*-smooth and* μ*-strongly-convex, then suppose* $\gamma < \frac{1}{L}$, *the progress made by one step of gradient descent is:*

$$\mathbb{E}\|y_{t+1} - y^*\|^2 \leq (1 - \mu\gamma)\|y^* - y_t\|^2 + 2\gamma^2\sigma^2$$

where y^* *is the minimum of* $g(y)$ *and we have update rule* $g(y_{t+1}) = g(y_t) - \gamma\nabla g(y_t, \xi)$, *where the error of stochastic gradient estimate is bounded by* σ^2.

Proof. First, by the strong convexity of of function $g(y)$, we have:

$$\begin{aligned} g(y^*) &\geq g(y_t) + \langle \nabla_y g(y_t), y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 \\ &= g(y_t) + \langle \nabla_y g(y_t), y^* - y_{t+1} \rangle + \langle \nabla_y g(y_t), y_{t+1} - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 \end{aligned}$$

Then by L-smoothness, we have: $\frac{L}{2}\|y_{t+1} - y_t\|^2 \geq g(y_{t+1}) - g(y_t) - \langle \nabla_y g(y_t), y_{t+1} - y_t \rangle$,

Combining above two inequalities and take expectation on both sides, we have

$$\begin{aligned} g(y^*) &\geq \mathbb{E}g(y_{t+1}) + \mathbb{E}\langle \nabla_y g(y_t), y^* - y_{t+1} \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L}{2}\mathbb{E}\|y_{t+1} - y_t\|^2 \\ &\geq \mathbb{E}g(y_{t+1}) + \gamma\|\nabla_y g(y_t)\|^2 + \langle \nabla_y g(y_t), y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L\gamma^2}{2}\mathbb{E}\|\nabla_y g(y_t, \xi)\|^2 \\ &\geq \mathbb{E}g(y_{t+1}) + \gamma\|\nabla_y g(y_t)\|^2 + \langle \nabla_y g(y_t), y^* - y_t \rangle \\ &\quad + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L\gamma^2}{2}\mathbb{E}\|\nabla_y g(y_t)\|^2 - \frac{L\gamma^2\sigma^2}{2} \\ &\geq \mathbb{E}g(y_{t+1}) + \langle \nabla_y g(y_t), y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 + \left(\gamma - \frac{L\gamma^2}{2}\right)\|\nabla_y g(y_t)\|^2 - \frac{L\gamma^2\sigma^2}{2} \end{aligned}$$

By definition of y^*, we have $g(y^*) \geq g(y_{t+1})$. Thus, we obtain

$$0 \geq \langle \nabla_y g(y_t), y^* - y_t \rangle + \frac{\mu}{2}\|y^* - y_t\|^2 + \left(\gamma - \frac{L\gamma^2}{2}\right)\|\nabla_y g(y_t)\|^2 - \frac{L\gamma^2\sigma^2}{2}$$

By $y_{t+1} = y_t - \gamma\nabla_y g(y_t, \xi)$, we have:

$$\begin{aligned} \mathbb{E}\|y_{t+1} - y^*\|^2 &= \mathbb{E}\|y_t - \gamma\nabla_y g(y_t, \xi) - y^*\|^2 \\ &= \|y_t - y^*\|^2 - 2\gamma\langle \nabla_y g(y_t), y_t - y^* \rangle + \gamma^2\mathbb{E}\|\nabla_y g(y_t, \xi)\|^2 \\ &\leq \big(1 - \mu\gamma\big)\|y_t - y^*\|^2 - 2\gamma\Big(\gamma - \frac{L\gamma^2}{2} - \frac{\gamma}{2}\Big)\|\nabla_y g(y_t)\|^2 + \big(L\gamma^3 + \gamma^2\big)\,\sigma^2 \end{aligned}$$

Then since we choose $\gamma < \frac{1}{L}$, we obtain:

$$\mathbb{E}\|y_{t+1} - y^*\|^2 \leq (1 - \mu\gamma)\|y^* - y_t\|^2 + 2\gamma^2\sigma^2$$

This completes the proof. □

Proposition 114. *Suppose we have function $g(y)$, which is L-smooth and μ-strongly-convex, then suppose $\gamma < \frac{1}{2L}$ and $\alpha_t < 1$, the progress made by one step of gradient descent is:*

$$\begin{aligned}\|y_{t+1} - y^*\|^2 \leq (1 - \frac{\mu\gamma\alpha_t}{2})\|y_t - y^*\|^2 - \frac{\gamma^2\alpha_t}{4}\|\omega_t\|^2 \\ + \frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(x_t, y_t) - \mathbb{E}[w_t]\|^2 + \frac{3\gamma^2\alpha_t}{2}Var[\omega_t].\end{aligned}$$

where y^* is the minimum of $g(y)$ and we have update rule $g(y_{t+1}) = g(y_t) - \gamma\alpha_t\omega_t$.

Proof. First, Suppose we denote $\tilde{y}_{t+1} = y_t - \gamma\omega_t$, then we have $y_{t+1} = y_t + \alpha_t(\tilde{y}_{t+1} - y_t)$. By the strong convexity of of function $g(y)$, we have:

$$\begin{aligned}g(y^*) &\geq g(y_t) + \langle\nabla_y g(y_t), y^* - y_t\rangle + \frac{\mu}{2}\|y^* - y_t\|^2 \\ &= g(y_t) + \mathbb{E}\langle\mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle + \mathbb{E}\langle\nabla_y g(y_t) - \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle \\ &\quad - \gamma\langle\nabla_y g(y_t), \mathbb{E}[w_t]\rangle + \frac{\mu}{2}\|y^* - y_t\|^2 \qquad (4.57)\end{aligned}$$

where the expectation is *w.r.t* the stochasity of ω_t. Then by L-smoothness, we have:

$$\frac{L}{2}\mathbb{E}\|\tilde{y}_{t+1} - y_t\|^2 \geq \mathbb{E}g(\tilde{y}_{t+1}) - g(y_t) + \gamma\langle\nabla_y g(y_t), \mathbb{E}[\omega_t]\rangle \qquad (4.58)$$

Combining the 4.57 with 4.58, we have

$$
\begin{aligned}
g(y^*) &\geq \mathbb{E}g(\tilde{y}_{t+1}) + \mathbb{E}\langle \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle + \mathbb{E}\langle \nabla_y g(y_t) - \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle \\
&\quad + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L}{2}\mathbb{E}\|\tilde{y}_{t+1} - y_t\|^2 \\
&\geq \mathbb{E}g(\tilde{y}_{t+1}) + \gamma\|\mathbb{E}[w_t]\|^2 + \langle \mathbb{E}[w_t], y^* - y_t\rangle + \mathbb{E}\langle \nabla_y g(y_t) - \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle \\
&\quad + \frac{\mu}{2}\|y^* - y_t\|^2 - \frac{L\gamma^2}{2}\mathbb{E}\|\omega_t\|^2 \\
&\geq \mathbb{E}g(\tilde{y}_{t+1}) + \langle \mathbb{E}[w_t], y^* - y_t\rangle + \mathbb{E}\langle \nabla_y g(y_t) - \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle \\
&\quad + \frac{\mu}{2}\|y^* - y_t\|^2 + \left(\gamma - \frac{L\gamma^2}{2}\right)\|\mathbb{E}[\omega_t]\|^2 - \frac{L\gamma^2}{2}Var[\omega_t]
\end{aligned}
$$

where Var denotes the variance. By definition of y^*, we have $g(y^*) \geq g(\tilde{y}_{t+1})$. Thus, we obtain

$$
\begin{aligned}
0 &\geq \langle \mathbb{E}[w_t], y^* - y_t\rangle + \mathbb{E}\langle \nabla_y g(y_t) - \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle \\
&\quad + \frac{\mu}{2}\|y^* - y_t\|^2 + \left(\gamma - \frac{L\gamma^2}{2}\right)\|\mathbb{E}[\omega_t]\|^2 - \frac{L\gamma^2}{2}Var[\omega_t]
\end{aligned} \tag{4.59}
$$

Considering the upper bound of the second term $\langle \nabla_y g(y_t) - w_t, y^* - y_{t+1}\rangle$, we have

$$
\begin{aligned}
&- \mathbb{E}\langle \nabla_y g(y_t) - \mathbb{E}[w_t], y^* - \tilde{y}_{t+1}\rangle \\
&= -\langle \nabla_y g(y_t) - \mathbb{E}[w_t], y^* - y_t\rangle + \langle \nabla_y g(y_t) - \mathbb{E}[w_t], \mathbb{E}[\omega_t]\rangle \\
&\leq \frac{1}{\mu}\|\nabla_y g(y_t) - \mathbb{E}[w_t]\|^2 + \frac{\mu}{4}\|y^* - y_t\|^2] + \frac{1}{\mu}\|\nabla_y g(y_t) - \mathbb{E}[w_t]\|^2 + \frac{\mu\gamma^2}{4}\|\mathbb{E}[\omega_t]\|^2 \\
&= \frac{2}{\mu}\|\nabla_y g(y_t) - \mathbb{E}[w_t]\|^2 + \frac{\mu}{4}\|y^* - y_t\|^2 + \frac{\mu\gamma^2}{4}\|\mathbb{E}[\omega_t]\|^2.
\end{aligned}
$$

Combining with Eq. 4.59:

$$0 \geq \langle \mathbb{E}[w_t], y^* - y_t \rangle - \frac{2}{\mu}\|\nabla_y g(y_t) - \mathbb{E}[w_t]\|^2 + \Big(\gamma - \frac{3L\gamma^2}{4}\Big)\|\mathbb{E}[w_t]\|^2 + \frac{\mu}{4}\|y^* - y_t\|^2 - \frac{L\gamma^2}{2}Var[\omega_t]$$

By $y_{t+1} = y_t - \gamma\alpha_t\omega_t$, we have:

$$\begin{aligned}\mathbb{E}\|y_{t+1} - y^*\|^2 &= \mathbb{E}\|y_t - \gamma\alpha_t\omega_t - y^*\|^2 = \|y_t - y^*\|^2 - 2\gamma\alpha_t\langle \mathbb{E}[\omega_t], y_t - y^*\rangle + \gamma^2\alpha_t^2\mathbb{E}[\|\omega_t\|^2] \\ &\leq \Big(1 - \frac{\mu\gamma\alpha_t}{2}\Big)\|y_t - y^*\|^2 - 2\gamma\alpha_t\Big(\gamma - \frac{\gamma\alpha_t}{2} - \frac{3L\gamma^2}{4}\Big)\|\mathbb{E}[\omega_t]\|^2 \\ &\quad + \frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(y_t) - \mathbb{E}[w_t]\|^2 + (L\gamma^3\alpha_t + \gamma^2\alpha_t^2)Var[\omega_t]\end{aligned}$$

Then since we choose $\gamma < \frac{1}{2L}$, $\alpha_t < 1$, we obtain:

$$\begin{aligned}\mathbb{E}\|y_{t+1} - y^*\|^2 \leq (1 - \frac{\mu\gamma\alpha_t}{2})\|y^* - y_t\|^2 - \frac{\gamma^2\alpha_t}{4}\|\mathbb{E}[\omega_t]\|^2 \\ + \frac{4\gamma\alpha_t}{\mu}\|\nabla_y g(x_t, y_t) - \mathbb{E}[w_t]\|^2 + \frac{3\gamma^2\alpha_t}{2}Var[\omega_t].\end{aligned}$$

This completes the proof. □

4.10 More Experimental Details

In this section, we introduce more details of the experiments.

4.10.1 Federated Data Cleaning

The formulation of the problem is as follows:

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq \frac{1}{M}\sum_{m=1}^{M} f^{(m)}(x, y_x^{(m)}) = \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N_m^{(val)}}\sum_{n=1}^{N_m^{(val)}}\Theta(y_x;\xi_{m,n}^{val})\Big)$$
$$\text{s.t. } y_x = \arg\min_{y\in\mathbb{R}^d} g(x,y) = \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N_m^{(tr)}} x_{m,n}\Theta(y;\xi_{m,n}^{tr})$$

In the above formulation, we have M clients, each client $m \in [M]$ has a pair of (noisy) training set $\{\xi_{m,n}^{tr}\}_{n=1}^{N_m^{(tr)}}$ and validation set $\{\xi_{m,n}^{val}\}_{n=1}^{N_m^{(val)}}$, and $x_{m,n}, n \in [N_m^{(tr)}]$ are weights for training samples, y is the parameter of a model, and we denote the model by Θ. Note that y_x is the model learned over the weighted training set. We fit a model with 3 fully connected layers for the MNIST dataset. We also use L_2 regularization with coefficient 10^{-3} to satisfy the strong convexity condition.

In the Experiments, for FedNest and CommFedBiO, we choose learning rate 1 and hyper-learning rate 10000, for FedBiO, we choose learning rate 0.5, hyper learning rate 1000, for FedBiOAcc, we choose δ as 30, u as 10000, c_η as 0.2, C_γ as 0.2, τ as 0.01, η as 200 and γ as 1.

4.10.2 Federated Hyper-Representation Learning

$$\min_{x\in\mathbb{R}^p} h(x) \coloneqq \frac{1}{M}\sum_{m=1}^{M} f^{(m)}(x, y_x^{(m)}) = \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N_m}\sum_{n=1}^{N_m}\Big(\frac{1}{N_{m,n}^{val}}\sum_{i=1}^{N_{m,n}^{val}}\Theta(x, y_x^{(\mathcal{T}_{m,n})};\xi_i^{val})\Big)\Big)$$

$$\text{s.t. } y_x^{(\mathcal{T}_{m,n})} = \arg\min_{y \in \mathbb{R}^d} g^{(\mathcal{T}_{m,n})}(x, y) = \frac{1}{N_{m,n}^{tr}} \sum_{i=1}^{N_{m,n}^{tr}} \Theta(x, y; \xi_i^{tr})$$

In the above formulation, we have M clients, each client $m \in [M]$ has N_m tasks and each task $\mathcal{T}_{m,n}$ is defined by a pair of training set $\{\xi_i^{tr}\}_{i=1}^{N_{m,n}^{tr}}$ and validation set $\{\xi_i^{val}\}_{i=1}^{N_{m,n}^{val}}$. Θ defines the model, x is the parameter of the backbone model and y is the parameter of the linear classifier. In summary, the lower level problem is to learn the optimal linear classifier y given the backbone x, and the upper level problem is to learn the optimal backbone parameter x.

The Omniglot dataset includes 1623 characters from 50 different alphabets and each character consists of 20 samples. We create the Federated version of the Omniglot dataset. Firstly, we follow the experimental protocols of [89] to divide the alphabets to train /validation/test with 33/5/12, respectively. Then we distribute three alphabets to a client, in other words, we consider 11 clients in experiments. As in the non-distributed setting, we perform N-way-K-shot classification, more specifically, for each task, we randomly sample N characters from the alphabet over that client and for each character, we sample K samples for training and 15 samples for validation. We augment the characters by performing rotation operations (multipliers of 90 degrees). We use a 4-layer convolutional neural network where each convolutional layer has 64 filters of 3×3 [91]. For the MiniImageNet, it has 64 training classes and 16 validation classes. We distribute the training classes into four clients, similar to Omniglot, we also perform the N-way-K-shot classification. We use a 4-layer convolutional neural network where each convolutional layer has 64 filters of 3×3 [91] for experiments.

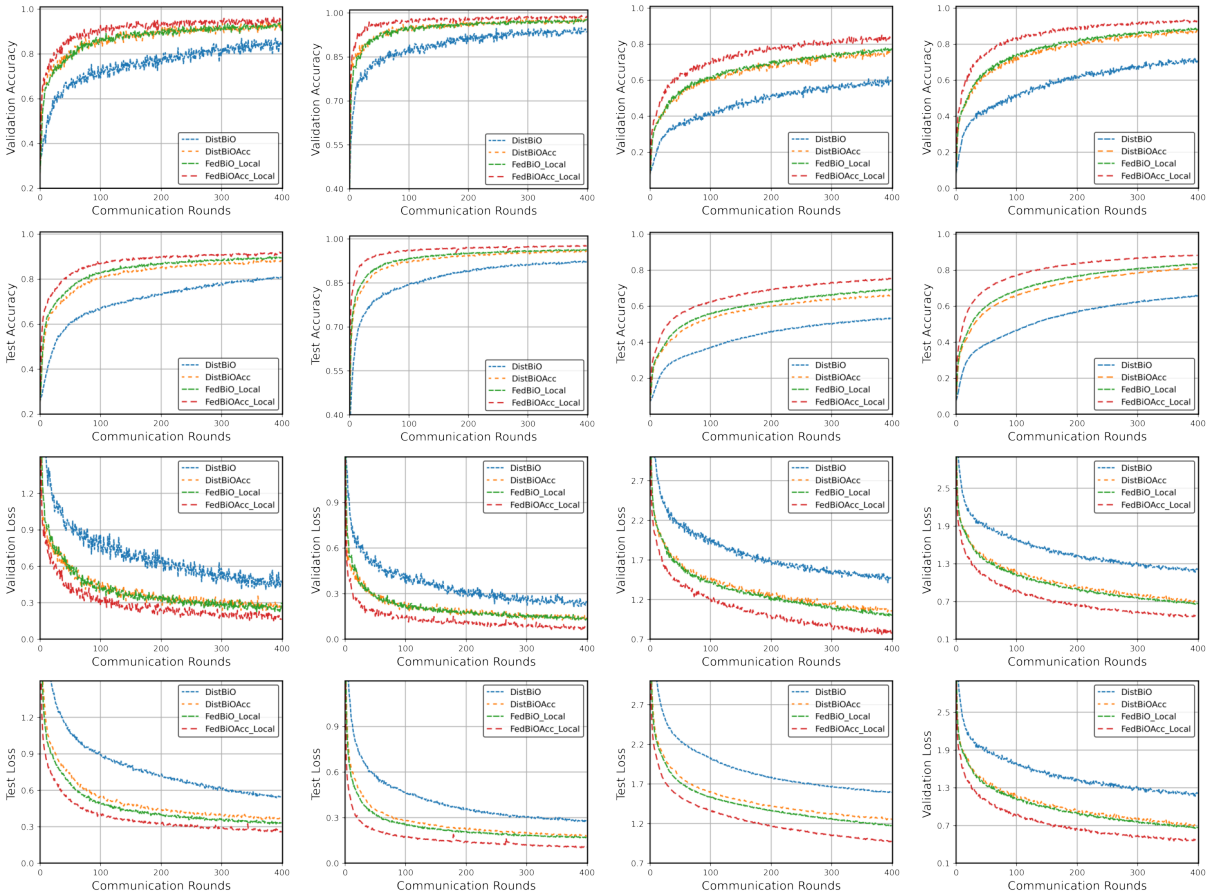


Figure 4.5: Results for the Omniglot Dataset. From Left to Right: 5-way-1-shot, 5-way-5-shot, 20-way-1-shot, 20-way-5-shot.

In the Experiments for Omniglot, for FedBiO, we choose learning rate 0.4, hyper learning rate 1, τ 0.5, for FedBiOAcc, we choose δ as 2, u as 10000, C_η as 100, τ as 0.5, *eta* as 1 and γ as 0.4. For MiniImageNet, for FedBiO, we choose learning rate 0.05, hyper learning rate 0.1, τ 0.01, for FedBiOAcc, we choose δ as 2, u as 10000, C_η as 100, τ as 0.01, *eta* as 1 and γ as 0.05.

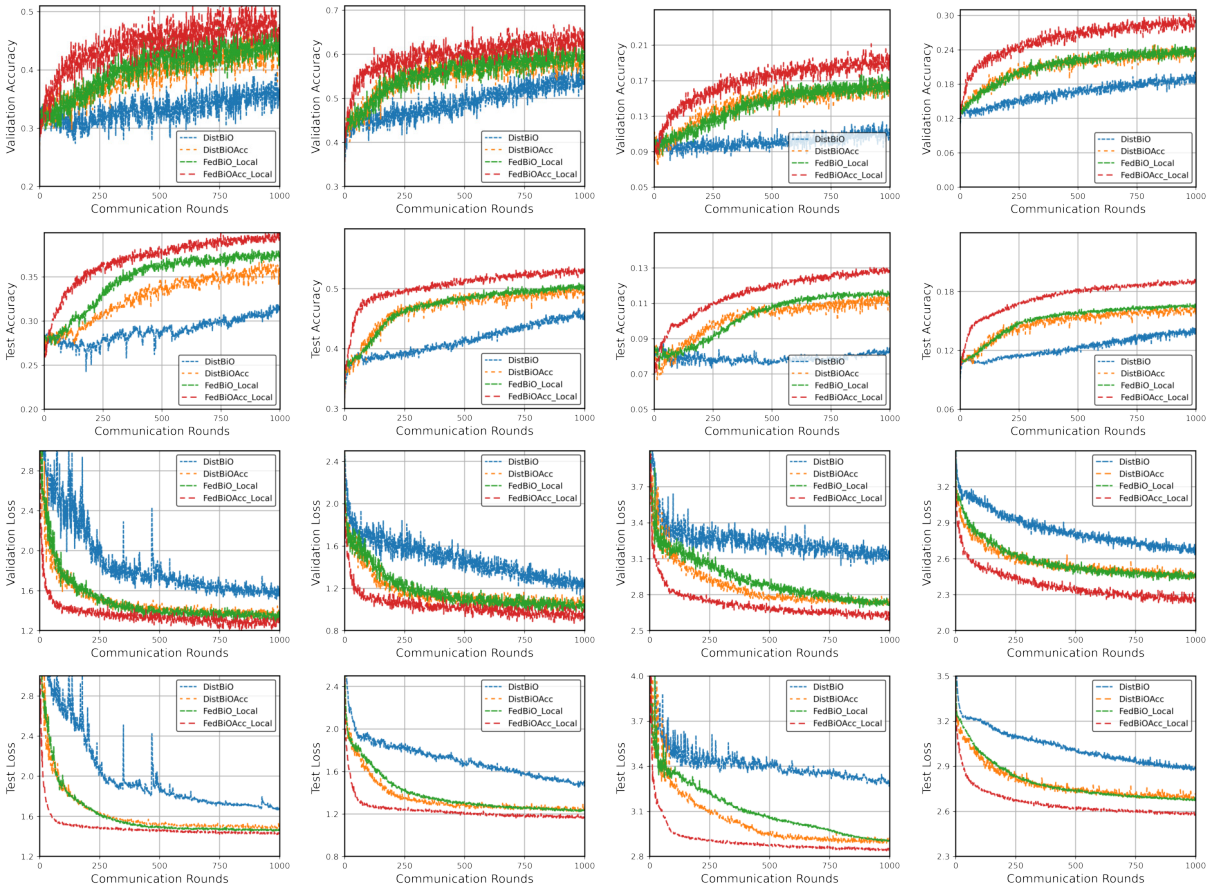


Figure 4.6: Results for the MiniImageNet Dataset. From Left to Right: 5-way-1-shot, 5-way-5-shot, 20-way-1-shot, 20-way-5-shot.

Chapter 5: Resolving the Tug-of-War: A Separation of Communication and Learning in Federated Learning

5.1 Introduction

In Federated Learning (FL) [92], a set of clients jointly solve a machine learning task under the coordination of a central server. The process of FL involves two category of operations: Learning and Communication. For the learning operation, each client optimizes their local objectives, and for the communication operation, clients exchange local parameters to facilitate knowledge sharing. In the existing FL pipeline, both Learning and Communication operations hinge on the same set of parameters. However, these two operations have fundamentally divergent requirements, akin to two teams engaged in a tug-of-war, each pulling in opposite directions. In fact, on the communication side, it is imperative that the parameters of all clients reside in a uniform space. Moreover, to mitigate the high communication costs, a major bottleneck in FL, it is beneficial to maintain these parameters within a low-dimensional space. In contrast, on the learning side, given the heterogeneity of devices, including variations in hardware and data distribution, it is advantageous to allow the parameter space to differ across clients. Furthermore, to accommodate the implementation of state-of-the-art large-scale machine learning models (such as

Transformers [121]), a high-dimensional parameter space is desirable. In summary, a huge discrepancy of requirements to parameter selection exists between the communication and the learning in FL.

To mitigate this discrepancy, we opt to 'break the rope' in this tug-of-war scenario. More precisely, we propose a two-layer structure on clients: a communication layer and a learning layer. As shown in Figure 5.1, we denote this framework as **FedSep**: The communication layer connects with the server and uploads/downloads parameter, while the learning layer performs local objective optimization. Then the communication layer and the learning layer are connected through encode/decode operations. The decode operation maps the parameter of the communication layer to the parameter of the learning layer and the encode operation performs the mapping in the opposite direction. More specifically, we formulate the decode operation as solving a minimization problem and does not assume an explicit relation between the parameter of communication layer and the parameter of the learning layer. In summary, we aim to use two distinct sets of parameters in FedSep to resolve the 'competition' between the communication and learning operations, meanwhile, the two set of parameters are close connected through decode/encode operations. Mathematically, FedSep can be formulated as solving a federated bilevel problem. The upper level problem corresponds to the learning problems on clients, while the lower level problem is the minimization problem in the decode operation. We propose an efficient algorithm to solve this bilevel problem, which is composed of three consecutive steps: Decode the communication parameter, Learning on local problems and Encode the learning parameter to the communication parameter. The convergence of the algorithm is guaranteed.

To demonstrate the flexibility of FedSep in incorporating the contradictory requirements of the communication and learning, we study two real-world FL tasks. In the first task, we study the communication-efficient FL task. In this task, we set the dimension of the communication parameter to be much smaller than the learning parameter, furthermore, we decode the communication parameter through a LASSO problem. In the second task, we study the model-heterogeneous FL task. In this task, the learning parameters on different clients have different dimension. We set the communication parameter to be the parameter of the server model, and the learning parameter be a subset of the communication parameter (server model). In particular, the decode operation is to select the subset adaptively based on the client's local data. We empirically verify the superior performance of our methods compared to other baselines. Finally, we summarize the main **contributions** of our work as follows:

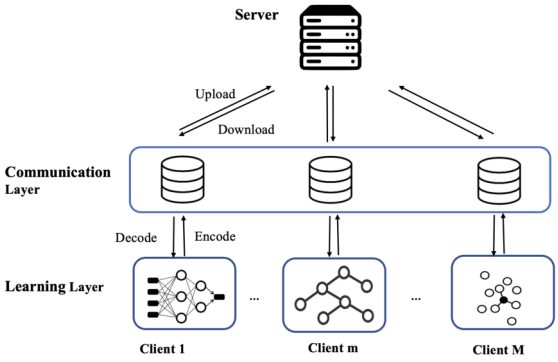


Figure 5.1: The structure of FedSep. FedSep follows the one-server-multiple-clients structure, in particular, each client has two layers: the communication layer and the learning layer. The two layers are connected through the decode and encode operations.

1. We propose a novel two-layer Federated Learning framework, *i.e.* **FedSep**, where the communication layer and the learning layer are separated on clients;
2. We let the decode operation from the communication layer to the learning layer be solving a minimization problem, and therefore, the **FedSep** framework has a bilevel optimization formulation. We propose an efficient algorithm to solve the bilevel

problem and show that its convergence rate matches that of the standard Federated Learning algorithms;

3. We apply the **FedSep** framework to solve two important FL tasks: Communication-Efficient FL and Model-Heterogeneous FL. Numerical experiments show the superior performance of the methods based on FedSep.

Notations. ∇ denotes the full gradient, ∇_x is the partial derivative for variable x, and higher-order derivatives follow similar rules. $||\cdot||_2$ is ℓ_2-norm for vectors and the spectral norm for matrices. AB denotes the multiplication of the matrix between the matrix A and B. $[K]$ represents the sequence of integers from 1 to K.

5.2 Related Works

Federated Learning. FL is a promising paradigm for performing machine learning tasks on distributed located data. Compared to traditional distributed learning in the data center, FL poses new challenges such as heterogeneity [96, 113, 115, 116], privacy [92, 122, 123] and communication bottleneck [124, 125, 126, 127, 128]. In particular, communication cost is one of the major bottlenecks in FL. A widely used approach to reduce communication burden is through compression: clients compress parameters before transmitting to the server. Various compressors [124, 125, 126, 127] have been studied in the literature. In particular, the local Top-K compressor selects the K largest coordinates of the parameter to transmit. From the view of FedSep: the vector of Top-K coordinates is the communication parameter, while the encode operation is finding these coordinates. Count Sketch [129] based compressors [128, 130] are also widely studied. Since the count sketch is linear,

from the point of view of FedSep: count sketch performs a linear mapping between the communication parameter and the learning parameter. Next, model-heterogeneous FL is also studied to solve the device heterogeneity issue. There are two main categories of heterogeneous model methods: knowledge-distillation (KD) [131, 132, 133] based and partial-training (PT) [134, 135, 136] based. KD-based methods treat the server model as the teacher and the clients' model as the student. A disadvantage of this category is the dependence on a public dataset. In contrast, PT-based methods select a subset of the server model for each client, and in the aggregation phase, the server simply aggregates the updated sub-models. We also develop a PT-based method where we select the sub-models adaptively based on the clients' local data.

Bilevel Optimization. Bilevel problem is a type of two-layer optimization problem. It includes an upper-level problem and a lower-level problem, and the upper-level problem relies on the solution of the lower-level problems. The study of bilevel problems dates back at least to the 1960s [16] and is followed by [8, 17, 18, 19]. In the machine learning community, Hyper-parameter Optimization problems [20, 21, 22, 23] are one of the most studied bilevel problems. Early bilevel optimization algorithms solved the lower problem exactly to update the upper variable. Recently, researchers developed algorithms that solve the lower problem approximately with a fixed number of steps and use the back-propagation technique to compute the approximate hyper-gradient [24, 25, 26, 27, 28]. More recently, single-loop algorithms [6, 11, 12, 13, 14, 15, 32, 33, 58, 65] based on alternative updates of lower and upper level problems are proposed. However, less work studies bilevel problems under the federated setting. [99, 100, 137] studies a type of federated bilevel problems where the lower level problems are federated. In particular, [99] solves the noisy label problem in

FL. Note that our FedSep is formulated as a type of federated bilevel problems, however, our problem is different from that in [99, 100], as our problem has a unique lower problem for each client.

5.3 Separating Communication and Learning in Federated Learning

In this section, we propose **FedSep**, a novel two-layer Federated Learning Framework. As shown in Figure 5.1, FedSep has the one-server-multiple-clients structure as in the classical Federated learning framework, we assume M clients in the training. The difference between the FedSep and classical FL framework lies in the client. Different from the classical FL, FedSep includes two separated layers on clients: a communication layer and a learning layer. The two layers have different responsibilities. As the name shows, the learning layer is responsible of finishing the learning task, *e.g.* fitting a neural network over a dataset. We assume that the learning layer on client $m \in [M]$ has parameter $y^{(m)} \in \mathbb{R}^{d^{(m)}}$. Note that the choice of $y^{(m)}$ should be adapted to the specific setting of each client, such as the data distribution, hardware resources *etc.*. Next, the communication layer performs communication with the server (other clients), which includes the **upload** and **download** operations. We assume the communication layer on client m has parameter $x \in \mathbb{R}^p$. Note that the parameter of communication layer has identical formulation for all clients. Finally, the communication layer and the learning layer are connected through the **encode** and **decode** operations.

To model the various potential relationship between the communication and learning layer, we abstract the decode operation as solving a minimization problem, and the encode

Algorithm 11 Separating Communication and Learning in FL (**FedSep**)

Input: Initial states x_1; learning rates γ, η; mini-batchsize b_x, b_y
for $t = 1$ **to** T **do**
  Randomly sample a subset $\mathcal{M}_t$ of clients;
  for $m \in \mathcal{M}_t$ in parallel **do**
    // Decode stage, estimate $y_x^{(m)} = Dec^{(m)}\{x\}$;
    Receive the global state x_t from the server and set $x^{(m)} = x_t$, and initialize $y_0^{(m)}$;
    for $i = 1$ to I_{dec} **do**
      Randomly sample a minibatch of b_y samples $\mathcal{B}_y$;
      $y_i^{(m)} = y_{i-1}^{(m)} - \gamma \nabla_y g^{(m)}(x^{(m)}, y_{i-1}^{(m)}, \mathcal{B}_y)$
    end for
    Set $\hat{y}_0^{(m)} = y_{I_{dec}}^{(m)}$ as the estimation of $Dec^{(m)}(x^{(m)})$;
    // Learning stage, optimize $f^{(m)}(y)$;
    for $i = 1$ to I **do**
      Randomly sample a minibatch of b_y samples $\mathcal{B}_{\hat{y}}$;
      $\hat{y}_i^{(m)} = \hat{y}_{i-1}^{(m)} - \eta \nabla f^{(m)}(\hat{y}_{i-1}^{(m)}; \mathcal{B}_{\hat{y}})$;
    end for
    // Encode stage, encode the update of the learning layer back to the communication layer;
    Set $\Delta\hat{y}^{(m)} = \hat{y}_I^{(m)} - \hat{y}_0^{(m)}$ and compute $\Delta\hat{x}_t^{(m)} = \widetilde{Enc}^{(m)}\{\Delta\hat{y}^{(m)}\}$, where $\widetilde{Enc}^{(m)}\{\cdot\}$ is defined in Eq. (5.3) and we choose $|\mathcal{B}_x| = b_x$.
  end for
  $x_{t+1} = x_t - \frac{1}{|\mathcal{M}_t|} \sum_{m \in \mathcal{M}_t} \eta_g \Delta\hat{x}_t^{(m)}$
end for

operation be its reverse operation. As a result, we have the following bilevel formulation for the FedSep framework.

$$\min_{x \in \mathbb{R}^p} h(x) := \frac{1}{M} \sum_{m=1}^{M} h^{(m)}(x) := \frac{1}{M} \sum_{m=1}^{M} f^{(m)}(y_x^{(m)}), \quad y_x^{(m)} = \underset{y^{(m)} \in \mathbb{R}^{d^{(m)}}}{\arg\min}\, g^{(m)}(x, y^{(m)}) \tag{5.1}$$

In Eq. (5.1), M is the number of clients; $f^{(m)}(y), m \in [M]$ denotes the local problem solved by the client m; $g^{(m)}(x, y)$ is the minimization problem solved in the decode operation by the client m; x is the parameter of the communication layer; $y^{(m)}$ denotes the parameter of the learning layer on the client m.

Next, we propose a three-stage algorithm to solve Eq. (5.1). As shown by Algo-

rithm 11, we perform T global steps, and at each step $t \in T$, we randomly choose a subset of clients $\mathcal{M}_t$ to perform the training and then the server performs aggregation. The **Client training** is divided into three stages: **Decode stage** (lines 7-12), **Learning stage** (lines 14-17), and **Encode Stage** (line 19).

Decode Stage. In Eq. (5.1), the decode operation solves the lower optimization problem $g^{(m)}(x, y)$ to obtain $y_x^{(m)}$. We denote the decode operator on the m_{th} client by $Dec^{(m)}\{\cdot\}$, thus we have: $y_x^{(m)} = Dec^{(m)}\{x\}$. In Algorithm 11, we solve $g^{(m)}(x, y)$ approximately with I_{dec} steps of stochastic gradient descent (Line 8-11) and use the output as an approximation of $y_x^{(m)}$, *i.e.*. $y_{I_{dec}}^{(m)} = \widetilde{Dec}^{(m)}(x^{(m)}) \approx y_x^{(m)}$, where $\widetilde{Dec}^{(m)}(\cdot)$ represents the approximation of the exact decoder $Dec^{(m)}\{\cdot\}$.

Learning Stage. Next, we perform I steps of stochastic gradient descent to solve the local learning problem $f^{(m)}(y)$ (Line 14 - 17). This stage is similar to the local gradients in classical FL framework. Note we get $\hat{y}_I^{(m)}$ as the updated learning parameter $y^{(m)}$, more formally, we use $\Delta\hat{y}^{(m)} = \hat{y}_I^{(m)} - \hat{y}_0^{(m)} = \hat{y}_I^{(m)} - y_{I_{dec}}^{(m)}$ to represent the update in the learning stage.

Encode Stage. After the learning stage, we get the update of the local learning problem $\Delta\hat{y}^{(m)}$. Suppose we denote the encode operator $Enc^{(m)}\{\cdot\}$, then we get the update of the communication parameter as $\Delta\hat{x}^{(m)} = Enc^{(m)}(\Delta\hat{y}^{(m)})$ (Line 19). In particular, we choose the following encode operator:

$$Enc^{(m)}\{\cdot\} := -\nabla_x y_x^{(m)} = \nabla_{xy} g^{(m)}(x, y_x^{(m)}) \Big(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\Big)^{-1} \tag{5.2}$$

where $y_x^{(m)} = Dec^{(m)}(x)$ is the output of the exact decode operation and the second equal-

ity can be derived following the implicit function theorem under mild assumptions [6]. To reduce the computation complexity, we use Neumann series to approximate the inverse operation in Eq. (5.2). More specifically, we have the following approximation: $\left(\nabla_{yy} g^{(m)}(x,y)\right)^{-1} \approx \tau \sum_{q=0}^{Q} (I - \tau \nabla_{yy} g^{(m)}(x,y))^q$, where Q and τ are some constants. Since in the decode stage, we use $y_x^{(m)} \approx y_{I_{dec}}^{(m)}$, we get a stochastic approximation of Eq. (5.2) as:

$$\widetilde{Enc}^{(m)}\{\cdot\} \coloneqq \sum_{q=0}^{Q} \tau \nabla_{xy} g^{(m)}(x, y_{I_{dec}}^{(m)}; \mathcal{B}_x)(I - \tau \nabla_{yy} g^{(m)}(x, y_{I_{dec}}^{(m)}; \mathcal{B}_x))^q \tag{5.3}$$

Remark 115. *We use Eq. (5.2) as the encode operator due to the following fact about the hyper-gradient:*

$$\nabla h^{(m)}(x) = \nabla_x y_x^{(m)}(\nabla f^{(m)}(y_x^{(m)}))$$

Note that suppose $\Delta\hat{y}^{(m)} = -\nabla f^{(m)}(y_x^{(m)})$, which means the learning layer performs one step of gradient descent with learning rate 1 in the learning stage of Algorithm 11, then we have: $\nabla h^{(m)}(x) = Enc\{\Delta\hat{y}^{(m)}\} = \Delta x^{(m)}$. Therefore, we update the communication parameter x such that it optimize the overall problem $\nabla h^{(m)}(x)$.

Server Aggregation. Finally, the server aggregates the local updates of communication parameter $\Delta\hat{x}^{(m)}, m \in \mathcal{M}_t$ to get the new communication parameter as shown in Line 21 of Algorithm 11.

Difference with existing bilevel algorithms. Bilevel optimization problems are widely studied in the literature [6, 12]. However, FedSep cannot directly apply existing algorithms. In a standard bilevel algorithm, each step of update to the upper level variable

requires (approximately) solving of the corresponding lower level problem. However, in each epoch of FedSep, we only solve the lower level problem once and perform multiple steps of update to the upper level variable.

5.3.1 Convergence Analysis

In this section, we provide the convergence analysis of Algorithm 11. Before we state the convergence result, we first make the following assumptions.

Assumption 116. *Function $f^{(m)}(y)$ is possibly non-convex and $g^{(m)}(x,y)$ is μ-strongly convex* w.r.t *y for any given x.*

Assumption 117. *Function $f^{(m)}(y)$ is L-smooth and has C_f-bounded gradient.*

Assumption 118. *Function $g^{(m)}(x,y)$ is L-smooth; $\nabla_{xy} g^{(m)}(x,y)$ and $\nabla_{y^2} g^{(m)}(x,y)$ are Lipschitz continuous with constants L_{xy} and L_{y^2}, respectively.*

Assumption 119. *We have unbiased stochastic first-order and second-order gradient oracle with bounded variance.*

Note that Assumption 116-Assumption 118 are standard assumptions used to analyze bilevel problems [6, 12]. Assumption 119 is standard in analyzing stochastic optimization problems. For a full version of Assumption 119, please refer to Assumption C.1 in the Appendix.

In the learning stage of Algorithm 11, we perform I steps of updates to optimize the learning problem, and the update has the form of: $\Delta\hat{y}^{(m)} = \hat{y}_I^{(m)} - \hat{y}_0^{(m)} = \sum_{i=0}^{I-1} -\eta \nabla f^{(m)}(\hat{y}_i^{(m)}; \mathcal{B}_{\hat{y}})$, therefore we have: $\Delta\hat{x}_t^{(m)} = \widetilde{Enc}^{(m)}\{\Delta\hat{y}^{(m)}\} = \sum_{i=0}^{I-1} \eta \widetilde{Enc}^{(m)}\{-\nabla f^{(m)}(\hat{y}_i^{(m)}; \mathcal{B}_{\hat{y}})\}$. We de-

note $\Delta\hat{x}_{t,i}^{(m)} = \widetilde{Enc}^{(m)}\{\nabla f^{(m)}(\hat{y}_i^{(m)}; \mathcal{B}_{\hat{y}})\}$. **Then the main effort of the proof is bounding the estimation error of $\boldsymbol{\Delta\hat{x}_{t,i}^{(m)}}$ to the hyper-gradient $\boldsymbol{\nabla h^{(m)}(x)}$.**

More specifically, we first show that $\Delta\hat{x}_{t,i}^{(m)}$ estimates a term $\mu_i^{(m)}$ with bounded variance and bias as stated in the following proposition:

Proposition 120. *Suppose Assumptions 116-118 and 119 hold and $\tau < \frac{1}{L}$, we have $\|\mathbb{E}_\xi[\Delta\hat{x}_{t,i}^{(m)}] - \mu_i^{(m)}\| \leq G_1$, where $\mu_i^{(m)} = -\nabla_{xy} g^{(m)}(x, y_{I_{dec}}^{(m)})\Big(\nabla_{yy} g^{(m)}(x, y_{I_{dec}}^{(m)})\Big)^{-1} \nabla f^{(m)}(\hat{y}_i^{(m)})$, and $\mathbb{E}\|\Delta\hat{x}_{t,i}^{(m)} - \mathbb{E}_\xi[\Delta\hat{x}_{t,i}^{(m)}]\|^2 \leq G_2^2$. G_1 and G_2 are some constants related to Q in Eq. (5.3) and mini-batch size.*

Please refer to Proposition C.3 for the specific form of the constants in Proposition 120. Next, we have that $\mu_i^{(m)} \approx \nabla h^{(m)}(x)$ as shown in the following proposition:

Proposition 121. *Suppose Assumptions 117 and 118 hold, the following statements hold:*

a) $\|\mu_i^{(m)} - \nabla h^{(m)}(x)\|^2 \leq 2\hat{L}^2\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + 2\kappa^2 L^2 \|\hat{y}_i^{(m)} - y_x^{(m)}\|^2$, *where* $\hat{L} = O(\kappa^2)$.

b) $h^{(m)}(x)$ *is Lipschitz continuous in* x *with constant* $\bar{L}$ *i.e., for any given* $x_1, x_2 \in \mathbb{R}^p$, *we have* $\|\nabla h^{(m)}(x_2) - \nabla h^{(m)}(x_1)\|^2 \leq \bar{L}^2\|x_2 - x_1\|$, *where* $\bar{L} = O(\kappa^3)$.

where we denote the condition number as $\kappa = L/\mu$.

Please refer to Proposition C.2 for the proof. Proposition 121 shows that $\mu_i^{(m)}$ estimates the true hyper-gradient $\nabla h^{(m)}(x)$ with two types of errors: the estimation error of the decode operation ($\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2$) and the drift of the learning process ($\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2$), where Lemma C.4 and Lemma C.5 show bounds for these two errors. Furthermore, Proposition 121.b) shows that $h^{(m)}(x)$ is smooth. Next, we are ready to show the convergence of Algorithm 11:

Theorem 122. *Suppose Assumptions 116-119 hold, and we run T iterations of Algorithm 11, with the learning rates satisfy $\gamma < \frac{1}{L}$, $\eta\eta_g < \frac{1}{2I\bar{L}}$, then we have:*

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\|\nabla h(x_t)\|^2 + \frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right) \\
&\leq \frac{2h(x_1)}{TI\eta\eta_g} + \frac{\eta\eta_g\bar{L}G_2^2}{b_x M} + 12I^2\kappa^2L^2\eta^2C_f + \frac{4I^2\kappa^2L^2\eta^2\sigma^2}{b_y} \\
&\quad + \frac{4(3\kappa^2L^2+\hat{L}^2)\gamma\sigma^2}{\mu b_y} + 2G_1^2 + 2(3\kappa^2L^2+\hat{L}^2)(1-\mu\gamma)^{I_{dec}}C_0
\end{aligned}$$

where the expectation $\mathbb{E}_\xi$ is w.r.t *the noise of stochastic gradient. b_x, b_y denotes the batch size, and G_1, G_2 and C_0 are some constants.*

Please refer to Theorem C.8 in the Appendix for the proof. We can further control the noise terms in Theorem 122 by choosing the learning rates carefully:

Corollary 123. *Suppose we choose the learning rates as $\gamma = \min(\frac{1}{2L}, (\frac{1}{C_\gamma T})^{1/2})$, $\eta = \min\left(1, \left(\frac{8Ib_xM\bar{L}h(x_1)}{TG_2^2}\right)^{1/2}, \left(\frac{4\bar{L}h(x_1)}{C_\eta I^2 T}\right)^{1/3}\right)$ and $\eta_g = \frac{1}{2I\bar{L}}$, then we have:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big(\|\nabla h(x_t)\|^2 + \frac{1}{2I}\sum_{i=1}^{I}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\Big) = O\left(\frac{\kappa^3}{T} + \left(\frac{\kappa^5}{T}\right)^{1/2} + \left(\frac{\kappa^6}{T^2}\right)^{1/3} + \tilde{G}\right)$$

where $\tilde{G} = \kappa^2(1-\tau\mu)^{2(Q+1)} + \kappa^4(1-\mu\gamma)^{I_{dec}}$, C_η and C_γ are some constants.

To reach an ϵ stationary point, we need to run Algorithm 11 with $T = O(\kappa^5\epsilon^{-2})$ number of iterations, furthermore, we need $Q = O(\kappa\log\left(\frac{\kappa}{\epsilon}\right))$, $I_{dec} = O(\kappa\log\left(\frac{\kappa}{\epsilon}\right))$. Note that this matches the iteration complexity of standard FL algorithms [118] up to some logarithm factors (Q and I_{dec}). Note that Corollary 123 shows that the following term converges to 0: $\mathbb{E}\|\nabla h(x_t)\|^2 + \frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2$. In fact, for the first term, it shows

that x_t reaches to a stationary point of the bilevel problem $h(x)$, meanwhile, since we have:

$$\mathbb{E}_{\xi}[\bar{\Delta}\hat{x}_{t,i}] = \frac{1}{M}\sum_{m=1}^{M}\sum_{q=0}^{Q}\tau\nabla_{xy}g^{(m)}(x, y_{I_{dec}}^{(m)}) \times (I - \tau\nabla_{yy}g^{(m)}(x, y_{I_{dec}}^{(m)}))^{q}\nabla f^{(m)}(\hat{y}_i^{(m)})$$

Therefore, if $\|\nabla_{xy}g^{(m)}(x,y)\|$ is lower-bounded and $\|\nabla_{yy}g^{(m)}(x,y)\|$ is upper-bounded, we also get the learning parameter $\hat{y}_i^{(m)}$ converges to the stationary point of the local learning problem $f^{(m)}(y)$.

5.4 Application of FedSep to Real-world FL Problems

In FedSep, the separation of the communication layer and the learning layer makes it able to solve various challenges in Federated Learning. In this section, we apply FedSep to solve two challenging problems in Federated Learning: communication-efficient FL and model-heterogeneous FL.

5.4.1 Communication-efficient Federated Learning

Communication-cost is one of the major bottlenecks in Federated Learning due to slow connections between clients and the server. In Federated Learning, compression is commonly employed to mitigate communication costs, where clients compress the local updates before transferring to the server. A common compressor is choosing the Top-K coordinates, however, this approach is only good at reducing the upload communication cost. Since clients have different Top-K coordinates, the aggregated updates are often dense. More complicated approaches can save both upload and download communication cost, such as the Count-sketch based compressor [130]. In fact, we can develop a simple

yet effective approach to reduce the communication cost based on FedSep.

In our FedSep framework, suppose we have the learning parameter $\theta \in \mathbb{R}^d$, and the communication parameter $\omega \in \mathbb{R}^p$. We choose $p \ll d$ to have a communication efficient federated learning algorithm. More specifically, we consider the following formulation:

$$\min_{\omega \in \mathbb{R}^p} \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}(\theta_\omega^{(m)}; \mathcal{D}_{tr}^{(m)}) \ s.t. \ \theta_\omega^{(m)} = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{2} \left\| S^{(m)}\theta - \omega \right\|_2^2 + \beta \left\| \theta \right\|_1 \tag{5.4}$$

where $\mathcal{D}_{tr}^{(m)}$ denotes the training distribution of the m_{th} client; $\mathcal{L}(\cdot)$ denotes the loss function. In particular, $S^{(m)} \in \mathbb{R}^{p \times d}$ $(p \ll d)$ is a random sketch matrix whose coordinates are sampled from Gaussian distribution. We choose the quadratic optimization problem (LASSO) as the decode function.

Eq. (5.4) is a special case of Eq. (5.1): θ corresponds to the learning parameter y, ω corresponds to the communication parameter x, $\mathcal{L}(\theta; \mathcal{D}_{tr}^{(m)})$ corresponds to the learning problem $f^{(m)}(y)$ and the decoding problem is a LASSO problem. We can solve it through Algorithm 11.

The LASSO problem of the decode operation involves a non-smooth L_1 regularization term and we solve it with the standard proximal gradient method [138]. For the encode operation, we have:

$$Enc^{(m)}\{\cdot\} = -\nabla_\omega \theta_\omega^{(m)} = \gamma S^{(m)} U \Big(I - \Big(I - \gamma (S^{(m)})^T S^{(m)} \Big) U \Big)^{-1} \tag{5.5}$$

where I is the identity matrix and $U = Diag\{\mathbb{I}_{\gamma\beta}(\theta_\omega^{(m)} + \gamma (S^{(m)})^T (\omega - S^{(m)}\theta_\omega^{(m)}))\}$ where $\mathbb{I}_{\gamma\beta}(\cdot)$ is an indicator operator, which outputs 1 if the absolute value of the input is greater

than $\gamma\beta$. Please refer to Appendix A.1 for more details of the proximal gradient method and the encode operator.

Remark 124. *If we set $\beta = 0$ in Eq. (5.4),* i.e. *we remove the sparsity constraints, both encode and decode operators have explicit solutions. More specifically, we have the decode operator $Dec^{(m)}\{\cdot\} \coloneqq \left((S^{(m)})^T S^{(m)}\right)^{-1}(S^{(m)})^T$ and the encode operator $Enc^{(m)}\{\cdot\} \coloneqq S^{(m)}\left((S^{(m)})^T S^{(m)}\right)^{-1}$. If we choose $S^{(m)} = S$ for all $m \in [M]$, we get a linear compressor.*

5.4.2 Model-Heterogeneous Federated Learning

In practical Federated Learning applications, the scale of the model is inherently limited by the on-device resources of the participating clients, which often exhibit significant diversity. As a result, we can choose different scale of models based on the available resources of each individual client. One widely-used approach is the sub-model extraction, *i.e.* each client selects a part of the server model to perform local training. Different strategies can be used to do extraction: fixed [134, 139], random [135] and roll [136]. In contrast, we provide a data-dependent approach to select the sub-models. As shown by the following formulation:

$$\min_{\omega\in\mathbb{R}^p} \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}(\theta_\omega^{(m)}; \mathcal{D}_{tr}^{(m)}) \tag{5.6}$$

$$s.t.\ \theta_\omega^{(m)} = a_\omega^{(m)} \odot \omega,\ a_\omega^{(m)} = \arg\min_{a\in\{0,1\}^p} \mathcal{L}(a \odot \omega; \mathcal{D}_{val}^{(m)}) + \beta\mathcal{R}(T(a), p^{(m)}T_{tol})$$

where $\mathcal{D}_{tr}^{(m)}$ denotes the training distribution; $\mathcal{L}(\cdot)$ denotes the loss function and θ and ω are the learning parameter. Note that ω denotes the parameter of the full model and θ is

the parameter of the sub-model. In particular, we denote a mask vector $a_\omega^{(m)}$, whose value is from $\{0,1\}$. $\odot$ represents the coordinate-wise multiplication.

For the decode operation, we have $\theta_\omega^{(m)} = a_\omega^{(m)} \odot \omega$. So we first need to find an optimal mask $a_\omega^{(m)}$, instead of solving the complicated integer programming as in Eq. (5.6), we solve the following relaxed continuous problem:

$$a_\omega^{(m)} = \underset{a\in[0,1]^p}{\arg\min} \mathcal{L}(a \odot \omega; \mathcal{D}_{val}^{(m)}) + \beta \mathcal{R}(T(a), p^{(m)} T_{tol}) \tag{5.7}$$

Eq. (5.7) includes two parts. $\mathcal{L}(a \odot \omega; \mathcal{D}_{val}^{(m)})$ measures the loss of the extracted sub-model over $\mathcal{D}_{val}^{(m)}$. The second part $\mathcal{R}(T(a), p^{(m)} T_{tol})$ is called resource loss [140]. T_{tol} denotes FLOPS of the server model and $T(a)$ denotes FLOPS of the sub-model. Therefore, $p^{(m)} \in [0,1]$ controls the size of the sub-model. Following [140], we choose $\mathcal{R}(\cdot)$ as: $\mathcal{R}(T(a), p^{(m)} T_{tol}) = \log\Big(max(T(a), p^{(m)} T_{tol})/p^{(m)} T_{tol}\Big)$. In summary, we choose the 'best' sub-model which keeps the loss value small and meanwhile, satisfies the FLOPS constraint. Note that this is in contrast to the data-independent sub-model selection methods [134, 135, 136].

Next for the encode operator $Enc\{\cdot\}$, we follow the definition of Eq. (5.2) to have:

$$Enc^{(m)}\{\cdot\} = -\nabla_\omega \theta_\omega^{(m)} = -\nabla_\omega \Big(a_\omega^{(m)} \odot \omega\Big) = -Diag\{a_\omega^{(m)}\} - Diag\{\omega\} \nabla_\omega a_\omega^{(m)} \tag{5.8}$$

Where the last equality follows from the chain rule. Finally, combine Eq. (5.7) and Eq. (5.8), we can solve Eq. (5.6) with Algorithm 11.

Remark 125. *For the encode operator Eq. (5.8), we have* $Enc^{(m)}(\cdot) \approx -Diag\{a_\omega^{(m)}\}$ *by*

omitting the second term in Eq. (5.8), where $\Delta\theta$ denotes the updates to the sub-model θ. As shown in Line 21 of Algorithm 11, we have the update of the server model be $\Delta\omega = \frac{1}{|\mathcal{M}_t|}\sum_{m\in\mathcal{M}_t}\eta_g Diag\{a_\omega^{(m)}\}\Delta\theta$. In other words, the server simply aggregates the updates of the sub-model, this is the aggregation rule widely used in the partial training literature [134, 135, 136].

5.5 Numerical Experiments

In this section, we perform numerical experiments to validate the efficacy of FedSep in solving communication-efficient FL and model-heterogeneous FL as discussed in Section 5.4. The code is written with Pytorch [141], and the Federated Learning environment is simulated via Pytorch.Distributed Package. We used servers with AMD EPYC 7763 64-core CPU and 8 NVIDIA V100 GPUs to run our experiments.

5.5.1 Experiments for Communication-Efficient Federated Learning

In this section, we provide numerical experiments for the communication-efficient FL task discussed in Section 5.4.1. We first introduce the experimental settings.

Dataset. We consider MNIST [52] and CIFAR-10 [142] in our experiments. We create the federated version of these datasets by evenly distributing the training samples among 10 clients. We consider an I.I.D setting and a Non-I.I.D setting. For the I.I.D setting, each client has data samples from all 10 classes, while for the Non-I.I.D setting, each client only has data samples from 2 classes.

Model. For experiments related to MNIST, we use a four-layer convolutional neural

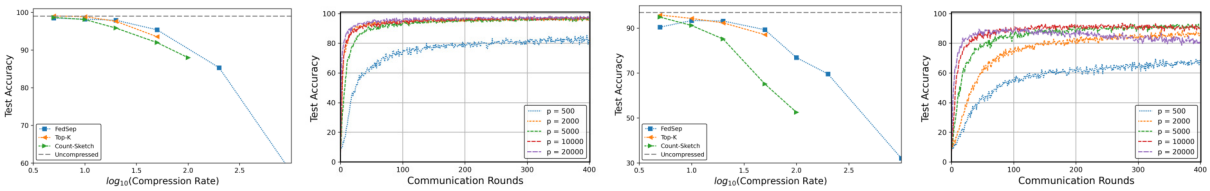


Figure 5.2: Test Accuracy w.r.t Communication Rate for FedSep and other baseline methods for MNIST Dataset. The first two plots show results under the I.I.D case, and the last two plots show results for the Non-I.I.D case. p in the second and the fourth plot is the dimension of the communication parameter. The local learning steps are set as $I = 5$.

network. The first three layers contain 32 kernels of size 3×3 and the fourth layer contains 32 kernels of size 2×2. For the experiments related to CIFAR-10, we increase the number of kernels to 64. The total number of parameters of the model is around 10^5. For the sketch matrix $S^{(m)}$ of our FedSep, we let all clients choose the same sketch matrix, and we test the dimension of the communication parameter $p \in \{10, 100, 500, 1000, 2000, 5000, 10000, 20000\}$.

Baselines and Hyper-parameters. For baselines, we first compare with the uncompressed baseline (the communication parameter and the learning parameter are in the same dimension); Next we compare with the local Top-K compressor (with error feedback [143]) and the Count-Sketch compressor [130]. Please refer to the Appendix for the hyper-parameter selection of FedSep and baselines.

The experimental results are summarized in Figure 5.2 for MNIST (Figure 3 in the Appendix for CIFAR-10). As shown by the figures, FedSep outperforms other baselines. In particular, our FedSep achieves much better performance in the high-compression-rate regime and can get similar performance as other baselines in the low-compression-rate regime. Please refer to the Appendix for more ablation studies.

Table 5.1: **Test accuracy comparison between FedSep with other model heterogeneous FL baseline methods.** High data heterogeneity represents $K = 2$ for CIFAR-10 and $K = 20$ for CIFAR-100; Lower data heterogeneity represents $K = 5$ for CIFAR-10 and $K = 50$ for CIFAR-100.

	Method	High Data Heterogeneity		Low Data Heterogeneity	
		CIFAR-10	**CIFAR-100**	**CIFAR-10**	**CIFAR-100**
KD-based	FedDF [131]	73.81 (± 0.42)	31.87 (± 0.46)	76.55 (± 0.32)	37.87 (± 0.31)
	DS-FL [132]	65.27 (± 0.53)	29.12 (± 0.51)	68.44 (± 0.47)	33.56 (± 0.55)
	Fed-ET [133]	**78.66 (± 0.31)**	**35.78 (± 0.45)**	**81.13 (± 0.28)**	**41.58 (± 0.36)**
PT-based	HeteroFL [134]	63.90 (± 2.74)	52.38 (± 0.80)	73.19 (± 1.71)	57.44 (± 0.42)
	Federated Dropout [135]	46.64 (± 3.05)	45.07 (± 0.07)	76.20 (± 2.53)	46.40 (± 0.21)
	ZeroFL [144]	64.61 (± 2.18)	51.39 (± 0.45)	83.31 (± 0.78)	53.62 (± 0.51)
	FedDST [145]	67.65 (± 1.27)	54.21 (± 0.34)	84.57 (± 0.28)	54.97 (± 0.44)
	Flash [139]	67.08 (± 1.46)	54.92 (± 0.29)	84.61 (± 0.37)	55.04 (± 0.32)
	FedRolex [136]	69.44 (± 1.50)	56.57 (± 0.15)	84.45 (± 0.36)	58.73 (± 0.33)
	FedSep (Ours)	**71.13 (± 0.94)**	**58.16 (± 0.25)**	**84.61 (± 0.37)**	**61.41 (± 0.29)**
	Homogeneous (smallest)	38.82 (± 0.88)	12.69 (± 0.50)	46.86 (± 0.54)	19.70 (± 0.34)
	Homogeneous (largest)	**75.74 (± 0.42)**	**60.89 (± 0.60)**	**84.48 (± 0.58)**	**62.51 (± 0.20)**

5.5.2 Experiments for Model-Heterogeneous Federated Learning

In this section, we provide numerical experiments for the task of a model-heterogeneous FL task discussed in Section 5.4.2. First, we introduce the experimental settings.

Dataset. We consider CIFAR-10 and CIFAR-100 [142] in our experiments. We create Federated datasets by evenly distributing images among clients; we have 100 clients in our experiments. We create data heterogeneity by controlling the number of classes K a client can have [136]. For CIFAR-10, we test $K \in \{2, 4, 5, 8\}$, while for CIFAR-100, we test $K \in \{5, 10, 20, 50\}$. Note that smaller values of K mean higher heterogeneity of the data. For FedSep, the validation and training set are the same.

Model. We choose ResNet-18 [146] in the experiments. Following the setting in [134, 136], we replace the batch normalization layer with static batch normalization and add a scalar module after each convolution layer. Instead of using the coordinate mask as defined in Eq. (5.6), we select kernels in each convolutional layer of ResNet-18; furthermore, we

parameterize the mask a with a neural network $HN(\phi)$ [140], *i.e.* $a = HN(\phi)$; finally, we reuse the masks and only update the masks every two epochs, this stabilizes the training. This reduces the dimension of a that we need to optimize. In experiments, the size of the submodel p (Eq. (5.6)) of a client is randomly chosen from $\{1, 0.5, 0.25, 0.125, 0.0625\}$, where the ratio p is *w.r.t* the server model. For our FedSep, after optimizing Eq. (5.7), we select the top p percent kernels of each convolutional layer by the value of the leaned mask a, and then we only use the selected kernels to perform the training. In the encode operation, we only evaluate the first term of Eq. (5.8) as stated in Remark 125.

Baselines and Hyper-parameters. We compare with both state-of-the-art Knowledge Distillation-based methods: FedDF [131], DS-FL [132] and Fed-ET [133], and Partial Training Based methods: HeteroFL [134], ZeroFL [144], Federated Dropout [135], FedDST [145], Flash [139] and FedRolex [136]. Fjord [147] gets similar performance as HeteroFL, so we do not include it in the comparision. For ZeroFL, FedDST and Flash, we consider their heterogeneous-model version by varying the compression rate among clients.

The experimental results are summarized in Table 5.1. As shown in the table, our FedSep gets comparable performance with the Knowledge Distillation based methods, which needs additional public data, furthermore, FedSep outperforms the partial training based methods, including the state-of-the-art FedRolex [136] method. This result shows the effectiveness of the adaptive sub-model extraction strategy used by our FedSep. Our method chooses the most important part of the model at each step for clients, thus our method converges faster than other data-independent methods. For more experimental results, please refer to the Appendix.

5.6 Proof of Theorems

In this section, we present the proofs for Algorithm 11. First, we denote some notation for clarity. Recall that we denote $\Delta\hat{y}^{(m)} = \hat{y}_I^{(m)} - \hat{y}_0^{(m)} = \sum_{i=0}^{I-1} -\eta \nabla f^{(m)}(\hat{y}_i^{(m)}; \mathcal{B}_{\hat{y}})$ to be the update of the learning problem on client m. Then we denote $\mu_i^{(m)}$ as:

$$\mu_i^{(m)} = -\nabla_{xy} g^{(m)}(x, y_{I_{dec}}^{(m)}) \Big(\nabla_{yy} g^{(m)}(x, y_{I_{dec}}^{(m)})\Big)^{-1} \nabla f^{(m)}(\hat{y}_i^{(m)})$$

and denote $\mu^{(m)}$ as:

$$\mu^{(m)} = -\nabla_{xy} g^{(m)}(x, y_{I_{dec}}^{(m)}) \Big(\nabla_{yy} g^{(m)}(x, y_{I_{dec}}^{(m)})\Big)^{-1} \Delta\hat{y}^{(m)}$$

while in Algorithm 11, we use the stochastic encoder $\widetilde{Enc}\{\cdot\}$ to compute $\Delta\hat{x}_t^{(m)}$,

$$\Delta\hat{x}_t^{(m)} = \sum_{q=0}^{Q} \tau \nabla_{xy} g^{(m)}(x, y_{I_{dec}}^{(m)}; \mathcal{B}_x)(I - \tau \nabla_{yy} g^{(m)}(x, y_{I_{dec}}^{(m)}; \mathcal{B}_x))^q \Delta\hat{y}^{(m)}$$

Since $x^{(m)}$ is not updated during local steps, we omit the superscript when clear from the context. We further denote $\Delta\hat{x}_{t,i}^{(m)}$ as

$$\Delta\hat{x}_{t,i}^{(m)} = -\sum_{q=0}^{Q} \tau \nabla_{xy} g^{(m)}(x, y_{I_{dec}}^{(m)}; \mathcal{B}_x)(I - \tau \nabla_{yy} g^{(m)}(x, y_{I_{dec}}^{(m)}; \mathcal{B}_x))^q \nabla f^{(m)}(\hat{y}_i^{(m)}; \mathcal{B}_{\hat{y}})$$

Lastly, recall the hyper-gradient of $h^{(m)}(x)$ has the form of:

$$\nabla h^{(m)}(x) = -\nabla_{xy} g^{(m)}(x, y_x^{(m)}) \Big(\nabla_{yy} g^{(m)}(x, y_x^{(m)})\Big)^{-1} \nabla f^{(m)}(y_x^{(m)})$$

Assumption 126 (Assumption 4)**.** *We have unbiased stochastic first order and second order derivative oracle with bounded variance, more specifically, denote* $z = (x, y)$*, we have:*

a) we have $\nabla f^{(m)}(y;\xi)$*, such that:* $E[\nabla f^{(m)}(y;\xi)] = \nabla f^{(m)}(y)$ *and* $var(\nabla f^{(m)}(y;\xi)) \leq \sigma^2$*;*

b) we have $\nabla g^{(m)}(z;\xi)$*, such that:* $E[\nabla g^{(m)}(z;\xi)] = \nabla g^{(m)}(z)$ *and* $var(\nabla g^{(m)}(z;\xi)) \leq \sigma^2$*;*

c) we have $\nabla_{y^2} g^{(m)}(z,\xi)$*, such that:* $E[\nabla_{y^2} g^{(m)}(z;\xi)] = \nabla_{y^2} g^{(m)}(z)$ *and* $var(\nabla_{y^2} g^{(m)}(z;\xi)) \leq \sigma^2$*;*

d) we have $\nabla_{xy} g^{(m)}(z;\xi)$*, such that:* $E[\nabla_{xy} g^{(m)}(z;\xi)] = \nabla_{xy} g^{(m)}(z)$ *and* $var(\nabla_{xy} g^{(m)}(z;\xi)) \leq \sigma^2$*;*

Proposition 127. *Suppose Assumptions 117 and 118 hold, the following statements hold:*

a) $\|\mu_i^{(m)} - \nabla h^{(m)}(x)\|^2 \leq 2\hat{L}^2 \|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + 2\kappa^2 L^2 \|\hat{y}_i^{(m)} - y_x^{(m)}\|^2$*, where* $\hat{L} = O(\kappa^2)$*.*

b) $h^{(m)}(x)$ *is Lipschitz continuous in* x *with constant* $\bar{L}$ *i.e., for any given* $x_1, x_2 \in \mathbb{R}^p$*, we have* $\|\nabla h^{(m)}(x_2) - \nabla h^{(m)}(x_1)\|^2 \leq \bar{L}^2 \|x_2 - x_1\|$ *where* $\bar{L} = O(\kappa^3)$*.*

where we denote the condition number as $\kappa = L/\mu$*.*

Proof. We prove the Part a) here. Proof of b) and can be referred in Lemma 2.2 of [6].

$$
\begin{aligned}
&\|\mu_i^{(m)} - \nabla h^{(m)}(x)\| \\
&= \left\| \nabla_{xy} g(x, y_{I_{dec}}^{(m)}) \left(\nabla_{yy} g(x, y_{I_{dec}}^{(m)}) \right)^{-1} \nabla f^{(m)}(\hat{y}_i^{(m)}) \right.
\end{aligned}
$$

$$
\begin{aligned}
&\quad - \nabla_{xy} g(x, y_x^{(m)}) \left(\nabla_{yy} g(x, y_x^{(m)}) \right)^{-1} \nabla f^{(m)}(y_x^{(m)}) \Big\| \\
&\leq \left\| \nabla_{xy} g(x, y_{I_{dec}}^{(m)}) - \nabla_{xy} g(x, y_x^{(m)}) \right\| \left\| \left(\nabla_{yy} g(x, y_x^{(m)}) \right)^{-1} \nabla f^{(m)}(\hat{y}_i^{(m)}) \right\| \\
&\quad + \left\| \nabla_{xy} g(x, y_x^{(m)}) \right\| \left\| \left(\nabla_{yy} g(x, y_{I_{dec}}^{(m)}) \right)^{-1} \nabla f^{(m)}(\hat{y}_i^{(m)}) - \left(\nabla_{yy} g(x, y_x^{(m)}) \right)^{-1} \nabla f^{(m)}(y_x^{(m)}) \right\| \\
&\leq \frac{C_f L_{xy}}{\mu} \left\| y_{I_{dec}}^{(m)} - y_x^{(m)} \right\| + L \left\| \left(\nabla_{yy} g(x, y_{I_{dec}}^{(m)}) \right)^{-1} \nabla f^{(m)}(\hat{y}_i^{(m)}) - \left(\nabla_{yy} g(x, y_x^{(m)}) \right)^{-1} \nabla f^{(m)}(y_x^{(m)}) \right\| \\
&\leq \frac{C_f L_{xy}}{\mu} \left\| y_{I_{dec}}^{(m)} - y_x^{(m)} \right\| + C_f L \left\| \left(\nabla_{yy} g(x, y_{I_{dec}}^{(m)}) \right)^{-1} - \left(\nabla_{yy} g(x, y_x^{(m)}) \right)^{-1} \right\| \\
&\quad + \kappa \left\| \nabla f^{(m)}(\hat{y}_i^{(m)}) - \nabla f^{(m)}(y_x^{(m)}) \right\| \\
&\leq \left(\frac{C_f L_{xy}}{\mu} + \frac{\kappa C_f L_{yy}}{\mu} \right) \left\| y_{I_{dec}}^{(m)} - y_x^{(m)} \right\| + \kappa \left\| \nabla f^{(m)}(\hat{y}_i^{(m)}) - \nabla f^{(m)}(y_x^{(m)}) \right\|
\end{aligned}
$$

Suppose we denote $\hat{L} = \left(\frac{C_f L_{xy}}{\mu} + \frac{\kappa C_f L_{yy}}{\mu} \right)$, then we have:

$$
\left\| \mu_i^{(m)} - \nabla h^{(m)}(x) \right\|^2 \leq 2\hat{L}^2 \left\| y_{I_{dec}}^{(m)} - y_x^{(m)} \right\|^2 + 2\kappa^2 \left\| \Delta \nabla f^{(m)}(\hat{y}_i^{(m)}) - \nabla_y f(y_x^{(m)}) \right\|^2
$$

which completes the proof. □

Proposition 128. *(Lemma 4 and 7 in [32]) Suppose Assumptions 117, 118 and 119 hold and $\tau < \frac{1}{L}$, we have*

a) $\|\mathbb{E}_\xi[\Delta \hat{x}_{t,i}^{(m)}] - \mu_i^{(m)}\| \leq G_1$, *where* $G_1 = \kappa(1 - \tau\mu)^{Q+1} C_f$

b) $\mathbb{E}\|\Delta \hat{x}_{t,i}^{(m)} - \mathbb{E}_\xi[\Delta \hat{x}_{t,i}^{(m)}]\|^2 \leq G_2^2$, *where* $G_2^2 = (2C_f^2 + 12C_f^2 L^2 \tau^2 (Q+1)^2 + 4C_f^2 L^2 (Q+2)(Q+1)^2 \tau^4 \sigma^2)/b_x$

where we assume the mini-batch has $|\mathcal{B}_x| = b_x$

5.6.1 Lower Problem Solution Error

Lemma 129. *When $\gamma < \frac{1}{L}$, we have:*

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_{I_{dec}}^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 \leq \frac{(1-\mu\gamma)^{I_{dec}}}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_0^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 + \frac{2\gamma\sigma^2}{\mu b_y}$$

Proof. First, follow the property of the strong convex function, we have:

$$\mathbb{E}\left\|y_i^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 \leq (1-\mu\gamma)\mathbb{E}\left\|y_{i-1}^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 + \frac{2\gamma^2\sigma^2}{b_y}$$

where we choose $\gamma < 1/L$. Next, we telescope from $i = 1 \to I_{dec}$ to have

$$\begin{aligned}\mathbb{E}\left\|y_{I_{dec}}^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 &\leq (1-\mu\gamma)^{I_{dec}}\mathbb{E}\left\|y_0^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 + \frac{2\gamma^2\sigma^2}{b_y}\sum_{i=0}^{I}(1-\mu\gamma)^i \\ &\leq (1-\mu\gamma)^{I_{dec}}\mathbb{E}\left\|y_0^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 + \frac{2\gamma\sigma^2}{\mu b_y}\end{aligned}$$

Average over all clients, we get the claim in the lemma. □

Lemma 130. *For $I \geq 1$, than we have:*

$$\begin{aligned}&\frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2 \\ &\leq \frac{3I}{M}\sum_{m=1}^{M}\mathbb{E}\|\hat{y}_0^{(m)} - y_x^{(m)}\|^2 + 6I^2\eta^2\frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\|\nabla f^{(m)}(\hat{y}_i^{(m)})\|^2 + \frac{2I^3\eta^2\sigma^2}{b_y}\end{aligned}$$

Proof. By the update rule in Algorithm 11, we have:

$$\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2 = \mathbb{E}\|\hat{y}_{i-1}^{(m)} - \eta\nabla f^{(m)}(\hat{y}_{i-1}^{(m)}; \mathcal{B}_{\hat{y}}) - y_x^{(m)}\|^2$$

$$
\begin{aligned}
&\leq (1+\frac{1}{I})\mathbb{E}\|\hat{y}_{i-1}^{(m)} - y_x^{(m)}\|^2 + (1+I)\eta^2\|\nabla f^{(m)}(\hat{y}_{i-1}^{(m)}; \mathcal{B}_{\hat{y}})\|^2 \\
&\leq (1+\frac{1}{I})\mathbb{E}\|\hat{y}_{i-1}^{(m)} - y_x^{(m)}\|^2 + 2I\eta^2\|\nabla f^{(m)}(\hat{y}_{i-1}^{(m)}; \mathcal{B}_{\hat{y}})\|^2 \\
&\leq (1+\frac{1}{I})\mathbb{E}\|\hat{y}_{i-1}^{(m)} - y_x^{(m)}\|^2 + 2I\eta^2\|\nabla f^{(m)}(\hat{y}_{i-1}^{(m)})\|^2 + \frac{2I\eta^2\sigma^2}{b_y}
\end{aligned}
$$

where the first inequality is by the generalized inequality. Next we telescope over i, to obtain:

$$
\begin{aligned}
\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2 &\leq \sum_{j=1}^{i}(1+\frac{1}{I})^{i-j}\Big(2I\eta^2\|\nabla f^{(m)}(\hat{y}_j^{(m)})\|^2 + \frac{2I\eta^2\sigma^2}{b_y}\Big) + (1+\frac{1}{I})^i\mathbb{E}\|\hat{y}_0^{(m)} - y_x^{(m)}\|^2 \\
&\leq (1+\frac{1}{I})^I\sum_{i=1}^{I}\Big(2I\eta^2\|\nabla f^{(m)}(\hat{y}_i^{(m)})\|^2 + \frac{2I\eta^2\sigma^2}{b_y}\Big) + (1+\frac{1}{I})^I\mathbb{E}\|\hat{y}_0^{(m)} - y_x^{(m)}\|^2 \\
&\leq 3\mathbb{E}\|\hat{y}_0^{(m)} - y_x^{(m)}\|^2 + 6I\eta^2\sum_{i=1}^{I}\|\nabla f^{(m)}(\hat{y}_i^{(m)})\|^2 + \frac{2I^2\eta^2\sigma^2}{b_y}
\end{aligned}
$$

The third inequality uses the inequality $log(1 + a/x) \leq a/x$ for $x > -a$, so we have $(1 + a/x)^x \leq e^a$, Then we choose $a = 1$ and $x = I$. Finally, we use the fact that $e \leq 3$. It completes the proof by summing over all i. □

5.6.2 Descent Lemma

Lemma 131. *For all $t \in [T]$, the iterates generated satisfy:*

$$
\mathbb{E}\left\|\nabla h(x_t) - \mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 \leq \frac{1}{M}\sum_{m=1}^{M}\left(2\hat{L}^2\mathbb{E}\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + 2\kappa^2L^2\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2\right) + 2G_1^2
$$

Proof. By definition of $\bar{\Delta}\hat{x}_{t,i}$, $\mu_{t,i}^{(m)}$ and $\nabla h(x_t)$, we have:

$$
\begin{aligned}
&\mathbb{E}\left\|\nabla h(x_t) - \mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|\mathbb{E}_\xi[\Delta\hat{x}_t^{(m)}] - \nabla h^{(m)}(x_t)\right\|^2 \\
&\leq \frac{2}{M}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\mathbb{E}_\xi[\Delta\hat{x}_{t,i}^{(m)}] - \mu_{t,i}^{(m)}\right\|^2 + \left\|\mu_{t,i}^{(m)} - \nabla h^{(m)}(x_t)\right\|^2\right] \\
&\overset{(b)}{\leq} \frac{1}{M}\sum_{m=1}^{M}\left(2\hat{L}^2\mathbb{E}\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + 2\kappa^2\mathbb{E}\|\nabla f^{(m)}(\hat{y}_i^{(m)}) - \nabla f^{(m)}(y_x^{(m)})\|^2\right) + 2G_1^2 \\
&\leq \frac{1}{M}\sum_{m=1}^{M}\left(2\hat{L}^2\mathbb{E}\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + 2\kappa^2L^2\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2\right) + 2G_1^2
\end{aligned}
$$

where inequality (a) follows the generalized triangle inequality; inequality (b) follows the Proposition 127 and Proposition 128.

□

Lemma 132. *Suppose $\eta\eta_g \leq \frac{1}{2I\bar{L}}$ For $t \in [T]$, the iterates generated satisfy:*

$$
\begin{aligned}
\mathbb{E}[h(x_{t+1})] \leq \mathbb{E}[h(x_t)] - \frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 - \frac{\eta\eta_g}{4}\sum_{i=1}^{I}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_xM} + I\eta\eta_gG_1^2 \\
+ \frac{\eta\eta_g}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\left(\hat{L}^2\mathbb{E}\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + \kappa^2L^2\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2\right)
\end{aligned}
$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Using the smoothness of f we have:

$$
\begin{aligned}
\mathbb{E}[h(x_{t+1})] &\leq \mathbb{E}[h(x_t)] + \mathbb{E}\langle\nabla h(x_t), x_{t+1} - x_t\rangle + \frac{\bar{L}}{2}\mathbb{E}\|x_{t+1} - x_t\|^2 \\
&\overset{(a)}{=} \mathbb{E}[h(x_t)] - \eta_g\mathbb{E}\langle\nabla h(x_t), \mathbb{E}_\xi[\bar{\Delta}\hat{x}_t]\rangle + \frac{\eta_g^2\bar{L}}{2}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_t]\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_xM}
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbb{E}[h(x_t)] - \eta_g \eta \sum_{i=1}^{I} \mathbb{E}\langle \nabla h(x_t), \mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\rangle + \frac{I\eta^2\eta_g^2\bar{L}}{2}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} \\
&\overset{(b)}{=} \mathbb{E}[h(x_t)] - \frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 + \frac{\eta\eta_g}{2}\sum_{i=1}^{I}\mathbb{E}\|\nabla h(x_t) - \mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2 \\
&\quad - \left(\frac{\eta\eta_g}{2} - \frac{I\eta^2\eta_g^2\bar{L}}{2}\right)\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} \\
&\overset{(c)}{\leq} \mathbb{E}[h(x_t)] - \frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 - \frac{\eta\eta_g}{4}\sum_{i=1}^{I}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} + I\eta\eta_g G_1^2 \\
&\quad + \frac{\eta\eta_g}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\left(\hat{L}^2\mathbb{E}\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + \kappa^2 L^2\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2\right)
\end{aligned}
$$

where equality (a) follows from the iterate update given in Step 21 of Algorithm 11; (b) uses $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a-b\|^2]$; (c) follows the condition that $I\eta\eta_g \leq 1/2\bar{L}$ and Lemma 131. This completes the proof. □

5.6.3 Proof of Convergence Theorem

We prove the convergence of Algorithm 11 in this section.

Theorem 133. *Suppose we the learning rates*

$$
\eta = \min\left(1, \left(\frac{8Ib_x M\bar{L}h(x_1)}{TG_2^2}\right)^{1/2}, \left(\frac{4\bar{L}h(x_1)}{C_\eta I^2 T}\right)^{1/3}\right),\ \gamma = \min\left(\frac{1}{2L}, \left(\frac{1}{C_\gamma T}\right)^{1/2}\right)
$$

and $\eta_g = \frac{1}{2I\bar{L}}$*, then we have:*

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\|\nabla h(x_t)\|^2 + \frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right) \\
&= O\left(\frac{\kappa^3}{T} + \left(\frac{\kappa^5}{T}\right)^{1/2} + \left(\frac{\kappa^6}{T^2}\right)^{1/3} + \kappa^2(1-\tau\mu)^{2(Q+1)} + \kappa^4(1-\mu\gamma)^{I_{dec}}\right)
\end{aligned}
$$

To reach an ϵ stationary point, we choose $Q = O(\kappa \log\left(\frac{\kappa}{\epsilon}\right))$, $I_{dec} = O(\kappa \log\left(\frac{\kappa}{\epsilon}\right))$ and $T = O(\kappa^3 \epsilon^{-1.5})$ number of iterations.

Proof. By Lemma 132, and combine with Lemma 129 and Lemma 130 to have:

$$\begin{aligned}
\mathbb{E}[h(x_{t+1})] &\leq \mathbb{E}[h(x_t)] - \frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 - \frac{\eta\eta_g}{4}\sum_{i=1}^{I}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} + I\eta\eta_g G_1^2 \\
&\quad + \frac{\eta\eta_g}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\left(\hat{L}^2\mathbb{E}\|y_{I_{dec}}^{(m)} - y_x^{(m)}\|^2 + \kappa^2 L^2\mathbb{E}\|\hat{y}_i^{(m)} - y_x^{(m)}\|^2\right) \\
&\leq \mathbb{E}[h(x_t)] - \frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 - \frac{\eta\eta_g}{4}\sum_{i=1}^{I}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} + I\eta\eta_g G_1^2 \\
&\quad + \frac{I\hat{L}^2\eta\eta_g(1-\mu\gamma)^{I_{dec}}}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_0^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 + \frac{2I\hat{L}^2\eta\eta_g\gamma\sigma^2}{\mu b_y} + \frac{2I^3\kappa^2L^2\eta^3\eta_g\sigma^2}{b_y} \\
&\quad + \frac{3I\kappa^2L^2\eta\eta_g}{M}\sum_{m=1}^{M}\mathbb{E}\|\hat{y}_0^{(m)} - y_x^{(m)}\|^2 + \frac{6I^2\kappa^2L^2\eta^3\eta_g}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\|\nabla f^{(m)}(\hat{y}_i^{(m)})\|^2
\end{aligned}$$

By Algorithm 11, we have $\hat{y}_0^{(m)} = y_{I_{dec}}^{(m)}$, then we have:

$$\begin{aligned}
\mathbb{E}[h(x_{t+1})] &\leq \mathbb{E}[h(x_t)] - \frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 - \frac{\eta\eta_g}{4}\sum_{i=1}^{I}\mathbb{E}\left\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\right\|^2 + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} + I\eta\eta_g G_1^2 \\
&\quad + \frac{I(3\kappa^2L^2 + \hat{L}^2)\eta\eta_g(1-\mu\gamma)^{I_{dec}}}{M}\sum_{m=1}^{M}\mathbb{E}\left\|y_0^{(m)} - y_{x^{(m)}}^{(m)}\right\|^2 + \frac{2I\hat{L}^2\eta\eta_g\gamma\sigma^2}{\mu b_y} \\
&\quad + \frac{6I\kappa^2L^2\eta\eta_g\gamma\sigma^2}{\mu b_y} + \frac{6I^2\kappa^2L^2\eta^3\eta_g}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\|\nabla f^{(m)}(\hat{y}_i^{(m)})\|^2 + \frac{2I^3\kappa^2L^2\eta^3\eta_g\sigma^2}{b_y}
\end{aligned}$$

Assume that $\|y_0^{(m)} - y_{x^{(m)}}^{(m)}\|^2 \leq C_0$ for some constant C_0, and use Assumption 117. We sum over $t \in [T]$ to obtain:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left(\frac{I\eta\eta_g}{2}\mathbb{E}\|\nabla h(x_t)\|^2 + \frac{\eta\eta_g}{4}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right) \\
&\leq \frac{h(x_1)}{T} + \frac{I\eta^2\eta_g^2\bar{L}G_2^2}{2b_x M} + I\eta\eta_g G_1^2 + I(3\kappa^2L^2 + \hat{L}^2)\eta\eta_g(1-\mu\gamma)^{I_{dec}}C_0 + \frac{2I\hat{L}^2\eta\eta_g\gamma\sigma^2}{\mu b_y}
\end{aligned}$$

$$+\frac{6I\kappa^2L^2\eta\eta_g\gamma\sigma^2}{\mu b_y}+6I^3\kappa^2L^2\eta^3\eta_g C_f+\frac{2I^3\kappa^2L^2\eta^3\eta_g\sigma^2}{b_y}$$

Then we divide by $(I\eta\eta_g)/2$ on both sides and have:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\|\nabla h(x_t)\|^2+\frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right)\\&\leq\frac{2h(x_1)}{TI\eta\eta_g}+\frac{\eta\eta_g\bar{L}G_2^2}{b_xM}+12I^2\kappa^2L^2\eta^2C_f+\frac{4I^2\kappa^2L^2\eta^2\sigma^2}{b_y}\\&\quad+\frac{4(3\kappa^2L^2+\hat{L}^2)\gamma\sigma^2}{\mu b_y}+2G_1^2+2(3\kappa^2L^2+\hat{L}^2)(1-\mu\gamma)^{I_{dec}}C_0\end{aligned}$$

Next, we denote constant $C_\eta=\left(12\kappa^2L^2C_f+\frac{4\kappa^2L^2\sigma^2}{b_y}\right)$ and $C_\gamma=\frac{4(3\kappa^2L^2+\hat{L}^2)\sigma^2}{\mu b_y}$, and set $\eta_g=\frac{1}{2I\bar{L}}$, then we have:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\|\nabla h(x_t)\|^2+\frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right)\\&\leq\frac{4\bar{L}h(x_1)}{T\eta}+\frac{\eta G_2^2}{2Ib_xM}+C_\eta I^2\eta^2+C_\gamma\gamma+2G_1^2+2(3\kappa^2L^2+\hat{L}^2)(1-\mu\gamma)^{I_{dec}}C_0\end{aligned}$$

Then we choose

$$\eta=\min\left(1,\left(\frac{8Ib_xM\bar{L}h(x_1)}{TG_2^2}\right)^{1/2},\left(\frac{4\bar{L}h(x_1)}{C_\eta I^2T}\right)^{1/3}\right),\ \gamma=\min\left(\frac{1}{2L},\left(\frac{1}{C_\gamma T}\right)^{1/2}\right)$$

Then we obtain:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\|\nabla h(x_t)\|^2+\frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right)\\&\leq\frac{4\bar{L}h(x_1)}{T}+\left(\frac{2G_2^2\bar{L}h(x_1)}{Ib_xMT}\right)^{1/2}+\left(\frac{16I^2C_\eta\bar{L}^2h(x_1)^2}{T^2}\right)^{1/3}\end{aligned}$$

$$+\left(\frac{C_\gamma}{T}\right)^{1/2}+2G_1^2+2(3\kappa^2L^2+\hat{L}^2)(1-\mu\gamma)^{I_{dec}}C_0$$

Finally, since $\hat{L}=O(\kappa^2)$, $\bar{L}=O(\kappa^3)$ and $\mu\gamma=O(\kappa^{-1})$. Suppose we choose $I=O(1)$, then $C_\eta=O(\kappa^2)$, $C_\gamma=O(\kappa^5)$, and use $G_1=\kappa(1-\tau\mu)^{Q+1}C_f$ in Proposition 128, we have:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\|\nabla h(x_t)\|^2+\frac{1}{2I}\sum_{i=1}^{I}\mathbb{E}\|\mathbb{E}_\xi[\bar{\Delta}\hat{x}_{t,i}]\|^2\right)\\&=O\left(\frac{\kappa^3}{T}+\left(\frac{\kappa^5}{T}\right)^{1/2}+\left(\frac{\kappa^6}{T^2}\right)^{1/3}+\kappa^2(1-\tau\mu)^{2(Q+1)}+\kappa^4(1-\mu\gamma)^{I_{dec}}\right)\end{aligned}$$

and to reach an ϵ stationary point, we choose $Q=O(\kappa\log\left(\frac{\kappa}{\epsilon}\right))$, $I_{dec}=O(\kappa\log\left(\frac{\kappa}{\epsilon}\right))$ and $T=O(\kappa^5\epsilon^{-2})$ number of iterations. □

5.7 More Experimental Details

5.7.1 Communication-efficient FL

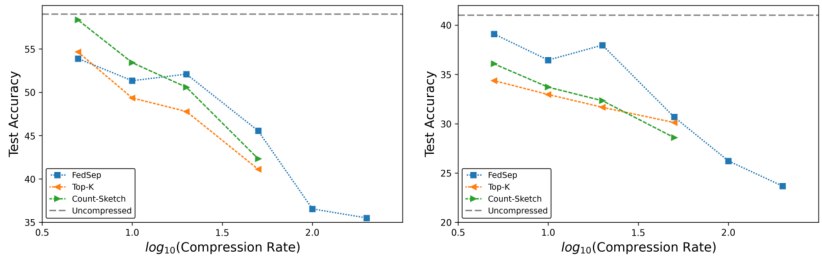


Figure 5.3: Test Accuracy w.r.t Communication Rate for FedSep and other baseline methods for CIFAR-10 Dataset. The left figure shows results under the I.I.D case, and the right figure show results for the Non-I.I.D case. The local learning steps are set as $I=5$.

In this section, we include more details related to the Communication-efficient FL. We first introduce details of some definitions.

Soft-threshold Operator. We use the definition of the soft threshold operator $Q_{\gamma\beta}(\cdot)$ in Section 4.1, formally, we have the following definition:

$$Q_{\gamma\lambda}(x) = sign(x)max(abs(x) - \gamma\lambda, 0) \tag{5.9}$$

Proximal Gradient Descent. we solve $\theta_{\omega}^{(m)}$ through the following update rule:

$$\theta^{+} = Q_{\gamma\beta}\Big(\theta + \gamma(S^{(m)})^{T}(\omega - S^{(m)}\theta)\Big) \tag{5.10}$$

where $Q_{\gamma\beta}(\cdot)$ is the soft-threshold operator and its definition is in Eq. (5.9) and γ is some constant. We denote the generalized gradient based on Eq. (5.10) as:

$$G^{*}(\theta) = \frac{1}{\gamma}\Big(\theta - \theta^{+}\Big) \tag{5.11}$$

In summary, for the decode operation $Dec\{\cdot\}$ in Eq. (5.4), we solve decode problem through gradient descent with the gradient operator defined by Eq. (5.11). As for the encode operation, we have $G^{*}(\theta_{\omega}^{(m)}) = 0$ by definition of $\theta_{\omega}^{(m)}$, then take derivation over ω on both sides, and use the chain rule, we get the expression in Eq. (5.5).

Next, we introduce more details of experimental settings.

Baselines and Hyper-Parameter. We first compare with the uncompressed baseline (the communication parameter and the learning parameter are in the same dimension); Next we compare with the local Top-K compressor and the Count-Sketch compressor [130]. For the count-sketch, we tune the size of sketch table; a typical good value for the length of the sketch table is half of the communication parameter dimension. For our FedSep, we

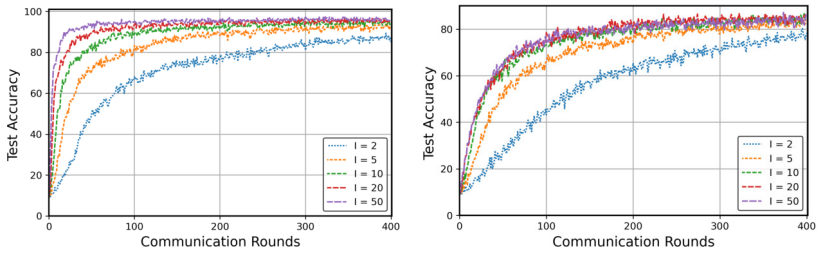


Figure 5.4: Test Accuracy w.r.t Communication Round for FedSep over the MNIST Dataset. The left figures shows results under the I.I.D case, and the right figure show results for the Non-I.I.D case. We vary the value of local step $I \in \{2, 5, 10, 20, 50\}$ in the figure, and we fix the dimension of the communication parameter as $p = 2000$.

tune the values of β and γ. A typical good value for γ is $[0.5, 5]$ and it depends on the dimension of the communication parameter p, for β, we choose $[0, 0.01]$. We use the SGD optimizer to optimize $\mathcal{L}(\theta_{\omega}^{(m)}; \mathcal{D}_{tr}^{(m)})$ for all methods, and set the learning rate at 0.1, and set $I_{dec} = 10$ for the decode operation, *i.e.* for solving the Lasso problem.

More Experimental Results. The experimental results for the CIFAR-10 dataset is included in Figure 5.3. We observe that FedSep also outperforms other baselines as in the MNIST data set. Finally, in Figure 5.4, we vary the number of local steps I under a fixed compression rate. As shown in the figure, our FedSep can benefit from more local steps. More specifically, for the homogeneous case, our FedSep can benefit from more local steps as large as $I = 50$, while for the heterogeneous case, increasing local steps brings little benefit to $I > 10$.

5.7.2 Model-Heterogeneous FL

In this section, we introduce more details of the experimental setting for Model-Heterogeneous FL task and present more experimentsl results.

Hyper-parameters. For our FedSep, we perform grid search to search the best

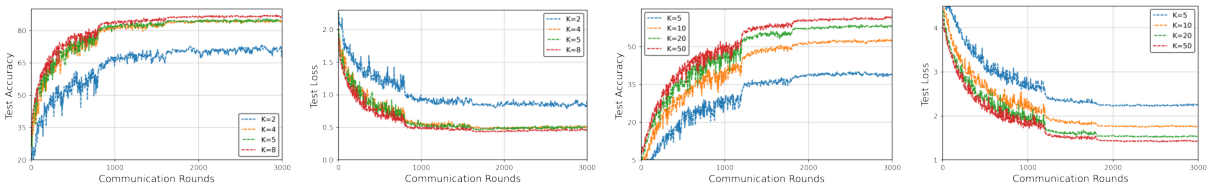


Figure 5.5: Test Accuracy and Loss w.r.t Communication Rounds for FedSep under different levels of data heterogeneity. K is the number of classes each client has. The first two plots show results for CIFAR-10, then the last two plots show results for CIFAR-100.

parameter setting. More specifically, we decode problem Eq. (5.7) with $\beta = 10$ and AdamW [148] with learning rate 0.01. Regarding the learning problem, for CIFAR-10, we use the Adam optimizer with learning rate 10^{-4} when $K = \{2, 4\}$ and 2×10^{-4} when $K \in \{5, 8\}$, for CIFAR-100, we use the Adam optimizer with learning rate 2×10^{-4} when $K = \{5, 10, 20, 50\}$. We decrease learning by a factor of 0.25 at the $\{800, 1600\}$ steps for the CIFAR-10 dataset and at the $\{1200, 1800\}$ global steps for the CIFAR-100 dataset.

More Experimental Results. Next, we test our FedSep at different levels of heterogeneity. The results are summarized in Figure 5.5. As shown by the figures, our FedSep is quite robust when clients have highly heterogeneous data.

Part III

Federated Learning Challenges with Hierarchies

Chapter 6: Communication-Efficient Robust Federated Learning with Noisy Labels

6.1 Introduction

In Federated Learning (FL) [92], a set of clients jointly solve a machine learning problem under the coordination of a central server. To protect privacy, clients keep their own data locally and share model parameters periodically with each other. Several challenges of FL are widely studied in the literature, such as user privacy [92, 122, 123], communication cost [124, 125, 126, 127, 128], data heterogeneity [93, 95, 96, 97, 149] *etc.*. However, a key challenge is ignored in the literature: *the label quality of user data*. Data samples are manually annotated, and it is likely that the labels are incorrect. However, existing algorithms of FL *e.g.* FedAvg [92] treat every sample equally; as a result, the learned models overfit the label noise, leading to bad generalization performance. It is challenging to develop an algorithm that is robust to the noise from the labels. Due to privacy concerns, user data are kept locally, so the server cannot verify the quality of the label of the user data. Recently, several intuitive approaches [150, 151, 152] based on the use of a clean validation set have been proposed in the literature. In this paper, we take a step forward and formally formulate the noisy label problem as a bilevel optimization problem; furthermore, we provide two

efficient algorithms that have guaranteed convergence to solve the optimization problem.

The basic idea of our approach is to identify noisy label samples based on its contribution to training. More specifically, we measure the contribution through the Shapley value [153] of each sample. Suppose that we have the training dataset $\mathcal{D}$ and a sample $s \in \mathcal{D}$. Then for any subset $\mathcal{S} \subset \mathcal{D}/\{s\}$, we first train a model with $\mathcal{S}$ only and measure the generalization performance of the learned model, then train over $S \cup \{s\}$ and calculate the generalization performance again. The difference in generalization performance of the two models reflects the quality of the sample label. If a sample has a correct label, the model will have better generalization performance when the sample is included in the training; in contrast, a mislabeled sample harms the generalization performance. Then we define the Shapley value of any sample as the average of the generalization performance difference in all possible subsets $\mathcal{S}$. However, the Shapley value of a sample is NP-hard to compute. As an alternative, we define a weight for each sample and turn the problem into finding weights that lead to optimal generalization performance. With this reformulation, we need to solve a bilevel optimization problem. Bilevel optimization problems [16, 17, 19] involve two levels of problems: an inner problem and an outer problem. Efficient gradient-based alternative update algorithms [29, 33, 154] have recently been proposed to solve non-distributed bilevel problems, but efficient algorithms designed for the FL setting have not yet been shown. In fact, the most challenging step is to evaluate the hypergradient (gradient $w.r.t$ the variable of the outer problem). In FL, hypergradient evaluation involves transferring the Hessian matrix, which leads to high communication cost. It is essential to develop a communication-efficient algorithm to evaluate the hypergradient and solve the Federated Bilevel Optimization problem efficiently.

More specifically, we propose two compression algorithms to reduce the communication cost for the hypergradient estimation: an iterative algorithm and a non-iterative algorithm. In the non-iterative algorithm, we compress the Hessian matrix directly and then solve a small linear equation to get the hypergradient. In the iterative algorithm, we formulate the hypergradient evaluation as solving a quadratic optimization problem and then run an iterative algorithm to solve this quadratic problem. To further save communication, we also compress the gradient of the quadratic objective function. Both the non-iterative and iterative algorithms effectively reduce the communication overhead of hypergradient evaluation. In general, the non-iterative algorithm requires communication cost polynomial to the stable rank of the Hessian matrix, and the iterative algorithm requires $O(log(d))$ (d is the dimension of the model parameters). Finally, we apply the proposed algorithms to solve the noisy label problem on real-world datasets. Our algorithms have shown superior performance compared to various baselines. We highlight the contribution of this paper below:

- We study Federated Learning with noisy labels problems and propose a learning-based data cleaning procedure to identify mislabeled data.
- We formalize the procedure as a Federated Bilevel Optimization problem. Furthermore, we propose two novel efficient algorithms based on compression, *i.e.* the Iterative and Non-iterative algorithms. Both methods reduce the communication cost of the hypergradient evaluation from $O(d^2)$ to be sub-linear of d.
- We show that the proposed algorithms have a convergence rate of $O(\epsilon^{-2})$ and validate their efficacy by identifying mislabeled data in real-world datasets.

Notations. ∇ denotes the full gradient, ∇_x is the partial derivative for variable x, and higher-order derivatives follow similar rules. $||\cdot||$ is ℓ_2-norm for vectors and the spectral norm for matrices. $||\cdot||_F$ represents the Frobenius norm. AB denotes the multiplication of the matrix between the matrix A and B. $\binom{n}{k}$ denotes the binomial coefficient. $[K]$ represents the sequence of integers from 1 to K.

6.2 Related Works

Federated Learning. FL is a promising paradigm for performing machine learning tasks on distributed located data. Compared to traditional distributed learning in the data center, FL poses new challenges such as heterogeneity [96, 113, 115, 116, 155], privacy [92, 122, 123] and communication bottleneck [124, 125, 126, 127, 128]. In addition, another challenge that receives little attention is the noisy data problem. Learning with noisy data, especially noisy labels, has been widely studied in a non-distributed setting [47, 48, 156, 157, 158, 159, 160, 161]. However, since the server cannot see the clients' data and the communication is expensive between the server and clients, algorithms developed in the non-distributed setting can not be applied to the Federated Learning setting. Recently, several works [150, 151, 152] have focused on FL with noisy labels. In [150], authors propose FOCUS: The server defines a credibility score for each client based on the mutual cross-entropy of two losses: the loss of the global model evaluated on the local dataset and the loss of the local model evaluated on a clean validation set. The server then uses this score as the weight of each client during global averaging. In [152], the server first trains a benchmark model, and then the clients use this model to exclude possibly corrupted data

samples.

Gradient Compression. Gradient compression is widely used in FL to reduce communication costs. Existing compressors can be divided into quantization-based [124, 125] and sparsification-based [126, 127]. Quantization compressors give an unbiased estimate of gradients, but have a high variance [128]. In contrast, sparsification methods generate biased gradients, but have high compression rate and good practical performance. The error feedback technique [127] is combined with sparsification compressors to reduce compression bias. Sketch-based compression methods [128, 130] are one type of sparsification compressor. Sketching methods [162] originate from the literature on streaming algorithms, *e.g.* The Count-sketch [129] compressor was proposed to efficiently count heavy hitters in a data stream.

Bilevel Optimization. Bilevel optimization [16] has gained more interest recently due to its application in many machine learning problems such as hyperparameter optimization [4], meta learning [38], neural architecture search [41] *etc.* Various gradient-based methods are proposed to solve the bilevel optimization problem. Based on different approaches to the estimation of hypergradient, these methods are divided into two categories, *i.e.* Approximate Implicit Differentiation (AID) [6, 13, 14, 29, 32, 33, 154] and Iterative Differentiation (ITD) [24, 25, 26, 27]. ITD methods first solve the lower level problem approximately and then calculate the hypergradient with backward (forward) automatic differentiation, while AID methods approximate the exact hypergradient [4, 6, 9, 12]. In [7], authors compare these two categories of methods in terms of their hyperiteration complexity. Finally, there are also works that utilize other strategies such as penalty methods [34], and also other formulations *e.g.* the inner problem has multiple minimizers [35, 36]. A recent work [163]

applied momentum-based acceleration to solve federated bilevel optimization problems.

6.3 Preliminaries

Federated Learning. A general formulation of Federated Learning problems is:

$$\min_{x\in\mathcal{X}} G(x) \coloneqq \frac{1}{N}\sum_{i=1}^{M} N_i g_i(x) \tag{6.1}$$

There are M clients and one server. N_i is the number of samples in the i_{th} client and N is the total number of samples. g_i denotes the objective function on the i_{th} client. To reduce communication cost, a common approach is to perform local sgd; in other words, the client performs multiple update steps with local data, and the model averaging operation occurs every few iterations. A widely used algorithm that uses this approach is FedAvg [92].

Count Sketch. The count-sketch technique [129] was originally proposed to efficiently count heavy-hitters in a data stream. Later, it was applied in gradient compression: It is used to project a vector into a lower-dimensional space, while the large-magnitude elements can still be recovered. To compress a vector $g \in \mathbb{R}^d$, it maintains counters $r \times c$ denoted as S. Furthermore, it generates sign and bucket hashes $\{h_j^s, h_j^b\}_{j=1}^r$. In the compression stage, for each element $g_i \in g$, it performs the operation $S[j, h_j^b(i)] += h_j^s[i] * g_i$ for $j \in [r]$. In the decompression stage, it recovers g_i as $\text{median}(\{h_j^s[i] * S[j, h_j^b(i)]\}_{j=1}^r)$. To recover τ heavy-hitters (elements g_i where $||g_i||^2 \geq \tau||g||^2$) with probability at least $1-\delta$, the count sketch needs $r \times c$ to be $O(\tau^{-1}\log(d/\delta))$. More details of the implementation are provided in [129].

Bilevel Optimization. A bilevel optimization problem has the following form:

$$\min_{x\in\mathcal{X}} h(x) \coloneqq F(x, y_x) \ s.t. \ y_x = \arg\min_{y\in\mathbb{R}^d} G(x, y) \tag{6.2}$$

As shown in Eq. (6.2), a bilevel optimization problem includes two entangled optimization problems: the outer problem $F(x, y_x)$ and the inner problem $G(x, y)$. The outer problem relies on the minimizer y_x of the inner problem. Eq. (6.2) can be solved efficiently through gradient-based algorithms [29, 154]. There are two main categories of methods for hypergradient (the gradient $w.r.t$ the outer variable x) evaluation: Approximate Implicit Differentiation (AID) and Iterative Differentiation (ITD). The ITD is based on automatic differentiation and stores intermediate states generated when we solve the inner problem. ITD methods are not suitable for the Federated Learning setting, where clients are stateless and cannot maintain historical inner states. In contrast, the AID approach is based on an explicit form of the hypergradient, as shown in Proposition 134:

Proposition 134. *(hypergradient) When y_x is uniquely defined and $\nabla^2_{yy} G(x, y_x)$ is invertible, the hypergradient has the following form:*

$$\nabla h(x) = \nabla_x F(x, y_x) - \nabla^2_{xy} G(x, y_x) v^* \tag{6.3}$$

where v^* is the solution of the following linear equation:

$$\nabla^2_{yy} G(x, y_x) v^* = \nabla_y F(x, y_x) \tag{6.4}$$

The proposition 134 is based on the chain rule and the implicit function theorem. The proof of Proposition 134 can be found in the bilevel optimization literature, such as [6].

6.4 Federated Learning with Noisy Labels

We consider the Federated Learning setting as shown in Eq. (6.1), *i.e.* a server and a set of clients. For ease of discussion, we assume that the total number of clients is M and that each client has a private data set $\mathcal{D}_i = \{s_j^i, j \in [N_i]\}$, $i \in [M]$ where $|\mathcal{D}_i| = N_i$. $\mathcal{D}$ denotes the union of all client datasets: $\mathcal{D} = \bigcup_{i=1}^{M} \mathcal{D}_i$. The total number of samples $|\mathcal{D}| = N$ and $N = \sum_{i=1}^{M} N_i$ (for simplicity, we assume that there is no overlap between the client data sets).

In Federated Learning, the local dataset $\mathcal{D}_i$ is not shared with other clients or the server; this protects the user privacy, but also makes the server difficult to verify the quality of data samples. A data sample can be corrupted in various ways; we focus on the noisy label issue. Current Federated Learning models are very sensitive to the label noise in client datasets. Take the widely used FedAvg [92] as an example; the server simply performs a weighted average on the client models in the global averaging step. As a result, if one client model is affected by mislabeled data, the server model will also be affected. To eliminate the effect of these corrupted data samples on training, we can calculate the contribution of each sample. Based on the contribution, we remove samples that have little or even negative contributions. More specifically, we define the following metric of sample

contribution based on the idea of Shapley value [153]:

$$\phi_j^i = \frac{1}{N} \sum_{\mathcal{S} \subset \mathcal{D}/\{s_j^i\}} \binom{N-1}{|\mathcal{S}|}^{-1} \left(\Phi\left(\mathcal{A}\left(\mathcal{S} \cup \{s_j^i\}\right)\right) - \Phi\left(\mathcal{A}\left(\mathcal{S}\right)\right) \right) \tag{6.5}$$

where $\mathcal{A}$ is a randomized algorithm (*e.g.* FedAvg) that takes the dataset $\mathcal{S}$ as input and outputs a model. Φ is a metric of model gain, *e.g.* negative population loss of the learned model, or negative empirical loss of the learned model in a validation set. In Eq. (6.5), for each $S \subset \mathcal{D}/\{s_j^i\}$, we calculate the marginal gain when s_j^i is added to the training and then average over all such subsets $\mathcal{S}$. It is straightforward to find corrupted data if we can compute ϕ_j^i, however, the evaluation of ϕ_j^i is NP-hard. As an alternative, we define the weight $\lambda_j^i \in [0, 1]$ for each sample and λ_j^i should be positively correlated with the sample contribution ϕ_j^i: large λ represents a high contribution and small λ means little contribution. In fact, finding sample weights that reflect the contribution of a data sample can be formulated as solving the following optimization problem:

$$\max_{\lambda \in \Lambda} \Phi(\mathcal{A}(\mathcal{D}; \lambda)) \tag{6.6}$$

The above optimization problem can be interpreted as follows: The sample weights should be assigned so that the gain of the model Φ is maximized. It is then straightforward to see that only samples that have large contributions will be assigned with large weights. Next, we consider an instantiation of Eq. (6.6). Suppose that we choose Φ as the negative empirical loss over a validation set D_{val} at the server and $\mathcal{A}$ fits a model parameterized by

Algorithm 12 Communication-Efficient Federated Bilevel Optimization (**CommFedBiO**)

Input: Learning rate η, γ, initial state (x_0, y_0), number of sampled clients S
for $k = 0$ **to** $K - 1$ **do**
  Sample S clients and broadcast current model state (x_k, y_k);
  for $m = 1$ to S clients in parallel **do**
    Set $y_0^m = y_k$
    for $t = 1$ to T **do**
      $y_{t+1}^m = y_t^m - \gamma \nabla g_m(x_k, y_t^m)$
    end for
  end for
  $y_{k+1} = y_k + \frac{1}{\sum_{m=1}^S N_m} \sum_{m=1}^S N_m (y_T^m - y_k)$
  // Two ways to estimate $\hat{\nabla} h(x_k)$
  Case 1: $\hat{\nabla} h(x_k) = \text{Iterative-approx}(x_k, y_{k+1})$
  Case 2: $\hat{\nabla} h(x_k) = \text{Non-iterative-approx}(x_k, y_{k+1})$
  $x_{k+1} = x_k - \eta \hat{\nabla} h(x_k)$
end for

ω over the data, then Eq. (6.6) can be written as:

$$\min_{\lambda \in \Lambda} \ell(\omega_\lambda; \mathcal{D}_{val}) \ s.t. \ \omega_\lambda = \arg\min_{\omega \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \lambda_j^i \ell(\omega; s_j^i) \tag{6.7}$$

where ℓ is the loss function *e.g.* the cross entropy loss. Eq. (6.7) involves two entangled optimization problems: an outer problem and an inner problem, and ω_λ is the minimizer of the inner problem. This type of optimization problem is known as Bilevel Optimization Problems [16] as we introduce in the preliminary section. Following a similar notation as in Eq. (6.2), we write Eq. (6.7) in a general form:

$$\min_{x \in \mathcal{X}} h(x) \coloneqq F(x, y_x) \text{ s.t. } y_x = \arg\min_{y \in \mathbb{R}^d} G(x, y) \coloneqq \frac{1}{N} \sum_{i=1}^{M} N_i g_i(x, y) \tag{6.8}$$

Compared to Eq. (6.7), we set λ as x, ω as y; $\ell(\omega_\lambda; \mathcal{D}_{val})$ as $F(x, y_x)$, and $1/N_i \sum_{j=1}^{N_i} \lambda_j^i \ell(\omega; s_j^i)$ as $g_i(x, y)$. In the remainder of this section, our discussion will

be based on the general formulation (6.8). We propose the algorithm **Comm-FedBiO** to solve Eq. (6.8) (Algorithm 12). Algorithm 12 follows the idea of alternative update of inner and outer variables in non-distributed bilevel optimization [12, 33, 154], however, it has two key innovations which are our contributions. First, since the inner problem of Eq. (6.8) is a federated optimization problem, we perform local sgd steps to save the communication. Next, we consider the communication constraints of federated learning in the hypergradient estimation. More specifically, we propose two communication-efficient hypergradient estimators, *i.e.* the subroutine *Non-iterative-approx* (line 11) and *Iterative-approx* (line 12).

To see the high communication cost caused by hypergradient evaluation. We first write the hypergradient based on Proposition 134,:

$$\nabla h(x) = \nabla_x F(x, y_x) - \nabla^2_{xy} G(x, y_x) v^*, \; v^* = \left(\sum_{i=1}^{M} \frac{N_i}{N} \nabla^2_{yy} g_i(x, y) \right)^{-1} \nabla_y F(x, y_x) \tag{6.9}$$

where we use the explicit form of v^* and replace $G(x, y)$ with the federated form in Eq. (6.8). Eq. (6.9) includes two steps: calculating v^* and evaluating $\nabla h(x)$ based on v^*. For the second step, clients transfer $\nabla_{xy} g_i(x, y) v^*$ with communication cost $O(l)$ (the dimension of the outer variable x), we assume $l < d$ (d is the dimension of y). This is reasonable in our noisy label application: l is equal to the number of samples at a client and is very small in Federated Learning setting, while d is the weight dimension, which can be very large. Therefore, we focus on the first step when considering the communication cost. The inverse of the Hessian matrix in Eq. (6.9) can be approximated with various techniques such as the Neumann series expansion [6] or conjugate gradient descent [12]. However, clients must

first exchange the Hessian matrix $\nabla^2_{yy} g_i(x, y)$. This leads to a communication cost on the order of $O(d^2)$. In fact, it is not necessary to transfer the full Hessian matrix, and we can reduce the cost through compression. More specifically, we can exploit the sparse structure of the related properties; *e.g.* The Hessian matrix has only a few dominant singular values in practice. In Sections 4.1 and 4.2, we propose two communication-efficient estimators of hypergradient $\nabla h(x_k)$ based on this idea. In general, our estimators can be evaluated with the communication cost sublinear to the parameter dimension d.

6.4.1 hypergradient Estimation with iterative algorithm

In this section, we introduce an iterative hypergradient estimator. Instead of performing the expensive matrix inversion as in Eq. (6.9), we solve the following quadratic optimization problem:

$$\min_v \ q(v) \coloneqq \frac{1}{2} v^T \nabla^2_{yy} G(x, y_x) v - v^T \nabla_y F(x, y_x) \tag{6.10}$$

The equivalence is observed by noticing that:

$$\nabla q(v) = \nabla^2_{yy} G(x, y_x) v - \nabla_y F(x, y_x)$$

If $q(v)$ is strongly convex ($\nabla^2_{yy} G(x, y_x)$ is positive definite), the unique minimizer of the quadratic function $q(v)$ is exactly v^* as shown in Eq. (6.9). Eq. (6.10) is a simple positive definite quadratic optimization problem and can be solved with various iterative gradient-based algorithms. To further reduce communication cost, we compress the gradient $\nabla q(v)$.

Algorithm 13 Iterative Approximation of hypergradient (**Iterative-approx**)

Input: State (x, y), initial value v_0, learning rate α, number of sampled clients S
The server evaluates $\nabla_y F(x, y)$ and samples S clients uniformly and broadcasts state (x, y) to each client;
for $i = 0$ to $I - 1$ **do**
  for $m = 0$ to S in parallel **do**
    Each client makes Hessian-vector product queries to compute $\nabla^2_{yy} g_m v^i$, then send its sketch $S^i_{g_m}$ to the server;
  end for
  Server: $S^i_G = \frac{1}{\sum_{m=1}^{S} N_m} \sum_{m=1}^{S} N_m * S^i_{g_m}$
  Server: $\Delta = U(\alpha S^i_G + S(e^i))$
  Server: $v^{i+1} = v^i - (\Delta - \alpha \nabla_y F(x, y))$
  Server: $S(e^{i+1}) = \alpha S^i_G + S(e^i) - S(\Delta)$
end for
Output: $\hat{\nabla} h(x) = \nabla_x F - \nabla^2_{xy} G v^I$

More specifically, $\nabla q(v)$ can be expressed as follows in terms of g_i:

$$\nabla q(v) = \frac{1}{N} \sum_{i=1}^{M} N_i \nabla^2_{yy} g_i(x, y_x) v - \nabla_y F(x, y_x) \tag{6.11}$$

Therefore, clients must exchange the Hessian vector product to evaluate $\nabla q(v)$. This operation has a communication cost $O(d)$. This cost is considerable when we evaluate $\nabla q(v)$ multiple times to optimize Eq.(6.10). Therefore, we exploit compression to further reduce communication cost; *i.e.* clients only communicate the compressed Hessian vector product. Various compressors can be used for compression. In our paper, we consider the local Topk compressor and the Count Sketch compressor [129] in our paper. The local Top-k compressor is simple to implement, but it cannot recover the global Top-k coordinates, while the count-sketch is more complicated to implement, but it can recover the global Top-k coordinates under certain conditions. In general, gradient compression includes two phases: compression at clients and decompression at the server. In the first phase, clients compress

the gradient to a lower dimension with the compressor $S(\cdot)$, then transfer the compressed gradient to the server; In the second phase, the server aggregates the compressed gradients received from clients and decompresses them to recover an approximation of the original gradients. We denote the decompression operator as $U(\cdot)$. Then the update step of a gradient descent method with compression is as follows:

$$\begin{aligned} v^{i+1} &= v^i - C(\alpha \nabla q(v^i) + e^i), \\ e^{i+1} &= \alpha \nabla q(v^i) + e^i - C(\alpha \nabla q(v^i) + e^i) \end{aligned} \tag{6.12}$$

where $C(\cdot) \coloneqq U(S(\cdot))$ and α is the learning rate. Notice that we add an error accumulation term e^i. As shown by the update rule of e^i, it accumulates information that cannot be transferred due to compression and reintroduces information later in the iteration, this type of error feedback trick compensates for the compression error and is crucial for convergence. The update rule in Eq. (6.12) has communication cost sub-linear *w.r.t* parameter dimension d with either the Top-k compressor or the Count-sketch compressor. Furthermore, the iteration complexity of iterative algorithms is independent of the problem dimension, the overall communication cost of evaluating v^* is still sublinear *w.r.t* the dimension d. This is a great reduction compared to the $O(d^2)$ complexity when we transfer the Hessian directly as in Eq. (6.9). We term this hypergradient approximation approach the iterative algorithm, and the pseudocode is shown in Algorithm 13. Note that the server can evaluate $\nabla_y F(x, y)$, so we do not need to compress it, and we also omit the step of getting $\nabla^2_{xy} G v^I$ from the clients.

6.4.2 hypergradient Estimation with Non-iterative algorithm

In this section, we propose an efficient algorithm so that we can estimate v^* by solving the linear equation Eq. (6.4) directly. However, instead of transferring $\nabla^2_{yy} g_i(x, y_x)$, we transfer their sketch. More precisely, we solve the following linear equation:

$$S_2 \nabla^2_{yy} G(x, y_x) S_1^T \omega = S_2 \nabla_y F(x, y_x) \tag{6.13}$$

where $\hat{\omega} \in \mathbb{R}^{r_1}$ denotes the solution of Eq. (6.13). $S_1 \in \mathbb{R}^{r_1 \times d}$ and $S_2 \in \mathbb{R}^{r_2 \times d}$ are two random matrices. Then an approximation of the hypergradient $\nabla h(x)$ is:

$$\hat{\nabla} h(x) = \nabla_x F(x, y_x) - \nabla^2_{xy} G(x, y_x) S_1^T \hat{\omega} \tag{6.14}$$

To solve Eq. (6.13), clients first transfer $S_2 \nabla^2_{yy} g_i(x, y_x) S_1^T$ to the server with communication cost $O(r_1 r_2)$, then the server solves the linear system (6.13) locally. The server then transfers $\hat{\omega}$ to the clients and the clients transfer $\nabla_{xy} g_i(x, y_x) S_1^T \hat{\omega}$ back to the server, the server evaluates Eq. (6.14) to get $\hat{\nabla} h(x)$. The total communication cost is $O(r_1 r_2)$ (we assume that the dimension of the outer variable x is small).

We require S_1 and S_2 to have the following two properties: the approximation error $||\hat{\nabla} h(x) - \nabla h(x)||$ is small and the communication cost is much lower than $O(d^2)$, *i.e.* $r_1 r_2 \ll d^2$. We choose S_1 and S_2 as the following sketch matrices:

Definition 135. *A distribution $\mathcal{D}$ on the matrices $S \in \mathbb{R}^{r \times n}$ is said to generate a (ϵ, δ)-*

sketch matrix for a pair of matrices A, B with n rows if:

$$\Pr_{S\sim\mathcal{D}}\left[||A^TS^TSB - A^TB|| > \epsilon||A||_F||B||_F\right] \leq \delta$$

Corollary 136. *An $(\epsilon/l, \delta)$ sketch matrix S is a subspace embedding matrix for the column space of $A \in R^{n\times l}$.* i.e. *for all $x \in \mathbb{R}^l$* w.p. *at least $1-\delta$:*

$$||SAx||_2^2 \in [(1-\epsilon)||Ax||_2^2, (1+\epsilon)||Ax||_2^2]$$

Corollary 137. *For any $\epsilon, \delta \in (0, 1/2)$, let $S \in \mathbb{R}^{r\times n}$ be a random matrix with $r > 18/(\epsilon^2\delta)$. Furthermore, suppose that $\sigma \in \mathbb{R}^n$ is a random sequence where $\sigma(i)$ is randomly chosen from $\{-1, 1\}$ and $h \in \mathbb{R}^n$ is another random sequence where $h(i)$ is randomly chosen from $[r]$. Suppose that we set $S[h(i), i] = \sigma(i)$, for $i \in [n]$ and 0 for other elements; then S is a (ϵ, δ) sketch matrix.*

Approximately, the sketch matrices S are 'invariant' over matrix multiplication ($\langle SA, SB\rangle \approx AB$). An important property of a (ϵ, δ)-sketch matrix is the subspace embedding property in Corollary 136: The norm of the vectors in the column space of A is kept roughly after being projected by S. Many distributions generate sketch matrices, such as *sparse embedding matrice* [164]. We show one way to generate a sparse embedding matrix in Corollary 137. This corollary shows that we need to choose $O(\epsilon^{-2})$ number of rows for a sparse embedding matrix to be a (ϵ, δ) matrix. Finally, since we directly solve a (sketched) linear equation without using any iterative optimization algorithms, we term this hypergradient estimator as a non-iterative approximation algorithm. The pseudocode

Algorithm 14 Non-iterative approximation of hypergradient (**Non-iterative-approx**)

1: **Input:** State (x, y), random seeds τ_1, τ_2, number of rows r_1, r_2, number of sampled clients S
2: **Server:** Sample S clients uniformly and broadcast model state (x, y) to each sampled client
3: **for** $m = 1$ to S in parallel **do**
4: Each client generates S_1, S_2 with random seeds τ_1, τ_2, compute $S_2\nabla^2_{yy}g_jS_1^T$ and $\nabla^2_{xy}g_jS_1^T$ with Hessian-vector product queries
5: **end for**
6: **Server:** Collect and average sketches from clients to get $S_2\nabla^2_{yy}GS_1^T$ and $\nabla^2_{xy}GS_1^T$ and solves Eq. (6.13) with linear regression to get $\hat{\omega}$
7: **Output:** $\hat{\nabla}h(x) = \nabla_x F - \nabla^2_{xy}GS_1^T\hat{\omega}$

summarizing this method is shown in Algorithm 14. We omit the subscript of iterates when it is clear from the context. Note that the server sends the random seed to ensure that all clients generate the same sketch matrices S_1 and S_2.

6.5 Convergence Analysis

In this section, we analyze the convergence property of Algorithm 12. We first state some mild assumptions needed in our analysis, then we analyze the approximation error of the two hypergradient estimation algorithms, *i.e.* the iterative algorithm and the non-iterative algorithm. Finally, we provide the convergence guarantee of Algorithm 12.

6.5.1 Some Mild Assumptions

We first state some assumptions about the outer and inner functions as follows:

Assumption 138. *The function F and G has the following properties:*

a) $F(x, y)$ is possibly non-convex, $\nabla_x F(x, y)$ and $\nabla_y F(x, y)$ are Lipschitz continuous with constant L_F

b) $\|\nabla_x F(x,y)\|$ *and* $\|\nabla_y F(x,y)\|$ *are upper bounded by some constant* C_F

c) $G(x,y)$ *is continuously twice differentiable, and* μ_G*-strongly convex* w.r.t y *for any given* x

d) $\nabla_y G(x,y)$ *is Lipschitz continuous with constant* L_G

e) $\|\nabla^2_{xy} G(x,y)\|$ *is upper bounded by some constant* $C_{G_{xy}}$

Assumption 139. $\nabla^2_{xy} G(x,y)$ *and* $\nabla^2_{yy} G(x,y)$ *are Lipschitz continuous with constants* $L_{G_{xy}}$ *and* $L_{G_{yy}}$*, respectively.*

In Assumptions 138 and 139, we make assumptions about the function G, it is also possible to make stronger assumptions about the local functions g_i. Furthermore, these assumptions are used in the bilevel optimization literature [6, 12], especially, we require higher-order smoothness in Assumption 139 as bilevel optimization is involved with the second-order information. The next two assumptions are needed when we analyze the approximation property of the two hypergradient estimation algorithms:

Assumption 140. *For a constant* $0 < \tau < 1$ *and a vector* $g \in \mathbb{R}^d$*. If* $\exists$ i*, such that* $(g_i)^2 \geq \tau ||g||^2$*, then* g *has* τ*-heavy hitters.*

Assumption 141. *The stable rank of* $\nabla^2_{yy} G(x, y_x)$ *is bounded by* r_s*,* i.e. $\sum_{i=1}^{d} \sigma_i^2 \leq r_s \sigma_{max}^2$*, where* ($\sigma_{max}$) σ_i *denotes the (max) singular values of the Hessian matrix.*

The heavy-hitter assumption 140 is commonly used in the literature to show the convergence of gradient compression algorithms. To bound the approximation error of our iterative hypergradient estimation error, we assume $\nabla q(v)$ defined in Eq. (6.11) to satisfy

this assumption. Assumption 141 requires the Hessian matrix to have several dominant singular values, which describes the sparsity of the Hessian matrix. We assume Assumption 141 holds when we analyze the approximation error of the non-iterative hypergradient estimation algorithm.

6.5.2 Approximation Error of the iterative algorithm

In this section, we show the approximation error of the iterative algorithm. Suppose that we choose count-sketch as the compressor, we have the following theorem:

Theorem 142. *Assume Assumptions 138 and 140 hold. In Algorithm 13, set the learning rate* $\alpha = \frac{8}{\mu_G(i+a)}$ *with* $a > \max\left(1, \frac{2-\tau}{\tau}(\sqrt{\frac{2}{2-\tau}}+1)\right)$ *as a shift constant. If the compressed gradient has dimension* $O(\frac{log(dI/\delta)}{\tau})$*, then with probability* $1-\delta$ *we have:*

$$E[||v^I - v^*||^2] \leq \frac{C_1}{I^3} + \frac{C_2}{I^2} + \frac{C_3(I+2a)}{I^2}$$

where C_1*,* C_2*, and* C_3 *are constants.*

Remark 143. *The proof is included in Appendix A. As shown by Theorem 142, the approximation error of Algorithm 13 is of the order of* $O(1/I)$*, and the constants encompass compression errors. Finally, the communication cost is of the order of* $O(log(d))$*, which is sublinear* w.r.t *of dimension* d*.*

6.5.3 Approximation Error of the Non-iterative algorithm

In this section, we show the approximation error of the non-iterative algorithm. More precisely, we have Theorem 144:

Theorem 144. *For any given $\epsilon, \delta \in (0, 1/2)$, if $S_1 \in \mathbb{R}^{r_1 \times d}$ is a $(\lambda_1\epsilon, \delta/2)$ sketch matrix and $S_2 \in \mathbb{R}^{r_2 \times d}$ is a $(\lambda_2\epsilon, \delta/2)$ sketch matrix. Under Assumptions 138 and 141, with probability at least $1 - \delta$, we have the following:*

$$||\hat{\nabla} h(x) - \nabla h(x)|| \leq \epsilon ||v^*||$$

where $\lambda_1 = \frac{5\mu_G}{7\sqrt{r_s} C_{G_{xy}} L_G}, \lambda_2 = \frac{1}{3(r_1+1)}$ are constants.

Proof Sketch: The main step is to bound $||S_1^T \hat{\omega} - v^*||$, then the conclusion follows from the definition of the hypergradient. To bound $||S_1^T \hat{\omega} - v^*||$, we use the approximation matrix multiplication property of S_1 and the subspace embedding property of S_2 to have: $||S_1^T \hat{\omega} - v^*||_2 \leq C||v^*||_F ||\nabla^2_{yy} G(x, y_x)||_F$, where C is some constant. The last step is to use the stable rank and the smoothness assumption to bound $||\nabla^2_{yy} G(x, y_x)||_F$. The full proof is included in Appendix B.

Remark 145. *Based on Corollary 137, we have an (ϵ, δ) sketch matrix that has $r = O(\epsilon^{-2})$ rows. Combining with Theorem 144, we have $r_1 = O(r_s)$ and $r_2 = O(r_s^2)$. So, to reach the approximation error ϵ, the number of rows of the sketch matrices is $O(r_s^2)$. This shows that the stable rank (the number of dominant singular values) correlates with the number of dimensions to be maintained after compression.*

6.5.4 Convergence of the Comm-FedBiO algorithm

In this section, we study the convergence property of the proposed communication efficient federated bilevel optimization (**Comm-FedBiO**) algorithm. First, $h(x)$ is smooth based on Assumptions 138 and 139, as stated in the following proposition:

Proposition 146. *Under Assumption 138 and 139, $\nabla h(x)$ is Lipschitz continuous with constant L_h,* i.e.

$$||\nabla h(x_1) - \nabla h(x_2)|| \leq L_h ||x_1 - x_2||$$

where $\nabla h(x)$ is the hypergradient and is defined in Proposition 134.

The proof of Proposition 146 can be found in the Lemma 2.2 of [6]. We are ready to prove the convergence of Algorithm 12 in the following theorem. In this simplified version, we ignore the exact constants. A full version of the theorem is included in Appendix C.

Theorem 147. *Under Assumption 138 and 139, if we choose the learning rate $\eta = \frac{1}{2L_h\sqrt{K+1}}$ in Algorithm 12,*

a) Suppose that $\{x_k\}_{k\geq 0}$ is generated from the iterative Algorithm 13. Under Assumption 140, for $I = O(\sqrt{K})$, we have the following:

$$E[||\nabla h(x_k)||^2] \leq \frac{C_1}{K^{3/2}} + \frac{C_2}{K} + \frac{C_3}{\sqrt{K}}$$

where C_1, C_2, C_3 are some constants

b) Suppose $\{x_k\}_{k\geq 0}$ are generated from the non-iterative Algorithm 14. Under Assumption 141, for $\epsilon = O(K^{-1/4})$, it holds: $E[||\nabla h(x_k)||^2] \leq \frac{C}{\sqrt{K}}$, where C is some constant.

Firstly, by the smoothness of $h(x)$, we can upper-bound $h(x_{k+1})$ as:

$$h(x_{k+1}) \leq h(x_k) - \eta(\frac{1}{2} - \eta L_h)||\nabla h(x_k)||^2 + \eta(\frac{1}{2} + \eta L_h)||\hat{\nabla} h(x_k) - \nabla h(x_k)||^2$$

Next, we need to bound the error in the third term, where we can utilize the bound provided in Theorem 142 and Theorem 144. Finally, we find a suitable averaging scheme to obtain the bound for $E[||\nabla h(x_k)||^2]$.

Remark 148. *The convergence rate for the nonconvex-strongly-convex bilevel problem without using variance reduction technique is $O(1/\sqrt{K})$ [6], thus both estimation algorithms achieve the same convergence rate as in the non-distributed setting. For the non-iterative algorithm, we need to scale $\epsilon = O(K^{-1/4})$, while the iterative algorithm instead scales the number of iterations as $O(\sqrt{K})$ at each hyper-iteration. Comparing these two methods: The iterative algorithm has to perform multiple rounds of communication, but it distributes the computation burden over multiple communication clients (multiple clients by sampling different clients at each step) and requires one Hessian vector product per round. But if the communication is very expensive, we could instead use the non-iterative algorithm, which requires one round of communication.*

6.6 Numerical Experiments

In this section, we empirically validate our **Comm-FedBiO** algorithm. We consider three real-world datasets: MNIST [85], CIFAR-10 [142] and FEMNIST [165]. For MNIST and CIFAR-10. We create 10 clients, for each client, we randomly sample 500 images from

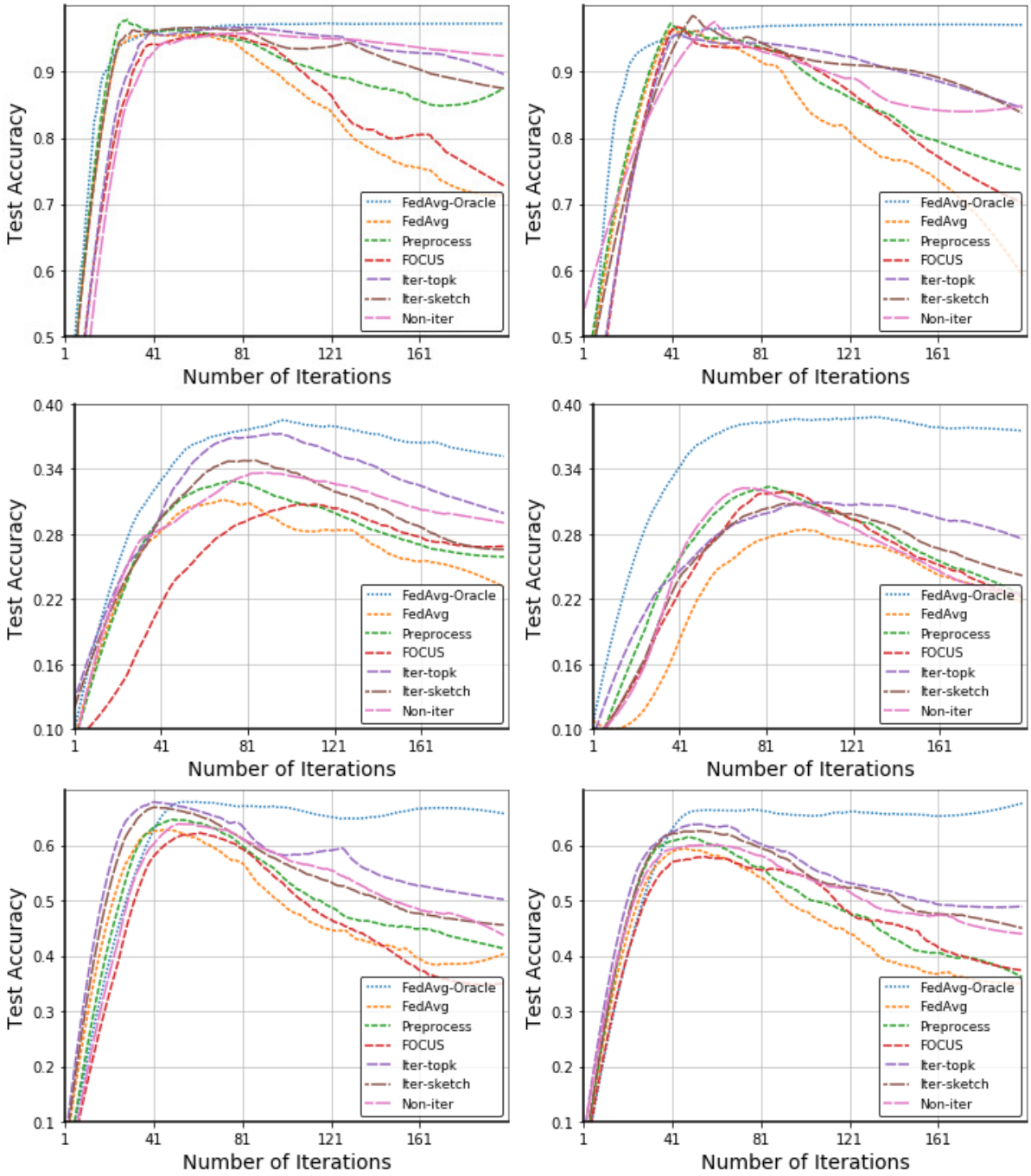


Figure 6.1: Test accuracy plots for our Comm-FedBiO (three variants: Iter-topK, Iter-sketch and Non-iter) and other baselines. The plots show the results for the MNIST dataset, the CIFAR-10 dataset, and the FEMNIST dataset from top to bottom. The plots in the left column show the i.i.d. case, and plots in the right column show the non-i.i.d. case. The compression rate of our algorithms is $20\times$ in terms of the parameter dimension d.

the original training set. For the server, we sample 500 images from the training set to construct a validation set. For FEMNIST, the entire dataset has 3,500 users and 805,263 images. We randomly select 350 users and distribute them over 10 clients. On the server side, we randomly select another 5 users to construct the validation set. Next, for label noise, we randomly perturb the labels of a portion of the samples in each client, and the portion is denoted ρ. We consider two settings: i.i.d. and non-i.i.d. setting. For the i.i.d. setting, all clients are perturbed with the same ratio ρ and we set $\rho = 0.4$ in experiments, while for the non-i.i.d. setting, each client is perturbed with a random ratio from the range of $[0.2, 0.9]$. The code is written with Pytorch, and the Federated Learning environment is simulated via Pytorch.Distributed Package. We used servers with AMD EPYC 7763 64-core CPU and 8 NVIDIA V100 GPUs to run our experiments.

For our algorithms, we consider three variants of *Comm-FedBiO* based on using different hypergradient approximation estimators: the non-iterative approximation method, the iterative approximation method with local Top-k and the iterative approximation method with Count-sketch compressor. We use Non-iter, Iter-topK and Iter-sketch as their short names. Furthermore, we also consider some baseline methods: a baseline that directly performs FedAvg [92] on the noisy dataset, an oracle method where we assume that clients know the index of clean samples (we denote this method as *FedAvg-Oracle*), the *FOCUS* [150] method which reweights clients based on a 'credibility score' and the *Preprocess* method [152] which uses a benchmark model to remove possibly mislabeled data before training.

We fit a model with 4 convolutional layers with 64 3×3 filters for each layer. The total number of parameters is about 10^5. We also use L_2 regularization with coefficient 10^{-3} to

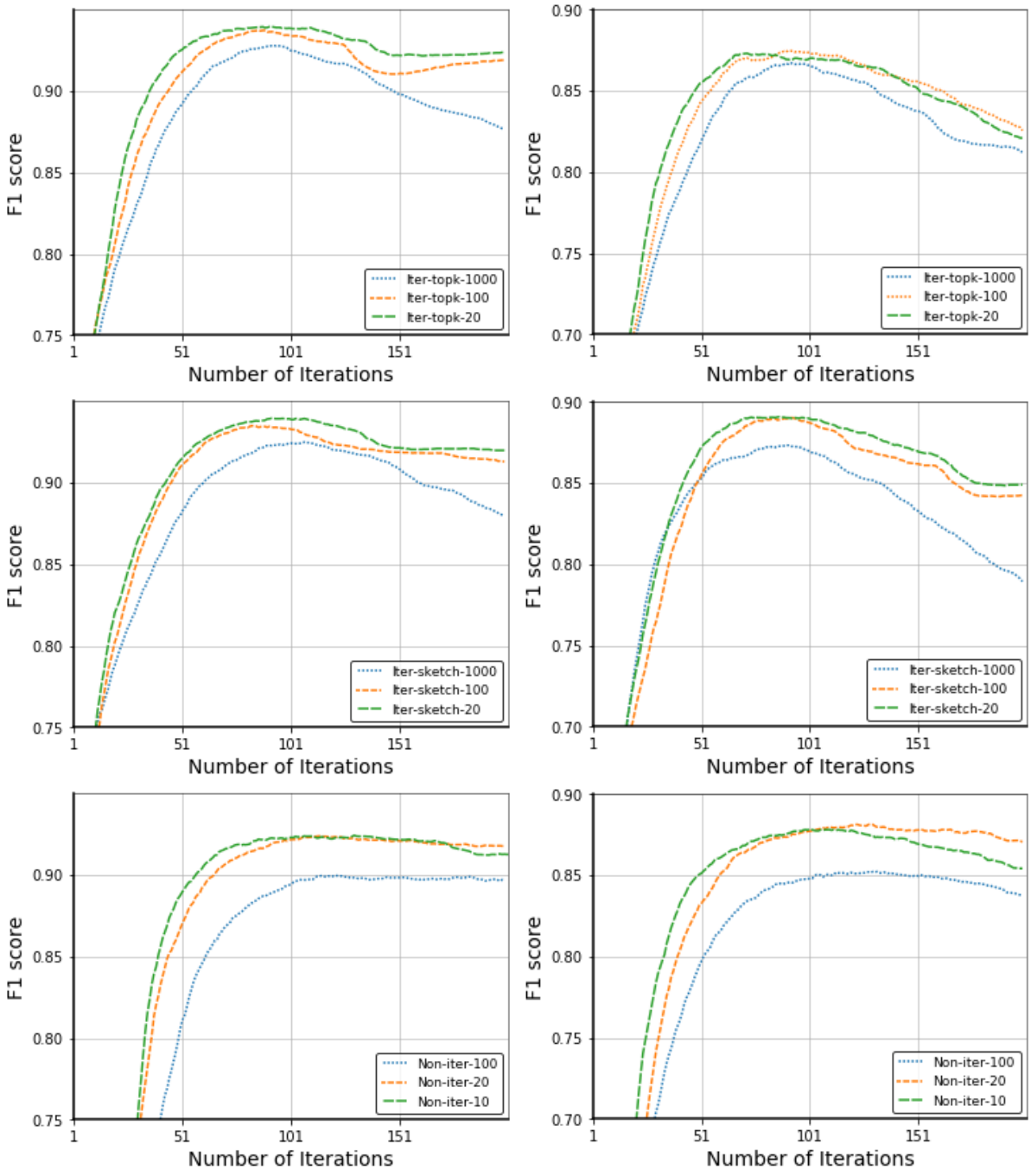


Figure 6.2: F1 score at different compression rates for the MNIST data set. The plots show the results for Iter-topK, Iter-sketch, and Non-iter from top to bottom. Plots in Left show the i.i.d case and in right show the non-i.i.d case.

satisfy the strong convexity condition. Regarding hyper parameters, for three variants of our *Comm-FedBiO*, we set hyper-learning rates (learning rate for sample weights) as 0.1, the learning rate as 0.01, and the local iterations T as 5. We choose a minibatch of size 256 for MNIST and FEMNIST datasets and 32 for CIFAR10 datasets. For *FedAvg* and *FedAvg-Oracle*, we choose the learning rate, local iterations, and mini-batch size the same as in our *Comm-FedBiO*. For *FOCUS* [150], we tune its parameter α to report the best results, for *Preprocess* [152], we tune its parameter filtering threshold and report the best results.

We summarize the results in Figure 6.1. Due to the existence of noisy labels, *FedAvg* overfits the noisy training data quickly and the test accuracy decreases rapidly. On the contrary, our algorithm mitigates the effects of noisy labels and gets a much higher test accuracy than *FedAvg*, especially for the MNIST dataset, our algorithms get a test accuracy similar to that of the oracle model. Compared to *FedAvg*, our *Comm-FedBiO* performs the additional hypergradient evaluation operation at each global iteration (lines 11 - 13 in Algorithm 12). However, the additional communication overhead is negligible. In Figure 6.1, we need the communication cost $O(d/20)$, where d is the parameter dimension. Our algorithms are robust in labeling noise with almost no extra communication overhead. Finally, our algorithms also outperform the baselines *FOCUS* and *Preprocess*. The *FOCUS* method adjusts weights at the client level, so its performance is not good when all clients have a portion of mislabeled data, As for the *Preprocess* method, the benchmark model (trained over a small validation set) can screen out some mislabeled data, but its performance is sensitive to the benchmark model's performance and a 'filtering threshold' hyperparameter.

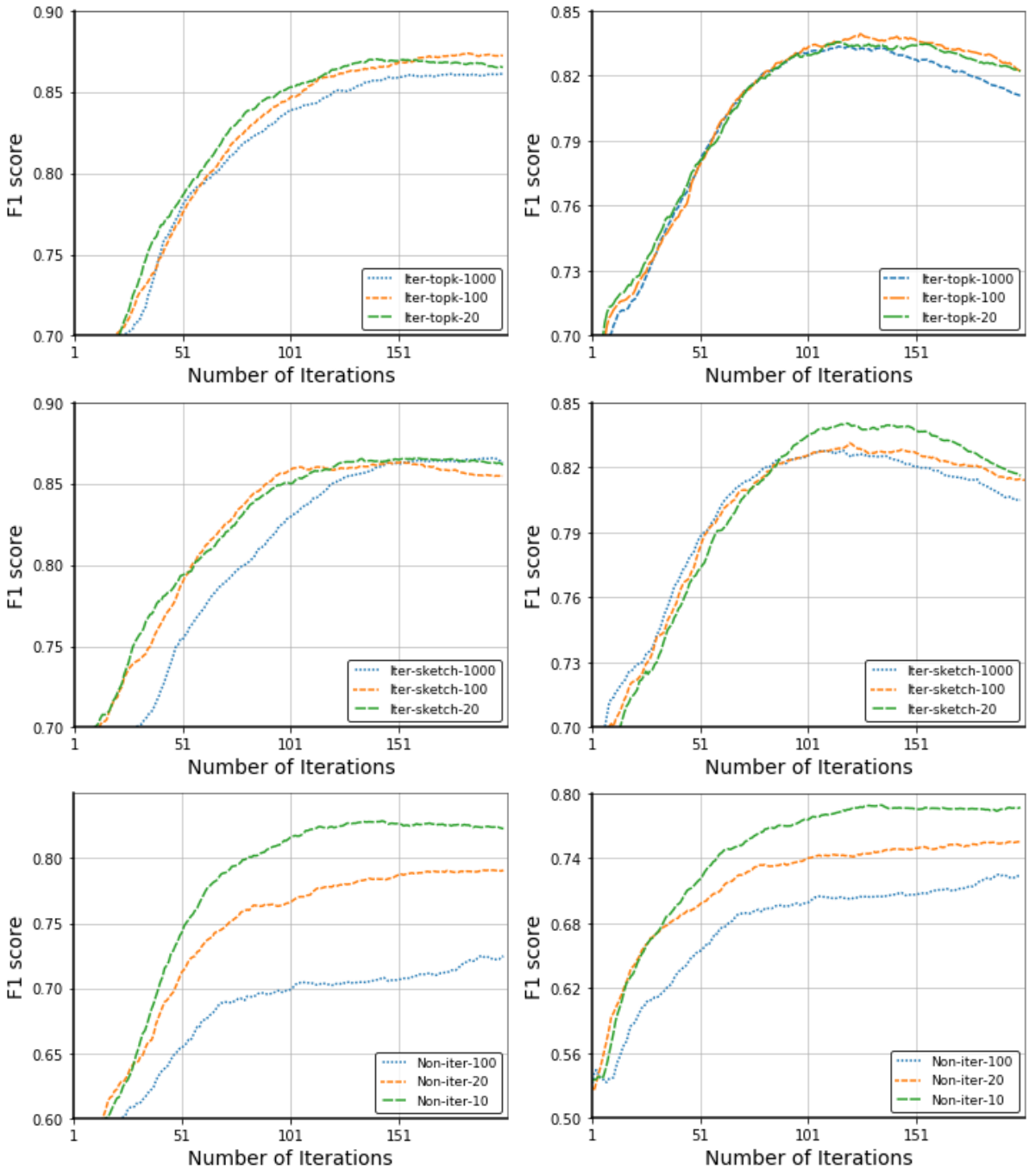


Figure 6.3: F1 score at different compression rates for the FEMNIST data set. The plots show results for Iter-topK, Iter-sketch and Non-iter from top to bottom. Plots on the left show the i.i.d. case, and in the right show the non-i.i.d case.

Next, we verify that our algorithms are robust at different compression rates. The results are summarized in Figures 6.2 and 6.3. We use the $F1$ score to measure the efficacy of our algorithms in identifying mislabeled samples. We use 0.5 as the threshold for mislabeled data: for all samples with weights smaller than 0.5, we assume that they are mislabeled. Then the F1 score is computed between the ground-truth mislabeled samples and predicted mislabeled samples of our algorithms. In Figures 6.2 and 6.3, we show the results of i.i.d. and non-i.i.d. cases for the MNIST and FEMNIST datasets. The iterative algorithms (Iter-topK and Iter-sketch) achieve higher compression rates than the Non-iter method. For iterative algorithms, performance decreases at the compression rate 1000×, while the Non-iter method works well at around 10× to 100×. A major reason for this phenomenon is that the gradient $\nabla q(v)$ is highly sparse in experiments, whereas the Hessian matrix itself is much denser.

6.7 Proof of Theorems

6.7.1 Proof for iterative algorithm

In the iterative algorithm, we optimize Eq. (6.10). The full version of Theorem 142 in the main text is stated as follows:

Theorem 149. *(Theorem 142) Under Assumption 138, 140 and* $||v^i|| \leq D_v$, $\alpha = \frac{8}{\mu_G(i+a)}$, *with* $a > \max(1, \frac{2-\tau}{\tau}(\sqrt{\frac{2}{2-\tau}}+1))$ *being some shift constant. Then, if we use the count-sketch size* $O(log(dI/\delta)/\tau)$, *with probability* $1-\delta$, *we have the following:*

$$E[||v^I - v^*||^2] \leq \frac{C_1}{I^3} + \frac{C_2}{I^2} + \frac{C_3}{I^2}$$

where $G_q^2 = 2L_G^2 D_v^2 + 2C_F^2$, $C_1 = 3a^3 D_v^2/4$, $C_2 = (384(2L_G + \mu_G)G_q^2)/(\mu_G^3 \tau(1-(1-\frac{\tau}{2})(1+\frac{1}{a})^2))$, $C_3 = 12(I+2a)G_q^2/\mu_G^2$

Proof. We first show that $E||\nabla q(v^i)||^2$ is upper bounded: $||\nabla q(v^i)||^2 = ||\nabla^2_{yy} G(x, y_x)v^i - \nabla_y F(x, y_x))||^2 \leq 2L_G^2 D_v^2 + 2C_F^2$. We denote $G_q^2 = 2L_G^2 D_v^2 + 2C_F^2$. Next, following the analysis in [126, 127], we consider the virtual sequence $\tilde{v}^i = v^i - e^i$, where we have:

$$\begin{aligned}\tilde{v}^i &= v^i - e^i = v^i - \alpha_{i-1}\nabla q(v^{i-1}) - e^{i-1} + C(\alpha_{i-1}\nabla q(v^{i-1}) + e^{i-1}) \\ &= v^{i-1} - e^{i-1} - \alpha_{i-1}\nabla q(v^{i-1}) = \tilde{v}^{i-1} - \alpha_{i-1}\nabla q(v^{i-1})\end{aligned}$$

Then we have:

$$\begin{aligned}&||\tilde{v}^i - v^*||^2 = ||\tilde{v}^{i-1} - \alpha_{i-1}\nabla q(v^{i-1}) - v^*||^2 \\ =&||\tilde{v}^{i-1} - v^*||^2 - 2\alpha_{i-1}\langle \tilde{v}^{i-1} - v^*, \nabla q(v^{i-1})\rangle + \alpha_{i-1}^2||\nabla q(v^{i-1})||^2 \\ \leq&||\tilde{v}^{i-1} - v^*||^2 - 2\alpha_{i-1}\langle \tilde{v}^{i-1} - v^{i-1}, \nabla q(v^{i-1})\rangle \\ &+ 2\alpha_{i-1}\langle v^* - v^{i-1}, \nabla q(v^{i-1})\rangle + \alpha_{i-1}^2 G_q^2\end{aligned} \tag{6.15}$$

In the last inequality, we use the fact that $\nabla q(v^{i-1})$ is upper-bounded. Then, since $q(v)$ is strongly convex, we have $q(v^*) \geq q(v^{i-1}) + \langle v^* - v^{i-1}, \nabla q(v^{i-1})\rangle + \frac{\mu_G}{2}||v^* - v^{i-1}||^2$. Furthermore, by the triangle inequality, we have: $||v^* - v^{i-1}||^2 \geq \frac{1}{2}||v^* - \tilde{v}^{i-1}||^2 - ||\tilde{v}^{i-1} - v^{i-1}||^2$. Combine these two inequalities, we can upper bound the third term in Eq. (6.15) and have:

$$\begin{aligned}||\tilde{v}^i - v^*||^2 \leq&(1 - \frac{\mu_G\alpha_{i-1}}{2})||\tilde{v}^{i-1} - v^*||^2 - 2\alpha_{i-1}\langle e^{i-1}, \nabla q(v^{i-1})\rangle \\ &+ \mu_G\alpha_{i-1}||e^{i-1}||^2 - 2\alpha_{i-1}(q(v^{i-1}) - q(v^*)) + \alpha_{i-1}^2 G_q^2\end{aligned} \tag{6.16}$$

Now, we bound $\langle e^{i-1}, \nabla q(v^{i-1})\rangle$, first by the triangle inequality, we have: $-\langle e^{i-1}, \nabla q(v^{i-1})\rangle \leq ||e^{i-1}||||\nabla q(v^{i-1})|| \leq (L_{G_y}||e^{i-1}||^2 + \frac{1}{4L_{G_y}}||\nabla q(v^{i-1})||^2)$. Then, by the smoothness of $q(v)$, we have

$$-\langle e^{i-1}, \nabla q(v^{i-1})\rangle \leq (L_{G_y}||e^{i-1}||^2 + \frac{1}{2}(q(v^{i-1}) - q(v^*)))$$

combine the two inequalities with Eq. (6.16), we have:

$$\begin{aligned} q(v^{i-1}) - q(v^*) \leq & \frac{(1-\mu_G\alpha_{i-1}/2)}{\alpha_{i-1}}||\tilde{v}^{i-1} - v^*||^2 - \frac{1}{\alpha_{i-1}}||\tilde{v}^i - v^*||^2 \\ & + (2L_{G_y} + \mu_G)||e^{i-1}||^2 + \alpha_{i-1}G_q^2 \end{aligned} \tag{6.17}$$

Note that we rearrange the terms and move $q(v^{i-1}) - q(v^*)$ to the left. Now, we bound the term $||e^i||^2$. By Assumption 140, and the count sketch memory complexity in [129], suppose that we use the count sketch compressor and the compressed gradients have dimension $O(log(d/\delta)/\tau)$, we can recover the τ heavy hitters with probability at least $1-\delta$. Since we transfer compressed gradients I times, we have the communication cost of $O(log(dI/\delta)/\tau)$ by a union bound. Then for all $i \in [I]$, we have:

$$\begin{aligned} ||e^i||^2 =& ||\alpha_{i-1}\nabla q(v^{i-1}) + e^{i-1} - C(\alpha_{i-1}\nabla q(v^{i-1}) + e^{i-1})||^2 \\ \leq & (1-\tau)||\alpha_{i-1}\nabla q(v^{i-1}) + e^{i-1}||^2 \\ \leq & (1-\tau)(\alpha_{i-1}^2(1+\frac{1}{\gamma})||\nabla q(v^{i-1})||^2 + (1+\gamma)||e^{i-1}||^2) \\ \leq & (1-\tau)(1+\gamma)||e^{i-1}||^2 + (1-\tau)\alpha_{i-1}^2(1+\frac{1}{\gamma})G_q^2 \end{aligned} \tag{6.18}$$

Next, we choose $\gamma = \frac{\tau}{2(1-\tau)}$, then we can prove

$$||e^i||^2 \leq \frac{(1-\tau)(2-\tau)(1+\frac{1}{a})^2\alpha_i^2 G_q^2}{\tau(1-(1-\frac{\tau}{2})(1+\frac{1}{a})^2)}$$

by induction, we omit the derivation here due to space limitation. By $a > \frac{2-\tau}{\tau}(\sqrt{\frac{2}{2-\tau}}+1)$, so the denominator is positive. Inserting the bound for $||e^i||^2$ back to Eq. (6.17), we have:

$$\begin{aligned} q(v^{i-1}) - q(v^*) \leq & \frac{(1-\frac{\mu_G \alpha_{i-1}}{2})}{\alpha_{i-1}}||\tilde{v}^{i-1} - v^*||^2 - \frac{1}{\alpha_{i-1}}||\tilde{v}^i - v^*||^2 \\ & + \frac{2(2L_{G_y} + \mu_G)\alpha_{i-1}^2 G_q^2}{\tau(1-(1-\frac{\tau}{2})(1+\frac{1}{a})^2)} + \alpha_{i-1} G_q^2 \end{aligned}$$

Finally, we average v^i with weight $w_i = (i+a)^2$, choose $\alpha = \frac{8}{\mu_G(i+a)}$, then by Lemma 3.3 in [126] and the strong convexity of $q(v)$, we get the upper bound of $||v^I - v^*||^2$ as shown in the Theorem. □

6.7.2 Proof for Non-iterative algorithm

The proof for Corollary 136 is included in Theorem 9 in [164]. The full version of Theorem 144 in the main text is as follows:

Theorem 150. *(Theorem 144) For any given $\epsilon, \delta \in (0, 1/2)$, if $S_1 \in \mathbb{R}^{r_1 \times d}$ is a $(\lambda_1\epsilon, \delta/2)$ sketch matrix and $S_2 \in \mathbb{R}^{r_2 \times d}$ is a $(\lambda_2\epsilon, \delta/2)$ sketch matrix. Under Assumption 141, with probability at least $1-\delta$ we have:*

$$||\hat{\nabla} h(x) - \nabla h(x)|| \leq \epsilon ||v^*||$$

where $\lambda_1 = \frac{5\mu_G}{7\sqrt{r_s}C_{G_{xy}}L_{G_y}}, \lambda_2 = \frac{1}{3(r_1+1)}$.

Proof. For convenience, we denote $H_{yy} = \nabla^2_{yy}G(x, y_x)$, $H_{xy} = \nabla^2_{xy}G(x, y_x)$, $g_y = \nabla_y F(x, y_x)$, $g_x = \nabla_x F(x, y_x)$, $g = \nabla h(x)$, $\hat{g} = \hat{\nabla} h(x)$, and denote $\epsilon_1 = \lambda_1\epsilon$, $\epsilon_2 = (r_1 + 1)\lambda_2\epsilon$. Furthermore, we denote $v^* = \arg\min_v ||H_{yy}v - g_y||_2^2$, $\hat{\omega} = \arg\min_\omega ||S_2 H_{yy} S_1^T \omega - S_2 g_y||_2^2$, $v_{s_1} = \arg\min_v ||H_{yy}S_1^T S_1 v - g_y||_2^2$ and $\omega_{s_1} = \arg\min_\omega ||H_{yy}S_1^T\omega - g_y||_2^2$. Then we have the following.

$$\begin{aligned}(1-\epsilon_2)||H_{yy}S_1^T\hat{\omega} - g_y|| &\leq ||S_2 H_{yy}S_1^T\hat{\omega} - S_2 g_y|| \\ &\leq ||S_2 H_{yy}S_1^T\omega_{s_1} - S_2 g_y|| \leq (1+\epsilon_2)||H_{yy}S_1^T\omega_{s_1} - g_y||\end{aligned} \tag{6.19}$$

The second inequality is by the definition of $\hat{\omega}$, the first and third inequality use the fact that S_2 is a $(\lambda_2\epsilon, \delta/2)$ sketch matrix and by Corollary 136, it is a $\lambda_2\epsilon \times (r_1 + 1) = \epsilon_2$ subspace embedding matrix over the column space $[H_{yy}S_1^T, g_y]$, so Eq. (6.19) holds with probability $1-\delta/2$. Next, since S_1 is a $(\epsilon_1, \delta/2)$ sketching matrix, with probability $1-\delta/2$, we have:

$$\begin{aligned}&||H_{yy}S_1^T S_1 v^* - H_{yy}v^*|| \leq \epsilon_1 ||v^*||_F ||H_{yy}||_F \\ &\rightarrow ||(H_{yy}S_1^T S_1 v^* - g_y) - (H_{yy}v^* - g_y)|| \leq \epsilon_1 ||v^*||_F ||H_{yy}||_F \\ &\rightarrow ||H_{yy}S_1^T S_1 v^* - g_y|| \leq ||H_{yy}v^* - g_y|| + \epsilon_1 ||v^*||_F ||H_{yy}||_F\end{aligned} \tag{6.20}$$

In the last step, we use the triangle inequality $||x|| - ||y|| \leq ||x - y||$. Combining Eq. (6.20)

and the definition of v_{s_1}, also noticing that $span(S_1 v) \subset span(\omega)$, we have:

$$\begin{aligned} &||H_{yy} S_1^T \omega_{s_1} - g_y|| \leq ||H_{yy} S_1^T S_1 v_{s_1} - g_y|| \\ &\leq ||H_{yy} S_1^T S_1 v^* - g_y|| \leq ||H_{yy} v^* - g_y|| + \epsilon_1 ||v^*||_F ||H_{yy}||_F \end{aligned} \tag{6.21}$$

Finally we combine Eq. (6.19) and (6.21) to get:

$$||H_{yy} S_1^T \hat{\omega} - g_y|| \leq \frac{(1+\epsilon_2)}{(1-\epsilon_2)} (||H_{yy} v^* - g_y|| + \epsilon_1 ||v^*||_F ||H_{yy}||_F) \tag{6.22}$$

By the union bound, Eq. (6.22) holds with probability $1-\delta$. Since H_{yy} is positive definite (invertible), we have $||H_{yy} v^* - g_y|| = 0$. Eq. (6.22) can be simplified further as:

$$||H_{yy} S_1^T \hat{\omega} - g_y|| \leq \frac{\epsilon_1 (1+\epsilon_2)}{(1-\epsilon_2)} ||v^*||_F ||H_{yy}||_F$$

Moreover, $G(x, y)$ is μ_G-strongly convex (Assumption 138,), we have:

$$\begin{aligned} &||H_{yy} S_1^T \hat{\omega} - g_y||^2 = ||H_{yy} S_1^T \hat{\omega} - g_y - (H_{yy} v^* - g_y)||^2 \\ =&||H_{yy} (S_1^T \hat{\omega} - v^*)||^2 \geq \mu_G^2 ||S_1^T \hat{\omega} - v^*||^2 \end{aligned} \tag{6.23}$$

Combining the above two equations, we have the following.

$$||S_1^T \hat{\omega} - v^*|| \leq \frac{\epsilon_1 (1+\epsilon_2)}{\mu_G (1-\epsilon_2)} ||v^*||_F ||H_{yy}||_F \tag{6.24}$$

As for $||H_{yy}||_F$, by Assumption C, we have $||H_{yy}||_F = \sqrt{\sum_i \sigma_i^2} \leq \sqrt{r_s} \sigma_{max} = \sqrt{r_s} L_{G_y}$. Then Eq. (6.24) can be simplified to $||S_1^T \hat{\omega} - v^*|| \leq \frac{\sqrt{r_s} L_{G_y} \epsilon_1 (1+\epsilon_2)}{\mu_G (1-\epsilon_2)} ||v^*||$. Then we have the

following:

$$
\begin{aligned}
||\hat{g} - g|| &= ||H_{xy} S_1^T \hat{\omega} - H_{xy} v^*|| = ||H_{xy}(S_1^T \hat{\omega} - v^*)|| \\
&\leq \frac{\sqrt{r_s} C_{G_{xy}} L_{G_y} \epsilon_1 (1+\epsilon_2)}{\mu_G (1-\epsilon_2)} ||v^*||
\end{aligned}
\tag{6.25}
$$

By choice of $\epsilon_1 = \frac{5\mu_G \epsilon}{7\sqrt{r_s} C_{G_{yx}} L_{G_y}}$ and $\epsilon_2 = \frac{\epsilon}{3} < \frac{1}{6}$, *i.e.* $\frac{1+\epsilon_2}{1-\epsilon_2} = 1 + \frac{2\epsilon_2}{1-\epsilon_2} < \frac{7}{5}$. So, we get $||\hat{g} - g|| < \epsilon ||v^*||$ with probability at least $1 - \delta$. The proof is complete. □

6.7.3 Proof for Convergence Analysis

The full version of Theorem 147 in the main text is provided as follows:

Theorem 151. *(Theorem 147) Under Assumption 138, 139, we pick the learning rate* $\eta = \frac{1}{2L_h\sqrt{K+1}}$, *then we have:*

a) Suppose that $\{x_k\}_{k\geq 0}$ *is generated from the non-iterative Algorithm 3, under Assumption 141 and* $\epsilon = (K+1)^{-1/4}$, *we have the following:*

$$
E[||\nabla h(x_k)||^2] \leq \left(32 L_h \left(h(x_0) - h(x^*) \right) + \frac{16 C_F^2}{\mu_G^2} \right) \frac{1}{\sqrt{K}}
$$

b) Suppose $\{x_k\}_{k\geq 0}$ *are generated from the non-iterative Algorithm 4, under Assumption 140,* $I = L_h\sqrt{K+1}$, *we have:*

$$
E[||\nabla h(x_k)||^2] \leq \frac{C_1}{K^{3/2}} + \frac{C_2}{K} + \frac{C_3}{\sqrt{K}}
$$

where C_1, C_2 *and* C_3 *are some constants.* $C_1 = 12 C_{G_{xy}}^2 a^3 D_v^2 / L_h^3$, $C_2 = (6144(2L_{G_y} +$

$\mu_G)C^2_{G_{xy}}G^2_q)/(\mu^3_G\tau(1-(1-\frac{\tau}{2})(1+\frac{1}{a})^2)L^2_h)+384aC^2_{G_{xy}}G^2_q/\mu^2_GL^2_h$, $C_3 = 32L_h(h(x_0)-h(x^*))+(192C^2_{G_{xy}}G^2_q)/\mu^2_GL_h$.

Proof. As stated in Proposition 146, $h(x)$ is L_h smooth: $h(x_{k+1}) \leq h(x_k)+\langle\nabla h(x_k), x_{k+1}-x_k\rangle+\frac{L_h}{2}||x_{k+1}-x_k||^2$, and by the update rule of outer variable $x_{k+1} = x_k - \eta\hat{\nabla}h(x_k)$, we have:

$$\begin{aligned}
h(x_{k+1}) \leq& h(x_k) - \eta\langle\nabla h(x_k), \hat{\nabla}h(x_k)\rangle + \frac{L_h}{2}\eta^2||\hat{\nabla}h(x_k)||^2\\
\leq& h(x_k) - \eta||\nabla h(x_k)||^2 + \eta\langle\nabla h(x_k), \nabla h(x_k) - \hat{\nabla}h(x_k)\rangle\\
&+\frac{L_h}{2}\eta^2||\hat{\nabla}h(x_k) - \nabla h(x_k) + \nabla h(x_k)||^2
\end{aligned}$$

Using the triangle inequality and the Cauchy-Schwarz inequality, we have $\langle\nabla h(x_k), \nabla h(x_k) - \hat{\nabla}h(x_k)\rangle \leq \frac{1}{2}||\nabla h(x_k)||^2 + \frac{1}{2}||[\hat{\nabla}h(x_k) - \nabla h(x_k)||^2$ and $||\hat{\nabla}h(x_k) - \nabla h(x_k) + \nabla h(x_k)||^2 \leq 2||\hat{\nabla}h(x_k) - \nabla h(x_k)||^2 + 2||\nabla h(x_k)||^2$. We combine these two inequalities with the above inequality.

$$h(x_{k+1}) \leq h(x_k) - \eta(\frac{1}{2} - \eta L_h)||\nabla h(x_k)||^2 + \eta(\frac{1}{2} + \eta L_h)e_k \tag{6.26}$$

where we denote $||\hat{\nabla}h(x_k) - \nabla h(x_k)||^2$ as e_k. Next we prove the two cases respectively.

Case (a): By Theorem 144 and $||v^*|| = ||\nabla^2_{yy}G(x, y_x)^{-1}\nabla_y F(x, y_x)|| \leq \frac{C_F}{\mu_G}$, we have: $e_k \leq \epsilon^2||v^*_k||^2 \leq \epsilon^2 C^2_{F_y}/\mu^2_G$. Combine this inequality with Eq. (6.26) and telescope from 1 to K, we have::

$$\sum_{k=0}^{K-1}\eta(\frac{1}{2} - \eta L_h)||\nabla h(x_k)||^2 \leq h(x_0) - h(x^*) + \sum_{k=0}^{K-1}\eta(\frac{1}{2} + \eta L_h)\frac{\epsilon^2 C^2_F}{\mu^2_G}$$

We select x_k with probability proportional to $\eta(\frac{1}{2} - \eta L_h)$, and pick $\eta = \frac{1}{2L_h\sqrt{K+1}}$, $\epsilon = (K+1)^{-1/4}$ we have:

$$E[||\nabla h(x_k)||^2] \leq \frac{1}{\sqrt{K}}\left(32L_h(h(x_0) - h(x^*)) + \frac{16C_F^2}{\mu_G^2}\right)$$

where we use $\sum_{k=0}^{K-1} \eta(\frac{1}{2} - \eta L_h) \geq \frac{\sqrt{K}}{32L_h}$, $\sum_{k=0}^{K-1} \epsilon^2\eta(\frac{1}{2} + \eta L_h) \leq \frac{1}{2L_h}$.

Case (b): for the iterative algorithm, we notice that $e_k = ||\nabla_x F(x, y_x) - \nabla^2_{xy} G(x, y_x)v^I - \nabla_x F(x, y_x) - \nabla^2_{xy} G(x, y_x)v^*||^2 \leq C^2_{G_{xy}}||(v^I - v^*)||^2$. Combine this inequality with Eq. (6.26) and telescope from 1 to K as case (a), we have:

$$\sum_{k=0}^{K-1} \eta(\frac{1}{2} - \eta L_h)||\nabla h(x_k)||^2 \leq h(x_0) - h(x^*) + \sum_{k=0}^{K-1} \eta(\frac{1}{2} + \eta L_h)C^2_{G_{xy}}||(v^I - v^*)||^2$$

By choosing the learning rate *eta* and I as in the Theorem, then combine Theorem 142, it is straightforward to get the upper bound of $||\nabla h(x_k)||^2$ as stated in the Theorem. □

Chapter 7: Device-Wise Federated Network Pruning

7.1 Introduction

Machine learning algorithms often rely on large amounts of data, but privacy restrictions can prevent data from being easily shared across different organizations. For instance, hospitals may have isolated data that are limited in size and cannot be used to train a high-quality model with good predictive accuracy. Collaboration between organizations to train a machine learning model on their combined data can lead to better results, but sharing data is often not possible due to privacy policies and regulations [166]. This problem of 'data islands' is not limited to hospitals and can be found in other areas such as finance, government, and supply chains. Federated learning [92, 167, 168] has emerged as a popular research topic in the machine learning and computer vision communities as a solution to these issues.

Convolution Neural Networks (CNNs) have achieved remarkable success in various computer vision tasks[169, 170, 171], but to address real-world challenges, recent CNNs have become wider and deeper, leading to improved performance on various benchmarks. However, this increased capacity comes at the cost of higher computational and storage requirements, which prohibit CNNs from being deployed on edge devices. Consequently, numerous efforts [172, 173] have been made to reduce the size of CNNs to enable their

deployment on mobile and embedded devices. Among different directions, weight pruning [174] and structural pruning [175] are two major ways to reduce the model size. Network pruning methods have achieved promising results. However, most existing methods do not consider heterogeneous (non-iid) local data distributions. Instead, they upload local data to the server to train and prune the model based on the whole dataset.

There are several existing works [176, 177, 178, 179, 180] on network pruning under non-iid local data distributions. These methods mainly focus on weight pruning. SCBFwP [176] tries to perform channel pruning under non-iid data distributions, but they mostly rely on channel norms as the importance score, and they did not show how to scale their method to larger CNNs. Our method is designed to perform channel pruning given a certain computational budget (measured in FLOPs) on each device. Because, unlike weight pruning, channel pruning can achieve acceleration and compression without any post-processing steps.

Previous research [181, 182] show that data-driven pruning approaches often perform much better than using a fixed criterion (like channel norm [175]). Inspired by this result, we propose to discover the proper sub-network following the guidance of local data distributions. Specifically, we divide the learning of sub-networks into two parts. In the first part, we design a server-side sub-network, which is used to serve as a base model for device-wise sub-network. Ideally, weights that are not useful for any device will be removed from the server-side sub-network. In the second part, device-side sub-networks will be adaptively generated for each device. The device-side sub-network will be pruned to meet the specific resource requirement for each device. Both the server-side sub-network and device-side sub-networks are generated by mapping the device id(s) into the network embedding space.

The embedding for each sub-network will then be fed into a hypernetwork [183] to generate the corresponding structure. The hypernetwork and the embedding are trained together through gradient-based optimization algorithms in a federated fashion. To control the communication costs brought by training the hypernetwork, we set the fraction of update steps for the hypernetwork to be small. So that the overhead of training these sub-networks is not large. In addition, to improve the training efficiency given the limited update budget, we perform iterative training of model weights and the hypernetwork, which makes the hypernetwork adapt to changes in model weights. The training process of our method may pose challenges to the convergence of the model. We show that our method can be converted into a bi-level optimization problem under the FL setting. We further provide the theoretical convergence guarantee showing that our method can converge to a stationary point. The contribution of this work can be summarized as follows:

- We proposed a novel channel pruning method for federated learning. A server-side network and device-wise sub-networks are learned to achieve a better trade-off between the performance and the computational resource.
- We proposed to use an embedding layer and a hypernetwork to generate sub-networks on each device. As a result, no sub-network structure needs to be stored for each device.
- We provided the theoretical guarantee of convergence for our method for federated learning by reformulating our method as a bi-level optimization problem.
- Extensive experiments on CIFAR-10, CIFAR-100, and TinyImageNet show the effectiveness of our method across different models like ResNet-56, ResNet-18/34, and

MobileNet-V2.

7.2 Related Works

Federated Learning. Federated learning (FL) is a new kind of distributed learning approach that involves a server coordinating a group of clients/devices to learn a model. In FL, at each epoch, devices retrieve the model from the server, train the model locally for several steps, and then upload the updated model back to the server. The server aggregates the updates from devices to update the global model. FL presents several challenges that need to be addressed for effective implementation. Firstly, devices in FL often have different data distributions. Various methods are proposed to solve the data heterogeneity [96, 112, 113, 114, 115, 116, 155]. Second, the communication between devices and the server is expensive and is a critical bottleneck in FL training. Compression techniques are applied in FL to reduce the communication [124, 125, 127, 128, 130, 143], Finally, although in FL, the server does not have access to the user data, model inversion attack [184] is shown to recover the user information based on the model updates. Cryptography techniques are applied to improve the privacy of FL, such as homomorphic encryption [122, 185], differential privacy [92] and multiparty secure computation [123] *etc.*. In addition to the challenges discussed, there are several other challenges in FL, such as fairness and model interpretability, a more comprehensive review of FL can be found in [168, 186].

Network Pruning. (1) Regular Setting. Most network pruning methods assume they can easily access all samples without restrictions. Early works [174, 175] simply use L_1 or L_2 norm to measure the importance of weights or structures. Calculating norms

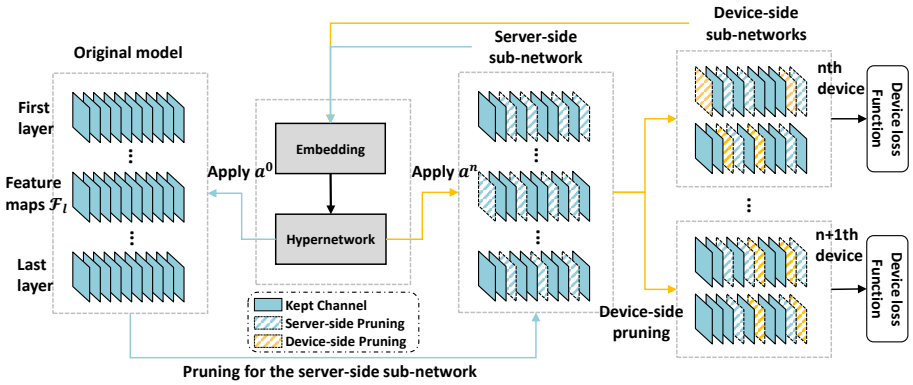


Figure 7.1: An overview of our proposed method. We first generate network embeddings for server-side and device-side sub-networks given their ids. We then use them as the inputs to the hypernetwork to produce the corresponding sub-networks. We then optimize the hypernetwork given loss functions on each device.

does not require samples, and it can be seamlessly extended to the FL setting. However, the performance of these methods is often worse than methods [181, 182, 187] that require samples for pruning. One direction of data-driven pruning methods relies on batch normalization (BN) [188] layers since BN is popular for the design of recent CNNs [146, 189]. These methods utilize the scaling factor of BN to indicate which channels are important. Liu *et al.* [190] use sparse regularization on the scaling factors of BN to prune channels, where a channel is pruned if its corresponding scaling factor is small. Pruning methods that involve BN are effective. However, BN layers can pose challenges in the FL setting due to the varying data distribution across devices, leading to significant differences in BN layers' running mean and variance. Another research direction frames channel pruning as a constrained optimization problem [182, 187, 191, 192, 193, 194, 195, 196, 197]. In this direction, learnable parameters are utilized to control each channel, and these parameters are end-to-end differentiable, allowing for gradient-based optimization methods. Since these methods do not rely on BN layers, they can potentially be extended to the FL setting. Alongside advancements in vision, Natural Language Processing (NLP) has significantly

progressed, demonstrated by key studies [198, 199, 200, 201, 202, 203, 204]. Concurrently, structure pruning has emerged as a method to improve the efficiency of large language models, as shown in recent research [205].

(2) Non-iid Setting. Without specialized treatment, regular pruning methods can impose strong biases in pruned models because of heterogeneous (non-iid) local data distributions. Shao *et al.* [176] employs local training with a full-size model to discard unimportant channels (measured in channel norms) on devices. FedPrune [178] guides pruning based on updated activations. LotteryFL [177] iteratively prunes a full-size model on devices. PruneFL [179] reduces local computational costs by finer pruning a coarse-pruned model. ZeroFL [144] partitions weights into active and non-active weights and stores sparsified weights and activations for backward propagation, and it also needs to store non-active weights and dense gradients. Bibikar *et al.* [145] employs mask adjustment on devices and sparse aggregation and magnitude pruning on the server to generate a new global model. The FedTiny [180] approach incorporates an adaptive batch normalization (BN) selection module, which adaptively obtains an initially pruned model that can better fit deployment scenarios. Most aforementioned methods focus on weight-level pruning/sparsity, often requiring high communication costs to compute importance scores for all parameters. On the other hand, our method learns the channel configuration of each layer, which is less resource-demanding.

Federated Bilevel Optimization. Our channel pruning can be viewed as a federated bilevel optimization problem (FedBiO). The general FedBiO problem has been studied in the literature [70, 100, 101]. FedNest [100] studied the general nested federated problems with FedBiO being a special case, and it utilized variance reduction to tackle

the heterogeneity of lower level problems; simFBO [101] and FedBiOAcc [71] adapts the single loop bilevel optimization problems to the federated learning setting. Some applications in FL can be viewed as bilevel optimization problems, such as noisy labels [99] and communication-efficient FL [71].

7.3 Methodology

7.3.1 Notations

We will first introduce our notations before formally describing our method. In a convolutional neural network (CNN), the feature map of the lth layer is denoted by $\mathcal{F}_l \in \mathrm{Re}^{C_l \times W_l \times H_l}$, where C_l represents the number of channels, and H_l and W_l represent the height and width of the current feature map. L denotes the total number of layers in the CNN. For simplicity, we ignore the mini-batch dimension of feature maps in our notation.

7.3.2 Federated Learning Setting

We first describe the federated learning problem considered in this paper. In the FL setting, we train a neural network on N local datasets $D_n,\ n \in \{1, 2, 3 \cdots, N\}$. Through this paper, the data distribution on local devices is heterogeneous. To train a neural network in this setting, we want to optimize the following optimization problem:

$$\min_{\mathcal{W}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\mathcal{W}, D_n), \tag{7.1}$$

where $\mathcal{W}$ is the weights of the CNN, and $\mathcal{L}$ is the objective function. One common method to minimize communication costs is by using local stochastic gradient descent (SGD), where the local device performs several update steps with their local data before averaging the model weights $\mathcal{W}$. FedAvg [92] is a popular algorithm that adopts this approach.

7.3.3 Generate Sub-network Architectures

To prune a model, we need to first generate the corresponding sub-network architectures. To achieve this goal, we use a binary vector $\mathbf{a} \in \{0, 1\}$ to represent whether to keep or prune a channel. To facilitate the learning of the sub-network architecture, we use a hypernetwork [183] (HN) to generate the architecture vector $\mathbf{a}$:

$$\mathbf{a} = \mathrm{HN}(e; \theta_{\mathrm{HN}}), \tag{7.2}$$

where e is the network embedding of the corresponding device or server, which will be discussed later, and θ_{HN} is the parameters of the HN. We use Straight-Through Gumbel-Sigmoid [206] to enable gradient calculation for the HN. To control the pruning of each channel, we apply $\mathbf{a}$ to the feature map of each layer:

$$\hat{\mathcal{F}}_l = \mathbf{a}_l \odot \mathcal{F}_l, \tag{7.3}$$

where $\hat{\mathcal{F}}_l$ is the feature map after applying $\mathbf{a}_l$ (the architecture vector of lth layer). Note that we insert $\mathbf{a}_l$ after normalization and activation layers, which correspond to control the output channels of the previous convolution layer and input channels of the next convolution

layer.

An alternative approach to control pruning is to add learnable parameters for each channel. However, this approach presents challenges in the context of FL. If we only train a single sub-network for compression, local parameters must be accumulated on the server. Using individual learnable parameters for each channel may result in significantly different parameters across devices due to the non-iid setting, rendering the final parameters meaningless (often close to 0.5 before binarization). Additionally, if we aim to learn device-wise sub-networks for pruning, we would need to train N sets of parameters for all devices, making it unclear how to share knowledge between devices.

7.3.4 Architecture Embedding for Server and Device Side Sub-networks

Our method aims to find the appropriate server-side sub-network and device-side sub-networks. The server-side sub-network serves as the weight bank for device-side sub-networks. In addition, it reduces the communication and training costs at the finetuning stage and alleviates the memory burden on each device. Device-side sub-networks are used to meet the resource constraint of each device at the inference time, assuming the training and test data distribution on each device are similar. Using HN potentially provides a unique opportunity to share knowledge between server-side and device-side sub-networks. To achieve this, we introduce an embedding layer to produce the embedding for each sub-network:

$$e_n = \text{Emb}(n; \theta_{\text{Emb}}),\ n = 0, \cdots, N, \tag{7.4}$$

where θ_{Emb} is the parameters of the embedding layer Emb, and n is the index for each device. In addition, we let $n = 0$ represent the embedding for the server-side sub-network. By putting Eq. 7.2 and Eq. 7.4 together, we can generate the server-side sub-network and device-side sub-networks by using:

$$\begin{aligned} \mathbf{a}^0 &= \text{HN}(\text{Emb}(0; \theta_{\text{Emb}}); \theta_{\text{HN}}), \\ \mathbf{a}^n &= \text{HN}(\text{Emb}(n; \theta_{\text{Emb}}); \theta_{\text{HN}}), \ n = 1, \cdots, N, \end{aligned} \tag{7.5}$$

where $\mathbf{a}^0$ is the server-side sub-network and $\mathbf{a}^n$ are device-side sub-networks. The embedding layer is used more frequently in natural language processing [207], but it well suits our task since it can covert the device id into a corresponding sub-network by combining Emb and HN.

7.3.5 Channel Pruning for Federated Learning

Given the aforementioned settings, we can now formally introduce the objective function for channel pruning. The channel pruning problem can be viewed as a constrained optimization problem, where the constraint is used to control the computational resource of the sub-network. The channel pruning objective function can be formulated as follows:

$$\begin{aligned} &\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\mathcal{W}, D_n^{\mathbf{a}}; \mathbf{a}^0 \odot \mathbf{a}^n) \\ &+ \lambda[(\mathcal{R}(T(\mathbf{a}^0), p_s T_{\text{total}}) + \frac{1}{N} \sum_{n=1}^{N} \mathcal{R}(T(\mathbf{a}^0 \odot \mathbf{a}^n), p_d^n T_{\text{total}})], \end{aligned} \tag{7.6}$$

Algorithm 15 Learning Server-side and Device-side Sub-networks

Require: $D_n, D_n^{\mathbf{a}}, p_s, p_d^n, \lambda, S, K, r_{\mathcal{W}}, r_\theta, r_{\text{HN}}$

Initialization: $k_{\text{HN}} \leftarrow 0$
Broadcast the current state of the CNN
for $k = 1 \rightarrow K$ **do**
  for each device in parallel do
    Calculate gradients w.r.t. the loss in Eq. (7.7)
    Update local CNN weights using the preferred optimizer
  end for
  if k mod $r_{\mathcal{W}} = 0$ **then**
    Randomly sample S devices
    Average CNN states and broadcast the updated states
  end if
  if k mod $r_{\text{HN}} = 0$ **then**
    for each device in parallel do
      Calculate gradients w.r.t. θ given the loss in Eq. (7.6)
      Update local HN weights (including Emb) using the preferred optimizer
    end for
    if k_{HN} mod $r_\theta = 0$ **then**
      Randomly sample S devices
      Average HN states and broadcast the updated states
    end if
    $k_{\text{HN}} \leftarrow k_{\text{HN}} + 1$
  end if
end for
Prune the model with resulting $\mathbf{a}^0$, and fine-tune it

where θ contains both θ_{HN} and θ_{Emb}, $\mathbf{a}^0$ and $\mathbf{a}^n$ are generated by using Eq. 7.5, $D_n^{\mathbf{a}}$ is a subset of the local datasets D_n, $\mathcal{R}$ is the regularization loss to control the FLOPs of the sub-network, $p_s \in (0, 1]$ is a predefined hyperparameter to control the preserved FLOPs of the server-side sub-network, $p_d^n \in (0, 1],\ n = 1, \cdots, N$ are also predefined hyperparameters to control the FLOPs of sub-networks on each device, $T(\mathbf{a}^0)$ or $T(\mathbf{a}^0 \odot \mathbf{a}^n)$ is the current FLOPs decided by the sub-network architecture $\mathbf{a}^0$ or $\mathbf{a}^0 \odot \mathbf{a}^n$, and T_{total} is the total FLOPs of the CNN. The FLOPs constraint $\mathcal{R}(x, y)$ is generally a regression problem, but regular regression loss functions, like MAE and MSE, can hardly push $\mathcal{R}$ to near zero values. We let $\mathcal{R}(x, y) = \log\left(\frac{\max(x,y)}{y}\right)$ to push $\mathcal{R}$ to be close to 0. In addition, we explicitly require

that if a channel is pruned by the server-side sub-network, the corresponding device-side sub-networks should not update the corresponding position, and the detail is shown in the supplementary materials.

We perform iterative updates between model weights and the sub-network architectures. When updating model weights, we use the following equation:

$$\min_{\mathcal{W}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\mathcal{W}, D_n; \mathbf{a}^0 \odot \mathbf{a}^n). \tag{7.7}$$

When training model weights, we freeze the sub-networks generated by the HN, and when training the HN, we also freeze $\mathcal{W}$.

The overview of our method is shown in Fig. 7.1. The algorithm of training our method for **one epoch** is shown in Alg. 15. In Alg. 15 D_n and $D_n^{\mathbf{a}}$ are local datasets and their sub-set for training the HN. λ is the hyperparameter to control FLOPs constraints, S is the number of sampled devices, K is the number of iterations within one epoch, $r_{\mathcal{W}}$ is the state average interval for the CNN, r_θ is the state average interval for the HN, r_{HN} decides the frequency of training the HN. To control the communication and the additional training costs, we introduce three hyperparameters: $r_{\mathcal{W}}$, r_θ and r_{HN}. $r_{\mathcal{W}}$, r_θ controls the communication costs for training the model and the HN. Larger $r_{\mathcal{W}}$ and r_θ will reduce the communication costs, but it may also negatively affect the quality of the final model and generated sub-networks under the FL setting. r_{HN} controls the overall training costs brought by HN. Similarly, larger r_{HN} results in smaller additional training costs, but it makes the training of HN harder since the difference of model weights is larger between consecutive training iterations of the HN. We follow Mime [208] for averaging states of the

optimizers.

Method	Dataset	Architecture	Base Acc	Δ-Acc	Acc	↓ FLOPs (D)	↓ FLOPs (S)
Filter Pruning [175]	CIFAR-10	ResNet-56	91.22%	-0.93%	90.29%	50%	50%
FedOSP				-0.28%	90.94%	50%	50%
FedILP				-0.08%	91.14%	50%	50%
DWNP				**+0.66%**	**91.88%**	50%	20%
Filter Pruning [175]	CIFAR-100	ResNet-18	66.57%	-1.31%	65.26%	50%	50%
FedOSP				-0.61%	65.96%	50%	50%
FedILP				-0.20%	66.37%	50%	50%
DWNP				**+1.74%**	**68.31%**	50%	20%
Filter Pruning [175]				-2.52%	64.05%	70%	70%
FedOSP				-1.79%	64.78%	70%	70%
FedILP				-1.44%	65.13%	70%	70%
DWNP				**+0.05%**	**66.62%**	70%	50%
Filter Pruning [175]		ResNet-34	69.05%	-1.22%	67.83%	50%	50%
FedOSP				-0.29%	68.76%	50%	50%
FedILP				+0.44%	69.49%	50%	50%
DWNP				**+2.17%**	**71.72%**	50%	20%
FedILP		MobileNet-V2	66.76%	-0.22%	66.64%	48%	48%
DWNP				**+1.46%**	**68.22%**	48%	20%

Table 7.1: Results of CIFAR-10 and CIFAR-100. 'Base Acc' represents the baseline training accuracy. 'Δ-Acc' represents the accuracy changes before and after pruning. 'Acc' represents the accuracy after pruning. '↓ FLOPs (D)' and '↓ FLOPs (S)' represent the pruned FLOPs of device-side and server-side sub-networks.

7.3.6 Theoretical Guarantee of Convergence

The objective of channel pruning is finding an optimal sub-network such that the FLOPs constraint is satisfied and the model performance is maximized. In fact, channel pruning in our setting can be viewed as a federated bilevel optimization problem [16, 17]. More formally, we combine Eq.(7.6) and Eq. (7.7) to have:

$$\begin{aligned}
&\min_{\theta} h(\theta) \coloneqq \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\mathcal{W}_\theta, D_n^{\mathbf{a}}; \mathbf{a}^0 \odot \mathbf{a}^n) \\
&+ \lambda[(\mathcal{R}(T(\mathbf{a}^0), p_s T_{\text{total}}) + \frac{1}{N}\sum_{n=1}^{N}\mathcal{R}(T(\mathbf{a}^0 \odot \mathbf{a}^n), p_d^n T_{\text{total}})], \\
&s.t. \mathcal{W}_\theta = \arg\min_{\mathcal{W}} \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\mathcal{W}, D_n; \mathbf{a}^0 \odot \mathbf{a}^n).
\end{aligned} \tag{7.8}$$

From a bilevel's perspective, we do channel pruning by iteratively performing the following steps until convergence: for a given sub-network structure from the HN and Emb,

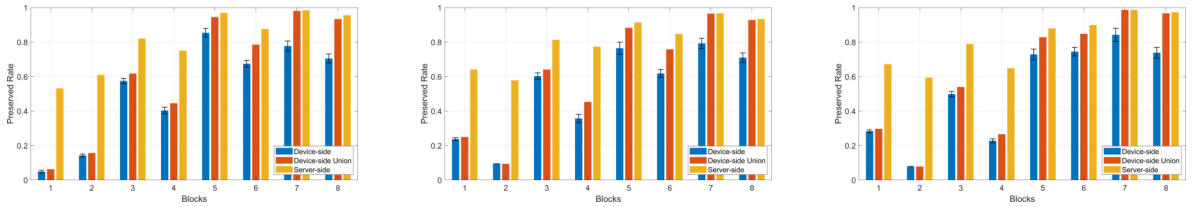
Architecture	Method	Base Top-1 Acc	Base Top-5 Acc	Δ Top-1 Acc	Δ Top-5 Acc	↓ FLOPs (D)	↓ FLOPs (S)
ResNet-18	Filter Pruning [175]	54.99%	78.60%	-1.01%	-0.33%	50%	50%
	FedOSP			-0.18%	+0.48%	50%	50%
	FedILP			+0.07%	+0.65%	50%	50%
	DWNP			**+1.06%**	**+1.10%**	50%	20%
ResNet-34	Filter Pruning [175]	56.32%	79.37%	-0.91%	-0.21%	50%	50%
	FedOSP			-0.20%	+0.21%	50%	50%
	FedILP			-0.03%	+0.34%	50%	50%
	DWNP			**+0.80%**	**+0.74%**	50%	20%

Table 7.2: Comparison results on TinyImageNet with ResNet-18/34. 'Base Top-1/5' represents the baseline training Top-1/5 accuracy. 'Δ Top-1/5 Acc' represents the Top-1/5 accuracy changes before and after pruning.

we first find the optimal model weight $\mathcal{W}$(solving the lower level problem in Eq. (7.8)); then we optimize the sub-network based on this optimal model weight(solving the upper level problem in Eq. (7.8)). Finding the optimal model weight is expensive, especially for modern deep neural networks; we instead optimize the HN and Emb weights θ and the model weight $\mathcal{W}$ alternatively as in Alg. 15. Furthermore, the gradient *w.r.t* the sub-network structure includes both a direct part, which is the direct gradient *w.r.t* θ, and an indirect part due to $\mathcal{W}_\theta$ is a function of θ (the minimizer of the lower level problem). However, the indirect gradient is expensive to evaluate and leads to minor empirical improvement in practice, so we only consider the direct gradient when updating θ in Alg. 15. The convergence of our alternative update method is guaranteed under mild assumptions [12, 32] as stated in Theorem 152 below:

Theorem 152. *Suppose we choose the upper level learning rate η and the lower level learning rate γ as:*

$$\eta = \min\left\{\frac{1}{4Lr_\theta}, \left(\frac{2bN\Delta_\theta}{K_{HN}L\sigma^2}\right)^{1/2}\right\},\ \gamma = \min\left\{\frac{1}{4L}, \left(\frac{\lambda bN}{K_{HN}L^2\sigma^2}\right)^{1/2}\right\},$$



(a) $\alpha = 0.05$ (b) $\alpha = 0.10$ (c) $\alpha = 0.50$

Figure 7.2: The layer-wise pruning rates for device-side sub-networks, the union of device-side sub-networks, and the server-side sub-network with different α. The union of device-side sub-networks represents the union of kept channels from all devices.

then we have:

$$\frac{1}{K_{HN}} \sum_{k_{HN}=0}^{K_{HN}-1} \mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 = O\left(\frac{1}{(bNK_{HN})^{1/2}}\right)$$

where b is the mini-batch size, N is the number of devices, and K_{HN} is the number of update steps to the upper level variable θ.

As stated in Theorem 152, our algorithm converges to a stationary point of Eq. (7.8), with a convergence rate of $O(K_{HN}^{-0.5})$. Furthermore, the algorithm achieves linear speed up *w.r.t* the number of devices and mini-batch size.

7.4 Numerical Experiments

7.4.1 Settings

Datasets and Models. We use CIFAR-10 [142], CIFAR-100 [142], and TinyImageNet [209, 210] to evaluate the performance of our method. Our method uses p_s and p_d^n to control the FLOPs for the server and each device. In the experiment section, we assume p_d^n has the

same value for different devices for a fair comparison with other methods. The detailed choices of p_s and p_d^n are listed in supplementary materials. We choose ResNets [146] and MobileNet-V2 [189] for comparison. For CIFAR-10, we compare our method with other baselines on ResNet-56. For CIFAR-100, we compare our method with other baselines on ResNet-18, ResNet-34, and MobileNet-V2. For TinyImageNet, ResNet-18 and ResNet-34 are used for comparisons. To reduce the negative effects caused by batch normalization layers, we replace batch normalization with layer normalization [211], which has been used frequently in recent designs of vision transformers [212] and CNNs [213]. For the main experiments, we consider N = 10 devices. We use the Dirichlet distribution with $\alpha = 0.5$, as described in [214], to create non-iid partitions on the devices for all datasets. Other settings of N and α are also verified for specific models and datasets. As described in section 7.3, we assume the training and test datasets on each device are similar. To accomplish this, we apply a random permutation to the samples drawn from the Dirichlet distribution for the training dataset and then split the test dataset based on the permuted samples. As a result, the training and test datasets distributions on each device are similar but not the same. More details are given in the supplementary materials. **Baselines.** In addition to the proposed method, we also build three baselines from the literature on channel pruning. **(1) Filter Pruning**: we directly adapt the Filter Pruning [175] to the FL setting, where there are no communication costs for pruning. In this setting, pruning is purely based on the channel norm of the weights. **(2) FedOSP** (**Fed**erated **O**ne-**S**hot **P**runing): this baseline can be seen as an improved version of channel pruning methods with differentiable gates [182, 187, 191] in the one-shot pruning setting. In this setting, we use the HN to generate one sub-network for all devices. The HN is learned in a one-shot

Method	$N = 10$		$N = 25$		$N = 50$	
	Base	Acc	Base	Acc	Base	Acc
FedILP	66.57%	66.37% -0.20%	65.86%	65.37% -0.49%	65.13%	64.88% -0.25%
DWNP		68.31% **+1.74%**		68.09% **+2.23%**		67.58% **+2.45%**

Table 7.3: Performance of pruned models given different numbers of devices N with ResNet-18 on CIFAR-100.

setting when model weights are frozen. **(3) FedILP** (**Fed**erated **I**terative-**L**earning and **P**runing): this baseline can be seen as the simplified version of our method without device-side sub-networks. Through the experiment section, our method is abbreviated as **DWNP** (**D**evice-**W**ise **N**etwork **P**runing). For all settings, we report the mean results across three runs.

Training Settings. We describe the hyperparameters in Alg. 15 in this section. For all methods, we let $S = N$, $\lambda = 2.0$, $r_{\mathcal{W}} = 5$, $r_\theta = 2$ and $r_{\text{HN}} = 10$. K is the number of iterations for training one epoch. For Filter Pruning and FedOSP, we train a base model for 200 epochs, and this base model also servers as the baseline model in Tab. 7.1 and Tab. 7.2. For FedILP and DWNP, we train the model and the hypernetwork from scratch for 200 epochs. For FedIPL and DWNP, we start the training of the hypernetwork after $\frac{1}{4}$ of the total training epochs, which avoids misleading pruning results when weights are not properly trained. We finetune the model for 200 epochs for all methods to recover the performance. For each local dataset, we sample 10% of the training samples to construct $D_n^{\mathbf{a}}$. When updating the local $\mathcal{W}$, we use SGD with momentum 0.9 and a start learning rate 0.1. When updating the local θ, we use Adam [215] with a start learning rate of 10^{-3}. Other training details are shown in the supplementary materials.

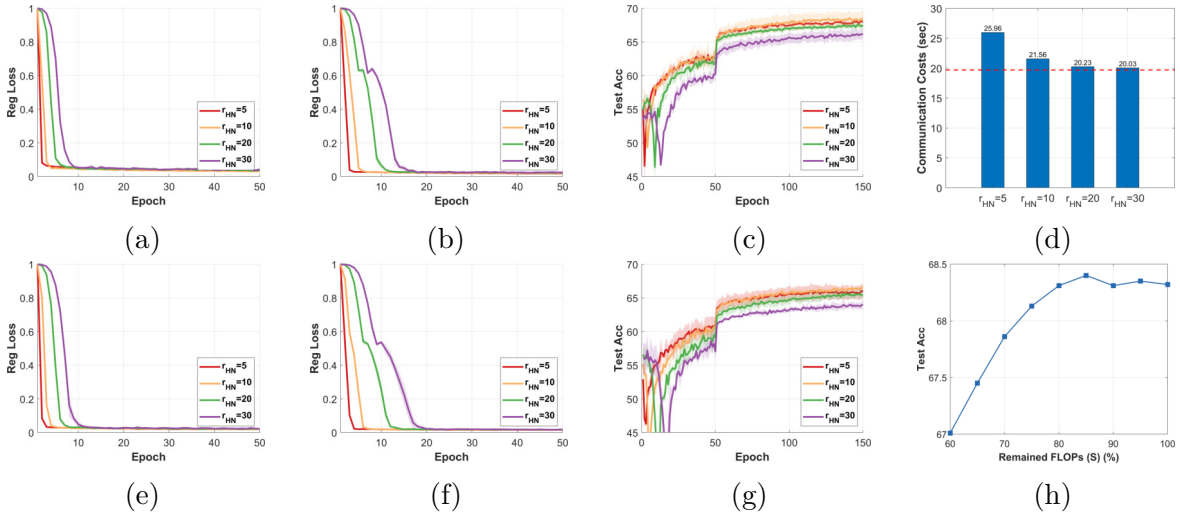


Figure 7.3: (a, e): normalized FLOPs regularization loss values for the server-side sub-network. (b, f): normalized FLOPs regularization loss values for device-side sub-networks. (d, h): test accuracy given different choices of r_{HN}. (d): communication costs. (h): trade-off between server-side and device-side sub-networks. Experiments are conducted on CIFAR-100 with ResNet-18 and when pruning 50% FLOPs (a,b,c) and 70% FLOPs (f,g,h) on devices.

7.4.2 Results

CIFAR-10/CIFAR-100. We tested different settings on CIFAR-10 and CIFAR-100 and found that our method DWNP consistently achieved the best performance across different model architectures and pruning rates. Specifically, DWNP outperformed the original model by 0.66%, 1.74%, 2.17%, and 1.46% for ResNet-56, ResNet-18, ResNet-34, and MobileNet-V2, respectively. This demonstrated that the design of device-wise sub-networks is beneficial for achieving a good trade-off for channel pruning under the federated learning setup. Our method even surpassed the original model when pruning 70% of FLOPs on ResNet-18. The relative ranking of other baselines is Filter Pruning, FedOSP, and FedILP. Our method achieved a prominent trade-off between performance and computational costs on more complex datasets, like CIFAR-100, with an improvement of 0.05%$\sim$2.17% over

Method	Architecture	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.05$	
		Base	Acc	Base	Acc	Base	Acc
FedILP	ResNet-18	66.57%	66.37% -0.20%	63.22%	62.77% -0.55%	61.61%	60.50% -1.11%
DWNP			68.31% **+1.74%**		64.51% **+1.29%**		62.59% **+0.98%**
FedILP	ResNet-34	69.05%	69.49% +0.44%	66.77%	66.46% -0.31%	64.52%	63.54% -0.98%
DWNP			71.72% **+2.17%**		68.20% **+1.43%**		65.53% **+1.01%**

Table 7.4: Performance of pruned models given different choices of α with ResNet-18/34 on CIFAR-100.

the original model when pruning 50% FLOPs or more. The performance of our method on MobileNet-V2 demonstrated that it could be seamlessly extended to lightweight models.

TinyImageNet. We present the results of ResNet-18 and ResNet-34 on TinyImageNet in Tab. 7.2. DWNP is 1.06%/1.10% better than the original model regarding the Top-1/5 accuracy for ResNet-18. For ResNet-34, the advantage is 0.80%/0.74% regarding the Top-1/5 accuracy. The advantage of our method compared to other baselines is still obvious, which ranges from 1.13% $\sim$ 2.07% and 0.45% $\sim$ 1.75% for Δ Top-1/5 accuracy for ResNet-18. We have similar observations for ResNet-34.

Across all settings, FedILP often performs better than FedOSP, indicating that learning model weights and architectures simultaneously are beneficial, as explained in section 7.3.6. In general, data-driven approaches perform better than pruning methods based on channel norms, suggesting that local data distributions should be considered explicitly when pruning under the FL setting. Indeed, the performance gain of DWNP is not free. For the server-side sub-network, the FLOPs reduction for DWNP is much smaller than other methods, and DWNP has to occupy more storage space on the server.

Other Settings. To verify whether our method can perform well in other settings, we change N and α to create different FL settings. In the first experiment, we use ResNet-

18 on CIFAR-100 to verify whether our method can achieve similar performance when changing the number of devices N. From Tab. 7.3, we can see the Δ-Acc is increased given more devices, which shows that our method is more resilient when increasing the number of devices N. On the other hand, the Δ-Acc of FedILP is similar or worse when increasing the number of devices, probably because the learning of the sub-network becomes harder when increasing N. In Tab. 7.4, we show the results when changing α on ResNet-18/34. A smaller α represents more diverse local data distributions and is often harder for model training. The table shows that both DWNP and FedILP are affected by decreasing α. However, DWNP can still maintain a positive performance gain for both ResNet-18/34. We plot the layer-wise pruning rate for channels with different α in Fig. 7.2. It can be seen that the sub-network architecture changes when changing local data distributions. For high heterogeneity ($\alpha = 0.05$), DWNP prefers to perverse more later layer channels, which is plausible because feature maps of later layers are more diverse on each device. In addition, the early stages of the model are not well utilized by device-side sub-networks. On the one hand, it is reasonable since CNNs tend to learn uniform representations from early stages. On the other hand, maybe we can add constraints to encourage the utilization of early stages or adjust the server-side sub-network so that it can be better used.

Detailed Analysis. We examine how r_{HN} changes the training dynamics during the optimization process. The training of model weights is not the focus of our paper, so we did not study $r_{\mathcal{W}}$. The effect of r_{θ} is not obvious compared to r_{HN}. We present our study in Fig. 7.3. We plot the first 50 epochs for regularization loss values after the training of HN begins. As described in the settings 7.4.1, the training of HN starts after 50 epochs of model weights training. We test 4 settings of r_{HN}: {5,10,20,30}. In short,

our method performs well when $r_{\text{HN}} \leq 10$. We can see an obvious performance drop when $r_{\text{HN}} = 30$. We also plot the overall communication costs for $\mathcal{W}$ and θ in Fig. 7.3d, and the red dashed line represents the costs for $\mathcal{W}$ only. For $r_{\text{HN}} \geq 10$, the communication overhead from training the HN becomes marginal. As a result, $r_{\text{HN}} = 10$ provides a good trade-off between performance and additional communication costs. In Fig. 7.3h, we show the trade-off between the model performance and the server-side FLOPs when pruning 50% of FLOPs on devices with ResNet-18 on CIFAR-100. We can see that our method can maintain a good performance when the remained server-side FLOPs are larger than 75%.

7.5 Proof of Theorems

In this section, we study the convergence rate of Algorithm 15. We first simplify the notations of objective functions: $\mathcal{L}^{(n)}(\mathcal{W};\theta) = \mathcal{L}(\mathcal{W}, D_n; \mathbf{a}^0 \odot \mathbf{a}^n)$ and

$$h^{(n)}(\mathcal{W},\theta) = \mathcal{L}(\mathcal{W}_\theta, D_n^{\mathbf{a}}; \mathbf{a}^0 \odot \mathbf{a}^n) + \lambda[(\mathcal{R}(T(\mathbf{a}^0), p_s T_{\text{total}}) + \mathcal{R}(T(\mathbf{a}^0 \odot \mathbf{a}^n), p_d^n T_{\text{total}})]$$

Next, we make some mild assumptions:

Assumption 153. *The function $h^{(n)}(\mathcal{W},\theta)$ is possibly non-convex, L-Lipschitz; $\mathcal{L}^{(n)}(\mathcal{W},\theta)$ is L-Lipschitz, μ-strongly convex* w.r.t. *$\mathcal{W}$ for any given θ;*

Assumption 154. *We have an unbiased stochastic first order derivative oracle with bounded variance σ^2;*

Assumption 155. *For any $m, j \in [N]$ and $z = (x, y)$, we have: $\|\nabla h^{(m)}(\mathcal{W},\theta) - \nabla h^{(j)}(\mathcal{W},\theta)\| \leq$*

ζ *and* $\|\nabla\mathcal{L}^{(m)}(\mathcal{W},\theta)-\nabla\mathcal{L}^{(j)}(\mathcal{W},\theta)\|\leq\bar{\zeta}$, *where* ζ, $\bar{\zeta}$ *are constants.*

We further assume the total number of update steps to $\mathcal{W}$ is K; the total number of update steps to θ is K_{HN}, and $K=r_{HN}K_{HN}$; the mini-batch size of stochastic query for both $\mathcal{W}$ and θ is b; the stochastic query to θ is denoted as G_k; we omit the gradient of $\mathcal{W}_\theta$ *w.r.t.* θ; we use $\bar{\theta}$ and $\bar{\mathcal{W}}$ to denotes the average of θ and $\mathcal{W}$ across devices. We denote:

$$\tilde{\mathcal{W}}_{\bar{\theta}}=\arg\min_{\mathcal{W}}\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}^{(n)}(\mathcal{W};\theta^{(n)}),\ \mathcal{W}_{\bar{\theta}}=\arg\min_{\mathcal{W}}\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}^{(n)}(\mathcal{W};\bar{\theta}).$$

since we make infrequent updates to θ, we omit the drift of the lower level solution caused by θ update, in other words, we assume $\tilde{\mathcal{W}}_{\bar{\theta}}\approx\mathcal{W}_{\bar{\theta}}$.

7.5.1 Useful Lemmas

Lemma 156. *For* $k\in[r_{HN}k_{HN},r_{HN}k_{HN}+r_{HN}-1]$, *we have:*

$$\mathbb{E}\left\|\bar{\mathcal{W}}_{r_{HN}k_{HN}+r_{HN}}-\mathcal{W}_{\bar{\theta}_{k_{HN}}}\right\|^2\leq(1-\lambda\gamma)^{r_{HN}}\mathbb{E}\left\|\bar{\mathcal{W}}_{r_{HN}k_{HN}}-\mathcal{W}_{\bar{\theta}_{k_{HN}}}\right\|^2 + \frac{1}{\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. For ease of notation, we denote $\mathcal{W}^*=\mathcal{W}_{\bar{\theta}_{k_{HN}}}$ in the proof, and follow Lemma 15 and Lemma 16 in [216], for $k\in[r_{HN}k_{HN},r_{HN}k_{HN}+r_{HN}-1]$, we have:

$$\mathbb{E}\left\|\bar{\mathcal{W}}_{k+1}-\mathcal{W}^*\right\|^2\leq(1-\lambda\gamma)\mathbb{E}\left\|\bar{\mathcal{W}}_k-\mathcal{W}^*\right\|^2+\frac{\gamma^2\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^3+12Lr_{\mathcal{W}}^2\bar{\zeta}^2\gamma^3$$

Telescope for $k \in [r_{HN}k_{HN}, r_{HN}k_{HN} + r_{HN} - 1]$, we have:

$$\begin{aligned}
&\mathbb{E}\left\|\bar{\mathcal{W}}_{r_{HN}k_{HN}+r_{HN}} - \mathcal{W}^*\right\|^2 \\
&\leq (1-\lambda\gamma)^{r_{HN}}\mathbb{E}\left\|\bar{\mathcal{W}}_{r_{HN}k_{HN}} - \mathcal{W}^*\right\|^2 \\
&\quad + \sum_{k=r_{HN}k_{HN}}^{r_{HN}k_{HN}+r_{HN}} (1-\lambda\gamma)^{k-r_{HN}k_{HN}}\left(\frac{\gamma^2\sigma^2}{N} + 12Lr_{\mathcal{W}}\sigma^2\gamma^3 + 12Lr_{\mathcal{W}}^2\gamma^3\bar{\zeta}^2\right) \\
&\leq (1-\lambda\gamma)^{r_{HN}}\mathbb{E}\left\|\bar{\mathcal{W}}_{r_{HN}k_{HN}} - \mathcal{W}^*\right\|^2 + \frac{1}{\lambda\gamma}\left(\frac{\gamma^2\sigma^2}{bN} + 12Lr_{\mathcal{W}}\sigma^2\gamma^3 + 12Lr_{\mathcal{W}}^2\gamma^3\bar{\zeta}^2\right)
\end{aligned}$$

This completes the proof. □

Lemma 157. *Suppose iterates $\theta_k^{(n)}$, $n \in [N]$, $k \in [K_{HN}]$ are generated from Algorithm 15, we have:*

$$\begin{aligned}
&(1 - 12L^2r_\theta^2\eta^2)\sum_{k=\tilde{k}_r}^{\tilde{k}_r+r_\theta-1}\sum_{n=1}^{N}\mathbb{E}\|\bar{\theta}_k - \theta_k^{(n)}\|^2 \\
&\leq Nr_\theta^3\eta^2\left(\frac{2\sigma^2}{b} + 8\zeta^2 + \frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN} + 12Lr_{\mathcal{W}}\sigma^2\gamma^2 + 12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)\right)
\end{aligned}$$

where $\tilde{k}_r \in [K_{HN}]$ and $\tilde{k}_r \% r_\theta = 0$, and the expectation is w.r.t the stochasticity of the algorithm.

Proof. Suppose we denote $\tilde{k}_r \in [K_{HN}]$ where $\tilde{k}_r \% r_\theta = 0$. By Algorithm 15, we have $\theta_{\tilde{k}_r}^{(n)} = \bar{\theta}_{\tilde{k}_r}$. For $k \neq \tilde{k}_r$, we have:

$$\theta_k^{(n)} = \theta_{\tilde{k}_r}^{(n)} - \sum_{\ell=\tilde{k}_r}^{k-1}\eta G_\ell^{(n)} \quad \text{and} \quad \bar{\theta}_k = \bar{\theta}_{\tilde{k}_r} - \sum_{\ell=\tilde{k}_r}^{k-1}\eta\bar{G}_\ell.$$

So we have:

$$\sum_{n=1}^{N}\|\theta_k^{(n)}-\bar{\theta}_k\|^2=\sum_{n=1}^{N}\Big\|\sum_{\ell=\tilde{k}_r}^{k-1}\Big(\eta G_\ell^{(n)}-\eta\bar{G}_\ell\Big)\Big\|^2\le r_\theta\sum_{\ell=\tilde{k}_r}^{k-1}\eta^2\sum_{n=1}^{N}\|G_\ell^{(n)}-\bar{G}_\ell\|^2$$

where the equality uses the fact $\theta_{\tilde{k}_r}^{(n)}=\bar{\theta}_{\tilde{k}_r}$ for $k\in[K]$, the inequality uses the generalized triangle inequality. Next,

$$\begin{aligned}
&\mathbb{E}\Big\|G_\ell^{(n)}-\bar{G}_\ell\Big\|^2\\
&=\mathbb{E}\Big\|\Big(G_\ell^{(n)}-\mathbb{E}_\xi[G_\ell^{(n)}]\Big)-\frac{1}{N}\sum_{j=1}^{N}\Big(G_\ell^{(j)}-\mathbb{E}_\xi[G_\ell^{(j)}]\Big)+\mathbb{E}_\xi[G_\ell^{(n)}]-\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_\xi[G_\ell^{(j)}]\Big\|^2\\
&\le 2\mathbb{E}\Big\|\Big(G_\ell^{(n)}-\mathbb{E}_\xi[G_\ell^{(n)}]\Big)-\frac{1}{N}\sum_{j=1}^{N}\Big(G_\ell^{(j)}-\mathbb{E}_\xi[G_\ell^{(j)}]\Big)\Big\|^2+2\mathbb{E}\Big\|\mathbb{E}_\xi[G_\ell^{(n)}]-\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_\xi[G_\ell^{(j)}]\Big\|^2\\
&\overset{(a)}{\le} 2\mathbb{E}\Big\|\Big(G_\ell^{(n)}-\mathbb{E}_\xi[G_\ell^{(n)}]\Big)\Big\|^2+2\mathbb{E}\Big\|\mathbb{E}_\xi[G_\ell^{(n)}]-\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_\xi[G_\ell^{(j)}]\Big\|^2\\
&\le 2\mathbb{E}\Big\|\Big(G_\ell^{(n)}-\mathbb{E}_\xi[G_\ell^{(n)}]\Big)\Big\|^2+4\mathbb{E}\Big\|\mathbb{E}_\xi[G_\ell^{(n)}]-\nabla h^{(n)}(\bar{\theta}_\ell)\Big\|^2\\
&\qquad+8\mathbb{E}\Big\|\nabla h^{(n)}(\bar{\theta}_\ell)-\nabla h(\bar{\theta}_\ell)\Big\|^2+8\mathbb{E}\Big\|\nabla h(\bar{\theta}_\ell)-\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_\xi[G_\ell^{(j)}]\Big\|^2\\
&\overset{(b)}{\le}\frac{2\sigma^2}{b}+4L^2\Big(\mathbb{E}\|\bar{\theta}_\ell-\theta_\ell^{(n)}\|^2+\mathbb{E}\|\mathcal{W}_{\bar{\theta}_\ell}-\mathcal{W}_\ell^{(n)}\|^2\Big)\\
&\qquad+8\zeta^2+\frac{8L^2}{N}\sum_{j=1}^{N}\Big(\mathbb{E}\|\bar{\theta}_\ell-\theta_\ell^{(j)}\|^2+\mathbb{E}\|\mathcal{W}_{\bar{\theta}_\ell}-\mathcal{W}_\ell^{(j)}\|^2\Big)
\end{aligned}\tag{7.9}$$

Average over $n\in[N]$, we have:

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\Big\|G_\ell^{(n)}-\bar{G}_\ell\Big\|^2\le\frac{2\sigma^2}{b}+8\zeta^2+\frac{12L^2}{N}\sum_{n=1}^{N}\Big(\mathbb{E}\|\bar{\theta}_\ell-\theta_\ell^{(n)}\|^2+\mathbb{E}\|\mathcal{W}_{\bar{\theta}_\ell}-\mathcal{W}_\ell^{(n)}\|^2\Big)$$

So we have:

$$\sum_{n=1}^{N}\|\theta_k^{(n)}-\bar{\theta}_k\|^2 \le r_\theta \sum_{\ell=\tilde{k}_r}^{k-1}\eta^2\sum_{n=1}^{N}\|G_\ell^{(n)}-\bar{G}_\ell\|^2$$
$$\le N r_\theta \sum_{\ell=\tilde{k}_r}^{k-1}\eta^2\left(\frac{2\sigma^2}{b}+8\zeta^2+\frac{12L^2}{N}\sum_{n=1}^{N}\left(\mathbb{E}\|\bar{\theta}_\ell-\theta_\ell^{(n)}\|^2+\mathbb{E}\|\mathcal{W}_{\bar{\theta}_\ell}-\mathcal{W}_\ell^{(n)}\|^2\right)\right)$$

Then we combine with Lemma 156 to get:

$$\sum_{n=1}^{N}\|\theta_k^{(n)}-\bar{\theta}_k\|^2$$
$$\le N r_\theta \sum_{\ell=\tilde{k}_r}^{k-1}\eta^2\left(\frac{2\sigma^2}{b}+8\zeta^2\right.$$
$$\left.+\frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{N}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)+\frac{12L^2}{N}\sum_{n=1}^{N}\left(\mathbb{E}\|\bar{\theta}_\ell-\theta_\ell^{(n)}\|^2\right)\right)$$
$$\le N r_\theta^2\eta^2\left(\frac{2\sigma^2}{b}+8\zeta^2+\frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)\right)$$
$$+12L^2r_\theta\eta^2\sum_{\ell=\tilde{k}_r}^{k-1}\sum_{n=1}^{N}\mathbb{E}\|\bar{\theta}_\ell-\theta_\ell^{(n)}\|^2$$

Next, we sum over k, we have:

$$(1-12L^2r_\theta^2\eta^2)\sum_{k=\tilde{k}_r}^{\tilde{k}_r+r_\theta-1}\sum_{n=1}^{N}\mathbb{E}\|\bar{\theta}_k-\theta_k^{(n)}\|^2$$
$$\le N r_\theta^3\eta^2\left(\frac{2\sigma^2}{b}+8\zeta^2+\frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)\right)$$

This completes the proof. □

7.5.2 Descent Lemma

Lemma 158. *Suppose $\eta\eta_g \leq \frac{1}{2I\bar{L}}$ For $t \in [T]$, the iterates generated satisfy:*

$$\mathbb{E}[h(\bar{\theta}_{k_{HN+1}})] \leq \mathbb{E}[h(\bar{\theta}_{k_{HN}})] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 - \frac{\eta}{4}\mathbb{E}\|\mathbb{E}_\xi[\bar{G}_{k_{HN}}]\|^2 + \frac{L^2\eta}{2N}\sum_{n=1}^{N}\left(\mathbb{E}\|\bar{\theta}_{k_{HN}} - \theta^{(n)}_{k_{HN}}\|^2 + \mathbb{E}\|\mathcal{W}_{\bar{\theta}_{k_{HN}}} - \bar{\mathcal{W}}_{(r_{HN}+1)k_{HN}}\|^2\right) + \frac{\eta^2 L\sigma^2}{2bN}$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Using the smoothness of h we have:

$$\begin{aligned}
\mathbb{E}[h(\bar{\theta}_{k_{HN+1}})] &\leq \mathbb{E}[h(\bar{\theta}_{k_{HN}})] + \mathbb{E}\langle\nabla h(\bar{\theta}_{k_{HN}}), \bar{\theta}_{k_{HN+1}} - \bar{\theta}_{k_{HN}}\rangle + \frac{L}{2}\mathbb{E}\|\bar{\theta}_{k_{HN+1}} - \bar{\theta}_{k_{HN}}\|^2 \\
&\overset{(a)}{=} \mathbb{E}[h(\bar{\theta}_{k_{HN}})] - \eta\mathbb{E}\langle\nabla h(\bar{\theta}_{k_{HN}}), \mathbb{E}_\xi[\bar{G}_{k_{HN}}]\rangle + \frac{\eta^2 L}{2}\mathbb{E}\|\mathbb{E}_\xi[\bar{G}_{k_{HN}}]\|^2 + \frac{\eta^2 L\sigma^2}{2bN} \\
&\overset{(b)}{=} \mathbb{E}[h(\bar{\theta}_{k_{HN}})] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}}) - \mathbb{E}_\xi[\bar{G}_{k_{HN}}]\|^2 \\
&\quad - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right)\mathbb{E}\|\mathbb{E}_\xi[\bar{G}_{k_{HN}}]\|^2 + \frac{\eta^2 L\sigma^2}{2bN} \\
&\overset{(c)}{\leq} \mathbb{E}[h(\bar{\theta}_{k_{HN}})] - \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 - \frac{\eta}{4}\mathbb{E}\|\mathbb{E}_\xi[\bar{G}_{k_{HN}}]\|^2 \\
&\quad + \frac{L^2\eta}{2N}\sum_{n=1}^{N}\left(\mathbb{E}\|\bar{\theta}_{k_{HN}} - \theta^{(n)}_{k_{HN}}\|^2 + \mathbb{E}\|\mathcal{W}_{\bar{\theta}_{k_{HN}}} - \bar{\mathcal{W}}_{(r_{HN}+1)k_{HN}}\|^2\right) + \frac{\eta^2 L\sigma^2}{2bN}
\end{aligned}$$

where equality (a) follows from the update step to the HN in Algorithm 15; (b) uses $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a - b\|^2]$; (c) follows the condition that $\eta_k \leq 1/(2L)$ and the smoothness of $h(\theta)$. This completes the proof. □

7.5.3 Proof of Convergence Theorem

We are ready to prove the main theorem 152 in this subsection.

Theorem 159. *Suppose we the upper level learning rate η and the lower level learning rate γ as:*

$$\eta = \min\left\{\frac{1}{4Lr_\theta}, \left(\frac{2bN\Delta_\theta}{K_{HN}L\sigma^2}\right)^{1/2}\right\}, \quad \gamma = \min\{\frac{1}{4L}, \left(\frac{\lambda bN}{K_{HN}L^2\sigma^2}\right)^{1/2}\}$$

then we have:

$$\frac{1}{K_{HN}}\sum_{k_{HN}=0}^{K_{HN}-1}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 \leq \left(\frac{2L\sigma^2\Delta_\theta}{bNK_{HN}}\right)^{1/2} + \left(\frac{L^2\sigma^2}{\lambda bNK_{HN}}\right)^{1/2}$$

where b is the mini-batch size, N is the number of devices

Proof. Combine Lemma 158, and Lemma 156 to have:

$$\begin{aligned}\mathbb{E}[h(\bar{\theta}_{k_{HN}+1})] \leq \mathbb{E}[h(\bar{\theta}_{k_{HN}})] &- \frac{\eta}{2}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 - \frac{\eta}{4}\mathbb{E}\|\mathbb{E}_\xi[\bar{G}_{k_{HN}}]\|^2 \\ &+ \frac{L^2\eta}{2N}\sum_{n=1}^{N}\mathbb{E}\|\bar{\theta}_{k_{HN}} - \theta_{k_{HN}}^{(n)}\|^2 + \frac{L^2\eta}{2}(1-\lambda\gamma)^{r_{HN}}\Delta_{\mathcal{W}} \\ &+ \frac{L^2\eta}{2\lambda}\left(\frac{\gamma\sigma^2}{bN} + 12Lr_{\mathcal{W}}\sigma^2\gamma^2 + 12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right) + \frac{\eta^2L\sigma^2}{2bN}\end{aligned}$$

Next, we sum over k_{HN}, and denote $\Delta_\theta = h(\theta_0)$, we have:

$$\begin{aligned}\frac{\eta}{2}\sum_{k_{HN}=0}^{K_{HN}-1}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 \leq \Delta_\theta &+ \frac{L^2\eta}{2N}\sum_{k_{HN}=0}^{K_{HN}-1}\sum_{n=1}^{N}\mathbb{E}\|\bar{\theta}_{k_{HN}} - \theta_{k_{HN}}^{(n)}\|^2 + \frac{K_{HN}L^2\eta}{2}(1-\lambda\gamma)^{r_{HN}}\Delta_{\mathcal{W}} \\ &+ \frac{K_{HN}L^2\eta}{2\lambda}\left(\frac{\gamma\sigma^2}{bN} + 12Lr_{\mathcal{W}}\sigma^2\gamma^2 + 12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right) + \frac{K_{HN}\eta^2L\sigma^2}{2bN}\end{aligned}$$

Suppose we choose $\eta < \frac{1}{4Lr_\theta}$. By Lemma 157, we have:

$$\sum_{k=\tilde{k}_r}^{\tilde{k}_r+r_\theta-1}\sum_{n=1}^{N}\mathbb{E}\|\bar{\theta}_k - \theta_k^{(n)}\|^2 \leq 4Nr_\theta^3\eta^2\left(\frac{2\sigma^2}{b} + 8\zeta^2 + \frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN} + 12Lr_{\mathcal{W}}\sigma^2\gamma^2 + 12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)\right)$$

So we have:

$$\begin{aligned}
&\frac{\eta}{2}\sum_{k_{HN}=0}^{K_{HN}-1}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 \\
&\leq \Delta_\theta + 2K_{HN}L^2 r_\theta^2\eta^3\left(\frac{2\sigma^2}{b}+8\zeta^2+\frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)\right) \\
&\quad + \frac{K_{HN}L^2\eta}{2}(1-\lambda\gamma)^{r_{HN}}\Delta_{\mathcal{W}} \\
&\quad + \frac{K_{HN}L^2\eta}{2\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)+\frac{K_{HN}\eta^2L\sigma^2}{2bN}
\end{aligned}$$

Multiply $\frac{2}{\eta K_{HN}}$ on both sides, we have:

$$\begin{aligned}
&\frac{1}{K_{HN}}\sum_{k_{HN}=0}^{K_{HN}-1}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 \\
&\leq \frac{2\Delta_\theta}{\eta K_{HN}} + 4L^2 r_\theta^2\eta^2\left(\frac{2\sigma^2}{b}+8\zeta^2+\frac{12L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)\right) \\
&\quad + L^2(1-\lambda\gamma)^{r_{HN}}\Delta_{\mathcal{W}} \\
&\quad + \frac{L^2}{\lambda}\left(\frac{\gamma\sigma^2}{bN}+12Lr_{\mathcal{W}}\sigma^2\gamma^2+12Lr_{\mathcal{W}}^2\gamma^2\bar{\zeta}^2\right)+\frac{\eta L\sigma^2}{bN}
\end{aligned}$$

By ignoring the higher order terms, we have:

$$\frac{1}{K_{HN}}\sum_{k_{HN}=0}^{K_{HN}-1}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 \leq \frac{2\Delta_\theta}{\eta K_{HN}}+\frac{\gamma L^2\sigma^2}{\lambda bN}+\frac{\eta L\sigma^2}{bN}$$

By choosing:

$$\eta=\min\left\{\frac{1}{4Lr_\theta},\left(\frac{2bN\Delta_\theta}{K_{HN}L\sigma^2}\right)^{1/2}\right\},\ \ \gamma=\min\{\frac{1}{4L},\left(\frac{\lambda bN}{K_{HN}L^2\sigma^2}\right)^{1/2}\}$$

then we have:

$$\frac{1}{K_{HN}}\sum_{k_{HN}=0}^{K_{HN}-1}\mathbb{E}\|\nabla h(\bar{\theta}_{k_{HN}})\|^2 \leq \left(\frac{2L\sigma^2\Delta_\theta}{bNK_{HN}}\right)^{1/2} + \left(\frac{L^2\sigma^2}{\lambda bNK_{HN}}\right)^{1/2}$$

this completes the proof. □

7.6 More Experimental Details

7.6.1 Experimental Settings

In Tab. 7.6, we provide the details choices of p_s and p_d^n. When training the base model, we use SGD as the local optimizer with a start learning rate of 0.1, momentum of 0.9, and weight decay of 0.0001. The learning rate is decayed to 0.01 and 0.001 at epochs 50 and 100, respectively. We set the mini-batchsize on each device to be $\lfloor 256/N \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function. For FedOSP, we also use the hypernetwork for pruning, and the pruning process is conducted on the subset $D_n^{\mathbf{a}}$. We train the hypernetwork for FedOSP for 200 epochs with ADAM and a constant learning rate of 0.001. With these settings, the additional costs for training the hypernetwork for FedOSP and other methods will be similar. During the finetuning process, we use exactly the same setting as the base model training process.

We assume the local data distribution on each device is similar, and we split the test dataset to each device according to this assumption. In [214], for each class, they sample a vector $v \sim \text{Dir}(\alpha)$, where r is a N (the number of devices) dimensional vector which lies

Inputs $n = 0, \cdots, N$,
Embedding(N, 32× L), Resize to (L,32)
GRU(32, 64), LayerNorm, ReLU FC$_l$(64,C_l), LayerNorm, $l = 1, \cdots, L$
Outputs $\bar{\mathbf{a}}_l$, $l = 1, \cdots, L$

Table 7.5: The architecture of the embedding and the hypernetwork used in our method.

in a simplex ($\sum_{n=1}^{N} v_n = 1$). For each class, we modify this vector by adding random noise:

$$\bar{v} = v + r, \ r \sim \text{Uniform}(-\beta, \beta). \tag{7.10}$$

To satisfy the simplex requirement, we produce the final by using $\hat{v} = \frac{\bar{v}}{\sum_{n=1}^{N} \bar{v}_n}$. In experiments to satisfy our assumption, we let $\beta = \frac{1}{10N}$. We then assign test samples to each device based on $\hat{v}$.

7.6.2 Details of the Hypernetwork and Embedding

The detailed architecture of the embedding and the hypernetwork is shown in Tab. 7.5. GRU [217] is used to capture inter-layer relationships and fully connected layers are used to capture inter-channel interactions.

After we have the outputs $\bar{\mathbf{a}}_l$ from the hypernetwork, we calculate the binary vectors $\mathbf{a}_l$ by using the following equations:

$$\mathbf{a}_l = \text{round}(\text{sigmoid}((\bar{\mathbf{a}}_l + g + c)/\tau)), \tag{7.11}$$

where $g \sim \text{Gumbel}(0, 1)$, and Gumbel is the Gumbel distribution, and c is a constant to make the pruning starts with the whole network. In practice, we set $c = 3.0$ for all

experiments. τ is the temperature parameter, which is set to 0.4 for all experiments.

Dataset	Architecture	p_d^n	p_s
CIFAR-10	ResNet-56	0.50	0.80
CIFAR-100	ResNet-18	0.50/0.30	0.80/0.60
	ResNet-34	0.50/0.30	0.80/0.60
	MobileNet-V2	0.50	0.80
TinyImageNet	ResNet-18	0.50	0.80
	ResNet-34	0.50	0.80

Table 7.6: Choice of p_r and p_p.

7.6.3 Ablation Study

Settings	Base Acc	Δ-Acc	Acc	↓ FLOPs (D)	↓ FLOPs (S)
FedILP	66.57%	-0.20%	66.37%	50%	50%
DWNP (OS)		+1.02%	67.59%	50%	50%
DWNP w/o Eq. 7		+1.43%	68.00%	50%	50%
DWNP		**+1.74%**	**68.31%**	50%	20%

Table 7.7: Results of CIFAR-100 with different settings.

In Tab. 7.7, we add more settings to study the effects of the design choices. 'DWNP (OS)' corresponds to one-shot pruning for server and device sub-networks. 'DWNP w/o Eq. 7' corresponds to not resolving the conflicts between server and device sub-networks. The one-shot setting results in 0.72% performance loss, which suggests that co-training indeed produces a better final model, as we discussed in section 3.6. The difference between 'DWNP w/o Eq. 7' and 'DWNP' is not large, but the cost of adding Eq. 7 is minimal. Thus, the performance gain is nearly free, and it also suggests that restricting the search space of the device-wise sub-networks is helpful.

Method	Architectures	Time (sec/img)	↓ Time	↓ FLOPs (D)	↓ FLOPs (S)
Original	ResNet-18	0.0559	-	-	-
Filter Pruning		0.0348	37.7%	50%	50%
FedOSP		0.0336	39.9%	50%	50%
FedILP		0.0345	38.3%	50%	50%
DWNP		0.0340	39.2%	50%	20%

Table 7.8: Inference Time Comparison on TinyImageNet.

7.6.4 Control the Relationship between the Server-side and Device-side Sub-networks

To better describe the relationship between the server-side sub-network $\mathbf{a^0}$ and device-side sub-networks $\mathbf{a^n}$, we explicitly require that if a channel is pruned by the server-side sub-network, the corresponding device-side sub-networks should not update the corresponding position. This is achieved by stopping gradients if the condition is met:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{a}^n[i]} = 0, \text{ if } i \in \text{Ind}(a^0), \tag{7.12}$$

where we define the index set $\text{Ind}(a^0)$ to be $\text{Ind}(a^0) = \{i \mid \mathbf{a^0}[i] = 0\}$, which is used to represent the index of pruned positions given the server-side sub-network. This constraint is dynamically changed during the learning process of sub-networks, but it will become stable in the middle to the late training stage. The benefits of this constraint are two-fold. Firstly, it implicitly embeds the relationship between the server-side sub-network and device-side sub-networks. Secondly, it reduces the potential search space for device-side sub-networks, which is beneficial for learning them.

Architecture	Method	Base Top-1 Acc	Δ Top-1 Acc	Acc	↓ FLOPs (D)	↓ FLOPs (S)
ResNet-18	Filter Pruning	73.52%	-1.28%	72.24%	50%	50%
	FedOSP		-0.32%	73.20%	50%	50%
	FedILP		-0.04%	73.48%	50%	50%
	DWNP		**+1.42%**	74.94%	50%	20%

Table 7.9: Comparison results on ImageNet-100 with ResNet-18.

7.6.5 ImageNet-100 Results

We use the ImageNet-100 subset to further evaluate our method following the partition shown in [218]. For ImageNet-100, We follow the same training setting of other datasets described in section 4.1, except that we reduced the training and finetuning epochs to 80 to save computational costs. We use 8 clients with $\alpha = 0.5$ on ImageNet-100. The comparison result is shown in Tab. 7.9. DWNP still consistently outperforms other baselines like other datasets. This experiment further demonstrates that our method can be easily scaled up to larger (higher resolution/more samples) datasets.

7.6.6 Inference Time Comparasion

We measure the inference time on the CPU with ResNet-18 on tiny-ImageNet in Tab. 7.8. The result is obtained by averaging inference time from 1000 input samples. For DWNP, the time is further averaged from 10 device-side sub-networks, and the range of inference time for DWNP is $0.0336 \sim 0.0351$ (sec/img). From the table, we can see that different methods have comparable acceleration rates giving similar FLOPs on each device.

Chapter 8: FedDA: Faster Adaptive Gradient Methods for Federated Constrained Optimization

8.1 Introduction

As an emerging machine learning technique, federated learning (FL) has recently been applied to many important health and biomedicine applications [219, 220, 221, 222, 223]. The FL enables big data analyses in healthcare (especially for rare diseases) via allowing a set of distributed located hospitals, medical centers, insurance companies, *etc.*, to jointly perform a machine learning task under the coordination of a central server over their privately-held data. For example, in a recent FL study, the data from 71 sites across 6 continents are used to generate an automatic tumor boundary detector for the rare disease of glioblastoma [224]. Among these FL healthcare applications, federated biomarker identification [225] is one of the most important learning tasks to help researchers and clinicians detect informative biomarkers from distributed biomedical datasets to understand the underlying disease mechanisms, diagnose disease earlier, and design drugs. Different to prediction tasks, the federated biomarker identification often utilizes the constrained formulations, such as Lasso and group Lasso, which often requires specific optimization solvers.

A widely used method in Federated Learning (FL) is the FedAvg (Local-SGD) algorithm [92]. As indicated by its name, FedAvg performs (stochastic) gradient descent steps on each client and averages local states periodically. Recently, researchers incorporated adaptive gradient methods to the FL setting [208, 226, 227] to accelerate FedAvg. For instance, [228] introduced FedAdam, which incorporates the Adam optimizer for server updates; [208] proposed MIME, which supports adaptive gradients in local updates. These adaptive gradient methods have demonstrated significant performance enhancements compared to FedAvg. However, these methods are designed for the unconstrained setting and cannot be directly applied over the constrained formulations as in the biomarker identification tasks. In fact, FL problems with constraints remain under-explored in the literature. In [229], authors proposed FedDualAvg to solve FL problems with non-smooth regularizers. In FedDualAvg, model parameters are transformed into a dual space using a map determined by the regularizer. Averaging is then conducted in this dual space across all clients. FedDualAvg can be used to solve federated constrained optimization problems when regularizers are indicator functions defined over the constraints set. However, FedDualAvg uses constant learning rate as FedAvg, thus suffers the same slow convergence rate. In fact, it is an open question *if adaptive learning rates can be used to accelerate the solving of federated constrained optimizations.*

In this work, we propose the **Fed**erated **D**ual-averaging **A**daptive-gradient (**FedDA**), a general adaptive gradient framework for federated constrained optimization. FedDA is based on the dynamic mirror descent view of adaptive gradients [31]. In fact, suppose we have an adaptive matrix H such that it is diagonal and its i_{th} diagonal element represents the adaptive gradient for the i_{th} coordinate of the model parameter. Then if we view the

Table 8.1: **Comparisons of representative Federated Learning algorithms for finding an ϵ-stationary point** *i.e.*, $\|\nabla f(x)\|^2 \leq \epsilon$ or its equivalent variants. $Gc(f,\epsilon)$ denotes the number of gradient queries *w.r.t.* $f^{(k)}(x)$ for $k \in [K]$; $Cc(f,\epsilon)$ denotes the number of communication rounds; **State** means what state the algorithm maintains locally (Primal/Dual); **Local-Adaptive** means whether the algorithm performs adaptive gradient descent locally or not; **Constrained** means whether the algorithm can solve both constrained and unconstrained problems or not.

Type	Algorithm	$Gc(f,\epsilon)$	$Cc(f,\epsilon)$	State	Local-Adaptive	Constrained
Non-adaptive	FedAvg [92]	$O(K^{-1}\epsilon^{-2})$	$O(\epsilon^{-1.5})$	Primal/Dual	×	×
	FedDualAvg [229]	$O(K^{-1}\epsilon^{-2})$	$O(K^{-1}\epsilon^{-2})$	Dual	×	✓
	FedCM [230]	$\tilde{O}(K^{-1}\epsilon^{-2})$	$\tilde{O}(K^{-1}\epsilon^{-2})$	Primal/Dual	×	×
	FedGLOMO [231]	$\tilde{O}(K^{-0.5}\epsilon^{-1.5})$	$\tilde{O}(\epsilon^{-1.5})$	Primal/Dual	×	×
	STEM [119]	$\tilde{O}(K^{-1}\epsilon^{-1.5})$	$\tilde{O}(\epsilon^{-1})$	Primal/Dual	×	×
Adaptive	FedAdam [226]	$O(K^{-1}\epsilon^{-2})$	$O(K^{-0.5}\epsilon^{-1})$	Primal	×	×
	Local-AMSGrad [227]	$O(K^{-1}\epsilon^{-2})$	$O(K^{-1}\epsilon^{-2})$	Primal	✓	×
	MIME-MVR [208]	$\tilde{O}(K^{-0.5}\epsilon^{-1.5})$	$O(K^{-0.5}\epsilon^{-1.5})$	Primal	✓	×
	FAFED [232]	$\tilde{O}(K^{-1}\epsilon^{-1.5})$	$\tilde{O}(\epsilon^{-1})$	Primal	✓	×
	FedDA-MVR(Ours)	$\tilde{O}(K^{-1}\epsilon^{-1.5})$	$\tilde{O}(K^{-0.25}\epsilon^{-1.25})$	Dual	✓	✓

parameter space as the primal space and the matrix H as a linear mirror map, the adaptive gradient update step is equivalent to a mirror descent update step. Since adaptive gradients are updated at each iteration, the mirror map H is also dynamic. Our FedDA is inspired by this mirror descent view of adaptive gradients.

In FedDA, the server maintains the global adaptive matrix and the global model weight. In each epoch, each client receives the current global adaptive matrix and global model weight. Then it performs multiple steps of 'dual-averaging' style mirror descent [233]. More specifically, the client aggregates dual states, and only recover model weight (primal states) through the adaptive matrix (mirror map) when the gradient query is needed. Note that the adaptive matrix is fixed during local updates. After local updates, the server averages the dual states and then updates the global primal state and the adaptive matrix. There are two important characteristics of FedDA. Firstly, we fix the adaptive matrix during local updates to makes sure that all clients share the same dual space. Secondly, the aggregation and averaging is performed in the dual space instead of the primal space.

In fact, in the constrained case, the mapping from dual space to the primal space involves the non-linear projection operation. Although averaging dual states leads to an unbiased estimate of the global gradient, averaging primal states leads to biased estimation due to projection. In summary, FedDA adopts the restarted dual-averaging strategy, where the adaptive matrix (mirror map) is refreshed at each global epoch and clients perform dual averaging locally with a fixed mirror map. Our **FedDA** is a general framework: it is flexible to the choices of adaptive gradient methods and also can combine with various gradient estimation methods. Furthermore, although it is designed for the constrained case, FedDA works well in the unconstrained FL setting too. We highlight our **contribution** as follows:

(i) We propose **FedDA**, a fast adaptive gradient framework for constrained federated optimization problems. The framework is based on a restarted dual averaging technique and incorporates a large family of adaptive gradient methods.

(ii) **FedDA-MVR**, an instantiation of our framework, obtains the sample complexity of $\tilde{O}(K^{-1}\epsilon^{-1.5})$ and communication complexity of $\tilde{O}(K^{-0.25}\epsilon^{-1.25})$. The iteration complexity matches the optimal rate of non-adaptive federated algorithms. **FedDA-MVR** uses the momentum-based variance-reduction gradient estimation, and exponential moving average of the gradient square as adaptive learning rates.

(iii) We empirically verify the efficacy of the FedDA in solving biomarker identification tasks: the colorrectal cancer prediction task and the splice site detection task. Furthermore, we also verify the efficacy of FedDA over non-constrained FL tasks through classification tasks over CIFAR-10 and FEMNIST datasets.

Notations. $\nabla f(x)$ denotes the first-order derivatives of the function $f(x)$ *w.r.t.*

variable x. ξ denotes a random sample and $\nabla f(x;\xi)$ is the stochastic estimate $\nabla f(x)$. $O(\cdot)$ is the big O notation, and $\tilde{O}(\cdot)$ hides logarithmic terms. I_d denotes a d-dimensional identity matrix. $Diag(x)$ denotes the matrix whose diagonal is the vector x. $\|\cdot\|$ denotes the ℓ_2 norm for vectors and the spectral norm for matrices, respectively. $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product. [K] denotes the set of $\{1, 2, ..., K\}$. For a random variable X, $\mathbb{E}[X]$ denotes its expectation.

8.2 Related Works

Federated Learning. Federated Learning [92] defines the task to learn from a set of distributed located clients under the coordination of a server. [92] proposed the FedAvg algorithm where each client performs multiple steps of gradient descent with its local data and then sends the updated model to the server for averaging. The FedAvg algorithm resembles the Local-SGD algorithm, which is studied in a more general distributed setting for a longer time [234]. The convergence of the local-SGD method has been heavily analyzed in the literature [94, 118, 127, 143, 216, 235, 236, 237]. Various acceleration methods of FedAvg are considered and we list a few representatives here. [55] adopted the idea of variance reduction technique for non-distributed finite sum problems: a 'control variate' which contains historical full gradient information is used to correct the bias of local gradients. [208] proposed a general framework (MIME) to translate a centralized optimizer into the FL setting, including adaptive gradient methods. In [119, 231], momentum-based variance reduction is applied to the FL setting to control the noise of the stochastic gradients. In [231], a server momentum state and a client momentum state are maintained, while in

[13], a momentum state and is matinained the momentum was averaged periodically similar to the primal state.

Adaptive gradient methods are also studied in the FL setting. The 'Adaptive Federated Optimization' [226] method proposed to use adaptive gradients on the server side while the local gradients are used to update the states of the adaptive gradient methods. [238, 239] extend the method in [226] to include the AMSGrad method. In [227], the authors first showed the divergence of a naive local AMSGrad method that directly averages the primal states periodically. The authors then proposed Local-AMSGrad, a method in which clients update adaptive learning rates locally and average at the synchronization step. At the server average step, both primal states and local adaptive learning rates are averaged to replace the old states. More recently, [227, 232] proposed to use fixed adaptive learning rates locally. Finally, another line of research [240, 241, 242, 243] considers federated adaptive learning rates through the compression approach, these methods communicate local gradients at every step, but the compression techniques are used to reduce the communication cost. All these methods study federated adaptive methods in the unconstrained case. For solving problems with constraint in FL, a related work is [229], where authors propose a modified local-SGD method based on the dual-averaging, however, it does not support the adaptive gradient methods.

Adaptive Gradients in the Non-distributed Learning. Adaptive gradient methods are widely used in the non-distributed machine learning setting. Adagrad [244] was one of the the first adaptive gradient method proposed in literature, and was shown to outperform SGD in the sparse gradient setting. Since Adagrad does not perform well under dense gradient setting and non-convex setting, some of its variants are proposed, such as

SC-Adagra [245] and SAdagrad [246]. Furthermore, Adam [215] and YOGI [247] proposed to use the exponential moving average instead of the arithmetic average used in Adagrad. Adam/YOGI is widely used in deep learning applications; however, Adam diverges in some settings and the gradient information quickly disappears, so AMSGrad [248] is proposed, and it applies an extra 'long term memory' variable to preserve the past gradient information to handle the convergence issue of Adam. The convergence of Adam-type methods is also studied in the literature [10, 31, 249, 250, 251]. Adaptive gradient methods with good generalization performance are also proposed, such as AdamW [252], Padam [253], Adabound [254], Adabelief [255] and AaGrad-Norm [256].

8.3 Preliminaries

In this section, we introduce some preliminaries. We consider the following formulation of Federated Learning (FL) with K clients:

$$\min_{x\in\mathcal{X}\subset\mathbb{R}^d}\left\{f(x):=\frac{1}{K}\sum_{k=1}^{K}\left\{f^{(k)}(x):=\mathbb{E}_{\xi^{(k)}\sim\mathcal{D}^{(k)}}[f^{(k)}(x;\xi^{(k)})]\right\}\right\}. \tag{8.1}$$

For the k_{th} client, we optimize the loss objective $f^{(k)}(x):\mathcal{X}\to\mathbb{R}$ which is smooth and possibly non-convex, and x denotes the variable of interest. $\mathcal{X}\subset\mathbb{R}^d$ is a compact and convex set. $\xi^{(k)}\sim\mathcal{D}^{(k)}$ is a random example that follows an unknown data distribution $\mathcal{D}^{(k)}$. The formulation in (8.1) includes both the homogeneous case *i.e.* $f^{(k)}(x)=f^{(j)}(x)$ for any $k,j\in[K]$, and the heterogeneous case *i.e.* $f^{(k)}(x)\neq f^{(j)}(x)$ for some $k,j\in[K]$.

Next, we introduce some basics of adaptive gradient methods from a mirror-descent perspective. Generally, mirror descent is associated with a mirror map $\Phi(x)$. Given the

objective $f(x)$ and the primal state $x_t \in \mathcal{X}$ at t_{th} step, we first map the primal state to the mirror space via the mirror map $y_t = \nabla\Phi(x_t)$, then we perform the gradient descent step in the mirror space: $y_{t+1} = y_t - \eta\nabla f(x_t)$, where η is the learning rate, finally, we map y_{t+1} back to the primal space as $x_{t+1} = \arg\min_{x\in\mathcal{X}} D_\Phi(x, \hat{x}_{t+1})$, where $y_{t+1} = \nabla\Phi(\hat{x}_{t+1})$ and $D_\Phi(x, y)$ denotes the Bregman Divergence associated to Φ: $D_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle\nabla\Phi(y), x - y\rangle$. Equivalently, the mirror descent step can also be written as a Bregman proximal gradient step as follows: $x_{t+1} = \arg\min_{x\in\mathcal{X}} \eta\langle\nabla f(x_t), x\rangle + D_\Phi(x, x_t)$. For the adaptive gradient methods, we uses the following mirror map: $\Phi(x) = \frac{1}{2}x^T Hx$, where H is a positive definite adaptive matrix. Many adaptive gradient methods can be written in the following proximal gradient descent form:

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \eta\langle\nu_t, x\rangle + \frac{1}{2}(x - x_t)^T H_t(x - x_t), \tag{8.2}$$

we replace the gradient $\nabla f(x)$ with the generalized gradient estimation ν_t, besides, we replace H with H_t based on the fact that the adaptive matrix is updated at every step. Next, we show some examples of adaptive gradients methods that can be phrased as the above formulation. For the Adagrad [244] method, we set

$$\nu_t = \nabla f(x_t, \xi_t),\ H_t = Diag(\sqrt{\mu_t}),\ \mu_t = \frac{1}{t}\sum_{i=1}^{t}\nu_i^2 \tag{8.3}$$

For Adam [215], we have:

$$\hat{\nu}_t = (1 - \beta_1)\nabla f(x_t, \xi_t) + \beta_1\hat{\nu}_{t-1}, \hat{\mu}_t = (1 - \beta_2)\nabla f(x_t, \xi_t)^2 + \beta_2\hat{\mu}_{t-1}$$

Algorithm 16 FedDA-Server

Input: Number of global epochs E, size of the first mini-batch b_1, tuning parameters $\{\beta_\tau\}_{i=1}^{E}$;
Initialize: Choose $x_0 \in \mathcal{X}$ and each client selects a mini-batch samples $\mathcal{B}_0^{(k)}$ of size b_1 to evaluate $\nabla f^{(j)}(x_0, \mathcal{B}_0^{(k)})$ locally and the server averages local gradients to compute ν_0;
for $\tau = 0$ **to** $E-1$ **do**
  for the client $k \in [K]$ in parallel **do**
    $(z_{\tau+1,I}^{(k)},\, \nu_{\tau+1,I}^{(k)}) =$ **FedDA**-client$(x_\tau,\, \nu_\tau,\, H_\tau)$
  end for
  Compute $z_{\tau+1} = \frac{1}{K}\sum_{k=1}^{K} z_{\tau+1,I}^{(k)}$;
  Compute $x_{\tau+1} = \arg\min_{x\in\mathcal{X}}\{-\langle x, z_{\tau+1}\rangle + \frac{1}{2}(x-x_\tau)^T H_\tau (x-x_\tau)\ \}$;
  Compute $\nu_{\tau+1} = \frac{1}{K}\sum_{k=1}^{K} \nu_{\tau+1,I}^{(k)}$, $H_{\tau+1} = \mathcal{V}(H_\tau, z_{\tau+1}, \beta_\tau)$;
end for

Algorithm 17 FedDA-Client $(x_\tau, \nu_\tau,\, H_\tau)$

Input: Number of local steps I, mini-batch size b, tuning parameters $\{\eta_{\tau+1,i}\}_{i=0}^{I-1}$, $\{\alpha_{\tau+1,i}\}_{i=1}^{I}$;
Initialize: $x_{\tau+1,0}^{(k)} = x_\tau$; $\nu_{\tau+1,0}^{(k)} = \nu_\tau$; $z_{\tau+1,0}^{(k)} = 0$;
for $i = 0$ **to** $I-1$ **do**
  Compute $z_{\tau+1,i+1}^{(k)} = z_{\tau+1,i}^{(k)} - \eta_{\tau+1,i}\nu_{\tau+1,i}^{(k)}$;
  Compute $x_{\tau+1,i+1}^{(k)} = \arg\min_{x\in\mathcal{X}}\{-\langle x, z_{\tau+1,i+1}^{(k)}\rangle + \frac{1}{2}(x-x_{\tau+1,0}^{(k)})^T H_\tau (x-x_{\tau+1,0}^{(k)})\ \}$;
  Compute $\nu_{\tau+1,i+1}^{(k)} = \mathcal{U}(\nu_{\tau+1,i}^{(k)}, x_{\tau+1,i+1}^{(k)}, x_{\tau+1,i}^{(k)}; \alpha_{\tau+1,i+1}, \mathcal{B}_{\tau+1,i+1}^{(k)})$, where $\mathcal{B}_{\tau+1,i+1}^{(k)}$ is a minibatch of size b of random samples from the client k;
end for
Output: Send $z_{\tau+1,I}^{(k)}$, $\nu_{\tau+1,I}^{(k)}$ to the server.

$$\nu_t = \hat{\nu}_t/(1-\gamma_1^t),\ \mu_t = \hat{\mu}_t/(1-\gamma_2^t), H_t = Diag(\sqrt{\mu_t} + \epsilon) \tag{8.4}$$

where $\beta_1, \beta_2, \gamma_1, \gamma_2$ are some constants.

8.4 Local Adaptive Gradients via Dual Averaging

In this section, we introduce **FedDA**, a fast adaptive gradient framework for federated constrained optimization problems. The procedure of **FedDA** is summarized in

Algorithm 16. In Algorithm 16, we perform E global steps and at each global step, every client runs Algorithm 17.

In Algorithm 17, clients receive the current model weight x_τ, gradient estimation ν_τ and adaptive gradient matrix H_τ. The clients then perform I local training steps: line 3-line 7 in Algorithm 17. For each step, we first accumulate the dual state in the variable $z_{\tau,i}^{(k)}$ (line 4), then we calculate the local primal state $x_{\tau,i}^{(k)}$ (line 5), which is a proximal gradient step similar to (8.2). The function of this step is to map the aggregated dual state $z_{\tau,i}^{(k)}$ back to the primal space, and we use the primal state to query the gradient to update the estimation of the gradient $\nu_{\tau,i}^{(k)}$ (line 6). Note that we use a fixed adaptive matrix H_τ during local steps, this makes the clients share the same dual space. In line 6 of Algorithm 17, we update the gradient estimation $\nu_{\tau,i}^{(k)}$. The update rule $\mathcal{U}(\cdot)$ is general, *e.g.*,the momentum-based variance reduction update (8.5) and the momentum update (8.6) as follows ($\alpha_{\tau,i}$ is a momentum coefficient):

$$\nu_{\tau+1,i+1}^{(k)} = \nabla f^{(k)}(x_{\tau+1,i+1}^{(k)}, \mathcal{B}_{\tau+1,i+1}^{(k)}) + (1-\alpha_{\tau+1,i+1})(\nu_{\tau+1,i}^{(k)} - \nabla f^{(k)}(x_{\tau+1,i}^{(k)}, \mathcal{B}_{\tau+1,i+1}^{(k)})) \quad (8.5)$$

and

$$\nu_{\tau+1,i+1}^{(k)} = \alpha_{\tau+1,i+1}\nabla f^{(k)}(x_{\tau+1,i+1}^{(k)}, \mathcal{B}_{\tau+1,i+1}^{(k)}) + (1-\alpha_{\tau+1,i+1})\nu_{\tau+1,i}^{(k)} \quad (8.6)$$

After the client finishes Algorithm 17, it returns the aggregated local dual states $z_{\tau+1,I}^{(k)}$ and the local gradient estimation $\nu_{\tau+1,I}^{(k)}$ to the server. The server first averages the local dual states (line 8 of Algorithm 16) to get $z_{\tau+1}$. We can average local dual states as all

clients have a common dual space. The server then calculates the new primal states $x_{\tau+1}$ as in line 9 of Algorithm 16. Next, the gradient estimation ν_τ is also updated by averaging local gradient estimates (line 9 of Algorithm 16). Finally, we update the adaptive matrix H_τ (line 9 of Algorithm 16). The update rule $\mathcal{V}$ is general, *e.g.*,

$$\mu_{\tau+1} = \beta_{\tau+1} z_{\tau+1}^2 / \eta_{\tau+1,I-1}^2 + (1-\beta_{\tau+1})\mu_\tau, H_{\tau+1} = Diag(\sqrt{\mu_{\tau+1}} + \epsilon) \tag{8.7}$$

and

$$\mu_{\tau+1} = \beta_{\tau+1} ||z_{\tau+1}|| / \eta_{\tau+1,I-1} + (1-\beta_{\tau+1})\mu_\tau, H_{\tau+1} = (\mu_{\tau+1} + \epsilon) I_d \tag{8.8}$$

where we set $\mu_0 = 0$, ϵ is some constant. In summary, Algorithm 16 aggregates and averages dual states at each global round. The adaptive matrix H_τ is fixed during local updates and is refreshed on the server side at each global round. Since the algorithm uses a new mirror map (adaptive gradient matrix) at each global round, we name the strategy as restarted dual averaging.

Remark 160. *A notable characteristic of FedDA is the dual-averaging strategy in local updates, this contrasts with the 'primal averaging' strategy used by many existing adaptive FL algorithms [208]. In fact, dual-averaging is essential for FedDA to solve federated constrained optimization problems, due to the fact that primal and dual states are connected through a non-linear mapping (*i.e. *the projection operation).*

Remark 161. *By choosing different update rules $\mathcal{U}$ and $\mathcal{V}$, we can create many variants of FedDA. An representative is* ***FedDA-MVR****, in which we update $\nu_{\tau,i}^{(k)}$ with momentum-*

based variance reduction ((8.5)) and the adaptive matrix H_τ with an exponential average of the gradient square ((8.7)).

Remark 162. *The argmin operation in line 8 of Algorithm 16 (line 5 of Algorithm 17) does not lead to extra computational cost. We solve it through two steps: solving the quadratic optimization problem (a generalized adaptive gradient step) and projection. The projection is necessary to make the solution feasible and for many constraints, the proximal operators are well defined.*

8.5 Theoretical Analysis

In this section, we provide the theoretical analysis of our **FedDA**. More specifically, we focus on the analysis of **FedDA-MVR**. We first state the assumptions we need in our analysis:

8.5.1 Some Mild Assumptions

Assumption 163 (Bounded Client Heterogeneity)**.** *The difference of gradients between different workers are bounded:* $\|\nabla f^{(k)}(x) - \nabla f^{(\ell)}(x)\|^2 \leq \zeta^2,\ \forall k, \ell \in [K]$.

We measure the heterogeneity of the clients in terms of gradient dissimilarity. The above assumption or its similar form is also exploited in the analysis of other FL Algorithms, such as in [119, 231].

Assumption 164. *The function $f(x)$ is bounded from below in $\mathcal{X}$,* i.e., $f^* = \inf_{x\in\mathcal{X}} f(x)$.

Assumption 165 (Unbiased and Bounded-variance Stochastic Gradient)**.** *The stochastic*

gradients are unbiased with bounded variance, i.e. $\mathbb{E}[\nabla f^{(k)}(x;\xi^{(k)})] = \nabla f^{(k)}(x)$ *and there exists a constant* σ *such that* $\mathbb{E}\|\nabla f^{(k)}(x;\xi^{(k)}) - \nabla f^{(k)}(x)\|^2 \leq \sigma^2, \forall\ \xi^{(k)} \sim \mathcal{D}^{(k)},\ \forall\ k \in [K]$

Assumption 164 guarantees the feasibility of the Federated Learning problem (8.1), and Assumption 165 is widely used in stochastic optimization analysis.

Assumption 166. *The adaptive matrix* H_τ *is symmetric positive definite,* i.e. *there exists a constant* $\rho > 0$ *such that* $H_\tau \succeq \rho I_d \succ 0,\ \forall t \geq 1,$

In our analysis, we assume the adaptive matrix is positive definite, and this requirement can be easily satisfied by many adaptive gradient methods. Firstly, most adaptive gradient methods always have non-negative adaptive learning rates, such as (8.3) and (8.4). To make it positive, we can add a bias term ϵ such as in the Adam update rule (8.4).

Assumption 167 (Sample Gradient Lipschitz Smoothness)**.** *The stochastic functions* $f^{(k)}(x,\xi^{(k)})$ *with* $\xi^{(k)} \sim \mathcal{D}^{(k)}$ *for all* $k \in [K]$*, satisfy the mean squared smoothness property, i.e, we have* $\mathbb{E}\|\nabla f^{(k)}(x;\xi^{(k)}) - \nabla f^{(k)}(y;\xi^{(k)})\|^2 \leq L^2\|x-y\|^2$ *for all* $x, y \in \mathbb{R}^d$

The smoothness assumption above is a slightly stronger requirement than the standard smooth condition, but this assumption is widely used in the analysis of variance reduction methods, such as SPIDER [257] and STORM [30].

8.5.2 Convergence Property of **FedDA-MVR**

In this subsection, we study the convergence property of our **FedDA-MVR** variant. For convenience of discussion, we **redefine the subscript** $t = \tau I + i$, *i.e.* we denote the t step as the i local step in the τ global round. Similarly, we denote the total number of

running steps as $T = EI$. We analyze our algorithm through the following measure:

$$\mathcal{G}_t = \frac{\rho^2}{\eta_t^2}||\tilde{x}_t - \tilde{x}_{t+1}||^2 + ||\bar{\nu}_t - \nabla f(\tilde{x}_t)||^2 \tag{8.9}$$

where $\bar{\nu}_t$ denotes the average gradient estimation at the t_{th} step and $\tilde{x}_t$ denotes the virtual global primal state at the t_{th} step (see Section B.2 in the appendix for formal definitions). In Remark B.18 of the appendix, we discuss the intuition of the measure $\mathcal{G}_t$. In particular, in the unconstrained case *i.e.* when $\mathcal{X} = R^d$, the measure upper-bounds the square norm of the gradient. Therefore, the convergence of our measure $\mathcal{G}_t$ means the convergence to a first-order stationary point. Now, we are ready to provide the main result of our convergence theorem.

Theorem 168. *In Algorithm 16, given the parameter* $\kappa = \frac{\rho K^{2/3}}{L}$, $c = \frac{96L^2}{K\rho^2} + \frac{\rho}{72\kappa^3 L K^{0.5} I^2}$, $w = \max\{2, 48^3 I^6 K^{3.5}\}$, $b_1 \geq 1$, $b \geq 1$, $\beta > 0$ *and choose learning rate* $\eta_t = \frac{\kappa}{(w_t + t + I)^{1/3}}$, *the momentum coefficient* $\alpha_t = c\eta_t^2$, *the adaptive gradient coefficient* $\beta_t = \beta$ *and the mini-batch size of* b_1 *for the first iteration and* b *for other iterations, then we have:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathcal{G}_t] \leq \left[\frac{96LK^{0.5}I^2}{T} + \frac{2L}{K^{2/3}T^{2/3}}\right](f(x_0) - f^*) + \left[\frac{72KI^4}{b_1 T} + \frac{3K^{0.5}I^2}{2b_1 K^{2/3}T^{2/3}}\right]\sigma^2$$
$$+ 192^2 \times \left(\frac{48K^{0.5}I^2}{T} + \frac{1}{K^{2/3}T^{2/3}}\right) \times \left(\frac{\sigma^2}{4b} + \frac{2\zeta^2}{21}\right)\log(T+1).$$

Note, by choosing a proper value of local updates I and using a minibatch of samples for the first iteration to decrease the noise, our result matches the best known convergence rate for stochastic federated gradient methods [119], *i.e.* our algorithms has sample complexity of $\tilde{O}(\epsilon^{-1.5})$ with a linear speed up *w.r.t* the number of clients K. More formally, we

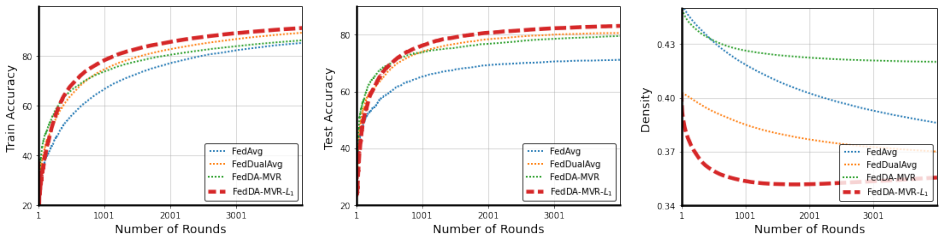


Figure 8.1: Results for the PATHMNIST [2] dataset. Plots show the Train Accuracy, Test Accuracy, Density of Model Parameters vs Number of Rounds (E in Algorithm 16) respectively. The post-fix of L_1 means the L_1 constraints. Number of local steps I is chosen as 5.

have:

Corollary 169. *Suppose in Algorithm 16, we set* $I = O((T/K^{3.5})^{1/6})$*, and use sample minibatch of size* $O(K^{0.5}I^2)$ *in the initialization, then we have:* $\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}[\mathcal{G}_t]\right) = \tilde{O}(K^{-2/3}T^{-2/3})$*, and to reach an* ϵ*-stationary point, we need to make* $\tilde{O}(K^{-1}\epsilon^{-1.5})$ *number of steps and need* $\tilde{O}(K^{-0.25}\epsilon^{-1.25})$ *number of communication rounds.*

As shown by Corollary 169, FedDA-MVR achieves iteration complexity $O(K^{-1}\epsilon^{-1.5})$ which matches the optimal rates of non-convex stochastic optimization [258]. In contrast, FedDualAvg [229] shows the convergence rate of $O(\epsilon^{-2})$ for the general non-convex objective under the strong bounded gradient assumption. Our FedDA-MVR achieves a faster convergence rate and does not require the bounded gradient assumption. As for the communication complexity, we reach $\tilde{O}(K^{-0.25}\epsilon^{-1.25})$. Some methods achieve $O(\epsilon^{-1})$ communication rate such as STEM [119] and FAFED [232]. However, STEM does not consider the adaptive gradient methods nor the constrained setting; FAFED considers the adaptive gradient methods, but it requires the strong assumption of bounded gradient and only considers the unconstrained setting.

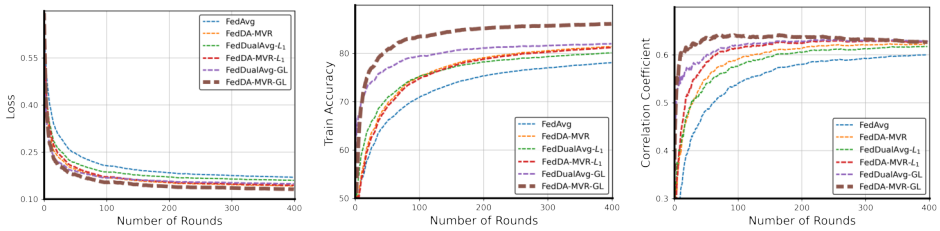


Figure 8.2: Results for the MEMset Donar Dataset [3]. Plots show the Train Loss, Train Accuracy and the Pearson Correlation Coefficient (between prediction and targets). The post-fix of L_1 means L_1 constraints and GL means Group Lasso constraints. Number of local steps I is 5.

8.6 Numerical Experiments

In this section, we perform experiments to verify the efficacy of our proposed **FedDA** on federated biomarker identification task and the general classification tasks. More specifically, we consider the variant of **FedDA-MVR** here, and defer experiments for other variants to Section A of the appendix. We consider three sets of experiments: Colorrectal Cancer Survival Prediction, Splice Site Detection and a general multiclass image classification task. All experiments are run on a machine with an Intel Xeon Gold 6248 CPU and 4 Nvidia Tesla V100 GPUs. The code is written in Pytorch. We simulate the Federated Learning environment through the Pytorch.distributed package.

8.6.1 Colorrectal Cancer Survival Prediction with Biomarker Identification

In this subsection, we consider a colorrectal cancer prediction task on the PATHMNIST dataset [2, 259], which contains 9 different classes and 89996 training images. We equally randomly split the training set into 10 clients. We use the original test set for the metric. In this task, we impose the L_1 sparsity constraint to identify biomarkers.

We compare with the following baselines: FedAvg [92] and FedDualAvg [229]. For our FedDA-MVR, we train with and without the L_1 constraint.

The results are summarized in Figure 8.1, the plots are averaged over 5 independent runs and then smoothed. In Figure 8.1, FedDualAvg and FedDA-MVR-L_1 consider the L_1 constraint, while FedAvg and FedDA-MVR do not. We show results of Train/Test Accuracy and also the number of non-zero (below a threshold) elements in the parameter (the rightmost plot in Figure 8.1). As shown in the plots, FedDA-MVR-L_1 outperforms unconstrained FedDA-MVR in all metrics, in particular, FedDA learns a much sparser model and therefore can better identify important factors for cancer survival than other methods. Furthermore, FedDA-MVR-L_1 also outperforms FedAvg and FedDualAvg in all metrics. This shows that our algorithm can effectively exploit adaptive gradient information in the constrained case. For more details of this experiment, please refer to Section A.1 of the appendix.

8.6.2 Splice Site Detection with Biomarker Identification

In this subsection, we consider a splice site detection task on the MEMset Donar Dataset [3]. Splice sites are the regions between coding (exons) and non-coding (introns) DNA segments. Splice site detection plays an important role in gene finding. We follow the train/test split in [3] and then randomly split the training set to 10 clients. Group Lasso is widely used to solve the splice site detection problem, so we use FedDA-MVR with the Group Lasso constraint in the experiments. For the baselines, we compare with FedAvg [92], FedDualAvg [229] and FedDA-MVR without constraints. For FedDualAvg,

we consider the L_1 constraint and Group Lasso constraint.

The results are summarized in Figure 8.2, the plots are averaged over 5 independent runs and then smoothed. As shown in the plots, FedDA-MVR-GL outperforms unconstrained FedDA-MVR and FedDA-MVR-L_1. In fact, FedDA-MVR-GL gets better performance by identifying meaningful feature groups. Furthermore, FedDA-MVR-GL also outperforms FedAvg and FedDualAvg-GL (FedDualAvg-L_1). This shows that our algorithm can effectively exploit acceleration of adaptive gradients. For more details of this experiment, please refer to Section A.1 of the appendix.

8.6.3 Image Classification Task with CIFAR10 and FEMNIST

FedDA is a general framework for FL and can also solve unconstrained tasks. In this subsection, we consider an unconstrained image classification task. More specifically, we consider two datasets: CIFAR10 [142] and FEMNIST [165]. We construct both homogeneous and heterogeneous cases based on CIFAR10. For the homogeneous case, we uniformly randomly distribute them into 10 clients and for the heterogeneous case, we create imbalanced class distribution among clients (please see Appendix A.3). FEMNIST is a Federated dataset of hand-written digits; it contains hand-written digits of 3550 users. Data distribution of FEMNIST is heterogeneous for different writing styles of people. We compare our method with following baselines: non-adaptive methods: FedAvg [92], FedCM [230], STEM [119] and adaptive methods: FedAdam [226], Local-Adapt [260], Local-AMSGrad [227], MIME-MVR [208].

For all methods, we tune their hyper-parameters to find the best setting. The results

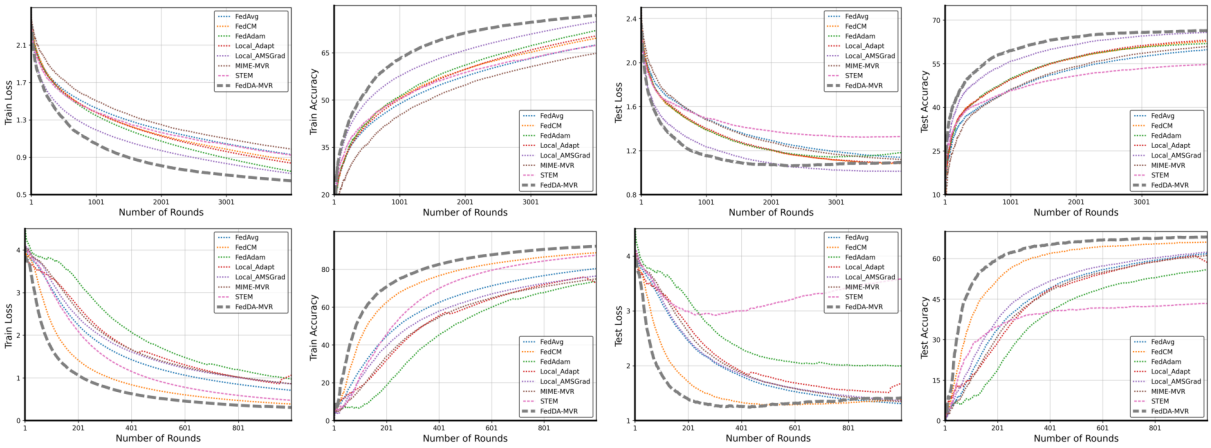


Figure 8.3: Results for (Homogeneous) CIFAR10 dataset (Top) and FEMNIST (Bottom). From left to right, we show Train Loss, Train Accuracy, Test Loss, Test Accuracy $w.r.t$ the number of rounds (E in Algorithm 16), respectively. I is chosen as 5.

are summarized in Figure 8.3, the plots are averaged over 5 runs and then smoothed. As shown in the figures, our FedDA-MVR outperforms all baselines. We observe that adaptive methods in general get better train and test performance. Finally, the superior performance of our method compared with the three adaptive baselines shows that our method exploits adaptive information better; for example, MIME-MVR also exploits the momentum-based variance reduction technique, but it fixes all optimizer states during local updates, in contrast, we only fix the adaptive matrix but update the momentum $\nu_t^{(k)}, k \in [K]$ at every step. For the full set of experiments, please refer to Section A.1 and A.2 of the appendix.

8.7 Proof of Theorems

In this section, we provide the convergence analysis of our algorithm.

8.7.1 Preliminary Propositions

Proposition 170. *Let* $\{\theta_k\}, k \in K$ *be* K *vectors. Then the following are true:* $||\theta_i+\theta_j||^2 \leq (1+\lambda)||\theta_i||^2 + (1+\frac{1}{\lambda})||\theta_j||^2$ *for any* $a > 0$ *and* $||\sum_{k=1}^K \theta_k||^2 \leq K\sum_{k=1}^K ||\theta_k||^2$

Proposition 171. *For a finite sequence* $z^{(k)} \in \mathbb{R}^d$ *for* $k \in [K]$ *define* $\bar{z} \coloneqq \frac{1}{K}\sum_{k=1}^K z^{(k)}$*, we then have* $\sum_{k=1}^K \|z^{(k)} - \bar{z}\|^2 \leq \sum_{k=1}^K \|z^{(k)}\|^2$.

Proposition 172. *Let* $z_0 > 0$ *and* $z_1, z_2, \ldots, z_T \geq 0$*. We have* $\sum_{t=1}^T \frac{z_t}{z_0+\sum_{i=t}^t z_i} \leq \log\left(1 + \frac{\sum_{i=1}^t z_i}{z_0}\right)$.

These propositions are standard results. For proofs, the reader can refer to Lemma 3 of [96] for Proposition 1 and Lemma C.1 and Lemma C.2 in [119] for Propositions 2 and 3.

8.7.2 Preliminary Lemmas in local updates

We first introduce some notation. For $0 \leq i \leq I$, we denote:

$$\psi_{\tau,i}^{(k)}(x) = -\langle x, z_{\tau,i}^{(k)}\rangle + \frac{1}{2}(x - x_{\tau,0}^{(k)})^T H_{\tau-1}(x - x_{\tau,0}^{(k)}), \tag{8.10}$$

then, by definition (Line 4 of Algorithm 17), we have:

$$x_{\tau,i}^{(k)} = \arg\min_{x\in\mathcal{X}} \psi_{\tau,i}^{(k)}(x), \tag{8.11}$$

we also define

$$\tilde{\psi}_{\tau,i}(x) = -\langle x, \bar{z}_{\tau,i}\rangle + \frac{1}{2}(x - x_{\tau,0})^T H_{\tau-1}(x - x_{\tau,0}), \tag{8.12}$$

where $\bar{z}_{\tau,i} = \frac{1}{K}\sum_{k=1}^{K} z_{\tau,i}^{(k)}$ is the virtual average of $z_{\tau,i}^{(k)}$ and $x_{\tau,0} = x_\tau$. Then we define

$$\tilde{x}_{\tau,i} = \arg\min_{x\in\mathcal{X}} \tilde{\psi}_{\tau,i}(x), \tag{8.13}$$

Remark 173. *Note that the global primal state $\tilde{x}_i$ is not the arithmetic mean of the local states $x_i^{(k)}$ in general.*

Finally, we also define

$$\tilde{d}_{\tau,i} = \frac{1}{\eta_{\tau,i}}(\tilde{x}_{\tau,i} - \tilde{x}_{\tau,i+1}),\ d_{\tau,i}^{(k)} = \frac{1}{\eta_{\tau,i}}(x_{\tau,i}^{(k)} - x_{\tau,i+1}^{(k)}), k \in [K], i \in [I], \tag{8.14}$$

Furthermore, recall that by the procedure of Algorithm 17 (line 6), we have

$$\bar{\nu}_{\tau,i} = \frac{1}{\eta_{\tau,i}}(\bar{z}_{\tau,i} - \bar{z}_{\tau,i+1}),\ \nu_{\tau,i}^{(k)} = \frac{1}{\eta_{\tau,i}}(z_{\tau,i}^{(k)} - z_{\tau,i+1}^{(k)}),\ k \in [K],\ i \in [I], \tag{8.15}$$

Remark 174. *When it is clear from the context, we omit the global epoch τ in the subscript of the definitions,* i.e. *we use $\psi_i^{(k)}(x)$, $\tilde{\psi}_i(x)$, $x_i^{(k)}$, $\tilde{x}_i$, $\tilde{d}_i$, $d_i^{(k)}$, $\bar{\nu}_i$, $\nu_i^{(k)}$ and H.*

Next, we introduce the following lemma related to local updates. We omit the global epoch number τ in the subscript.

Lemma 175. *For any $i \in [I]$ and $k \in [K]$, we have the following inequalities be satisfied:*

1. $\langle \nu_i^{(k)}, d_i^{(k)} \rangle \geq \rho ||d_i^{(k)}||^2, ||\nu_i^{(k)}|| \geq \rho ||d_i^{(k)}||$
2. $\langle \bar{\nu}_i, \tilde{d}_i \rangle \geq \rho ||\tilde{d}_i||^2, ||\bar{\nu}_i|| \geq \rho ||\tilde{d}_i||$;
3. $||z_i^{(k)} - \bar{z}_i|| \geq \rho ||x_i^{(k)} - \tilde{x}_i||$;

Proof. The first and second claims follow similar derivations, and we provide only the derivations for the first claim. First, if $i = 1$, we have

$$x_1^{(k)} = \arg\min_{x \in \mathcal{X}} \ -\langle x, z_1^{(k)} \rangle + \frac{1}{2}(x - x_0^{(k)})^T H (x - x_0^{(k)}),$$

by the first-order optimality condition, we have:

$$\langle -z_1^{(k)} + H(x_1^{(k)} - x_0^{(k)}), u - x_1^{(k)} \rangle \geq 0, \ \forall \ u \in \mathcal{X},$$

choose $u = x_0^{(k)}$ and use the fact that $z_1^{(k)} = -\eta_0 \nu_0$, we have:

$$\eta_0 ||\nu_0^{(k)}|| \times ||x_0^{(k)} - x_1^{(k)}|| \geq \eta_0 \langle \nu_0^{(k)}, x_0^{(k)} - x_1^{(k)} \rangle \geq (x_1^{(k)} - x_0^{(k)})^T H (x_1^{(k)} - x_0^{(k)}) \geq \rho ||x_0^{(k)} - x_1^{(k)}||^2$$

we use the Cauchy-Schwartz inequality in the leftmost inequality and use the strong convexity assumption of the adaptive matrix in the rightmost inequality, we get the result in the lemma.

Next if $i > 0$, by the definition of $\psi_i^{(k)}(x)$, we have:

$$\psi_i^{(k)}(x_{i+1}^{(k)}) - \psi_i^{(k)}(x_i^{(k)}) = -\langle z_i^{(k)}, x_{i+1}^{(k)} - x_i^{(k)} \rangle + \frac{1}{2}(x_{i+1}^{(k)} - x_i^{(k)})^T H (x_{i+1}^{(k)} + x_i^{(k)} - 2x_0^{(k)}) \tag{8.16}$$

Then by the definition of $x_i^{(k)}$, and the first order optimality condition, we have

$$\langle -z_i^{(k)} + H(x_i^{(k)} - x_0^{(k)}), u - x_i^{(k)} \rangle \geq 0, \ \forall \ u \in \mathcal{X},$$

if we pick $u = x_{i+1}^{(k)}$, we have $-\langle z_i^{(k)}, x_{i+1}^{(k)} - x_i^{(k)} \rangle \geq -(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_i^{(k)} - x_0^{(k)})$, plug this inequality to (8.16), we have:

$$\begin{aligned}
&\psi_i^{(k)}(x_{i+1}^{(k)}) - \psi_i^{(k)}(x_i^{(k)}) \\
&\geq -(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_i^{(k)} - x_0^{(k)}) + \frac{1}{2}(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} + x_i^{(k)} - 2x_0^{(k)}) \\
&\geq \frac{1}{2}(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} - x_i^{(k)})
\end{aligned}$$

Similarly for $\psi_{i+1}^{(k)}$, we have:

$$\psi_{i+1}^{(k)}(x_{i+1}^{(k)}) - \psi_{i+1}^{(k)}(x_i^{(k)}) = -\langle z_{i+1}^{(k)}, x_{i+1}^{(k)} - x_i^{(k)} \rangle + \frac{1}{2}(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} + x_i^{(k)} - 2x_0^{(k)})$$

and by the definition of $x_{i+1}^{(k)}$ and the first order optimality condition, we can get

$$\langle -z_{i+1}^{(k)} + H(x_{i+1}^{(k)} - x_0^{(k)}), u - x_{i+1}^{(k)} \rangle \geq 0, \ \forall \ u \in \mathcal{X},$$

pick $u = x_i^{(k)}$, we have $-\langle z_{i+1}^{(k)}, x_{i+1}^{(k)} - x_i^{(k)} \rangle \leq -(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} - x_0^{(k)})$, plug this inequality to the above equality, we have:

$$\begin{aligned}
&\psi_{i+1}^{(k)}(x_{i+1}^{(k)}) - \psi_{i+1}^{(k)}(x_i^{(k)}) \\
&\leq -(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} - x_0^{(k)}) + \frac{1}{2}(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} + x_i^{(k)} - 2x_0^{(k)}) \\
&\leq -\frac{1}{2}(x_{i+1}^{(k)} - x_i^{(k)})^T H(x_{i+1}^{(k)} - x_i^{(k)})
\end{aligned}$$

Next, by definition of $\psi_i^{(k)}(x)$ and $\psi_{i+1}^{(k)}(x)$, we have:

$$\psi_{i+1}^{(k)}(x_{i+1}^{(k)}) - \psi_{i+1}^{(k)}(x_i^{(k)}) = \psi_i^{(k)}(x_{i+1}^{(k)}) - \psi_i^{(k)}(x_i^{(k)}) + \eta_i \langle \nu_i^{(k)}, x_{i+1}^{(k)} - x_i^{(k)} \rangle$$

Finally, we combine the above relations and have:

$$\eta_i ||\nu_i^{(k)}|| \times ||x_i^{(k)} - x_{i+1}^{(k)}|| \geq \eta_i \langle \nu_i^{(k)}, x_i^{(k)} - x_{i+1}^{(k)} \rangle \geq (x_{i+1}^{(k)} - x_i^{(k)})^T H (x_{i+1}^{(k)} - x_i^{(k)}) \geq \rho ||x_i^{(k)} - x_{i+1}^{(k)}||^2$$

we use the Cauchy-Schwartz inequality in the leftmost inequality and use the strong convexity assumption of the adaptive matrix in the rightmost inequality, we get the result in the claim of the lemma.

Next, we prove the third claim, by the definition of $\psi_{i+1}^{(k)}$, we have:

$$\psi_{i+1}^{(k)}(x_{i+1}^{(k)}) - \psi_{i+1}^{(k)}(\tilde{x}_{i+1}) = -\langle z_{i+1}^{(k)}, x_{i+1}^{(k)} - \tilde{x}_{i+1} \rangle + \frac{1}{2} (x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H (x_{i+1}^{(k)} + \tilde{x}_{i+1} - 2x_0^{(k)})$$

By the definition of $x_{i+1}^{(k)}$ and first order optimality condition, we have

$$\langle -z_{i+1}^{(k)} + H(x_{i+1}^{(k)} - x_0^{(k)}), u - x_{i+1}^{(k)} \rangle \geq 0, \ \forall \ u \in \mathcal{X},$$

pick $u = \tilde{x}_{i+1}$, we have $-\langle z_{i+1}^{(k)}, x_{i+1}^{(k)} - \tilde{x}_{i+1} \rangle \leq -(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H (x_{i+1}^{(k)} - x_0^{(k)})$. Plug this inequality back to the above inequality, we have:

$$\begin{aligned}
&\psi_{i+1}^{(k)}(x_{i+1}^{(k)}) - \psi_{i+1}^{(k)}(\tilde{x}_{i+1}) \\
&\leq -(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H (x_{i+1}^{(k)} - x_0^{(k)}) + \frac{1}{2} (x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H (x_{i+1}^{(k)} + \tilde{x}_{i+1} - 2x_0^{(k)})
\end{aligned}$$

$$\leq -\frac{1}{2}(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H(x_{i+1}^{(k)} - \tilde{x}_{i+1})$$

Then for $\tilde{\psi}_{i+1}(x)$, we have:

$$\tilde{\psi}_{i+1}^{(k)}(x_{i+1}^{(k)}) - \tilde{\psi}_{i+1}(\tilde{x}_{i+1}) = -\langle \bar{z}_{i+1}, x_{i+1}^{(k)} - \tilde{x}_{i+1} \rangle + \frac{1}{2}(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H(x_{i+1}^{(k)} + \tilde{x}_{i+1} - 2\tilde{x}_0)$$

By the definition of $\tilde{x}_{i+1}$ and first order optimality condition, we have:

$$\langle -\bar{z}_{i+1} + H(\tilde{x}_{i+1} - \tilde{x}_0), u - \tilde{x}_{i+1} \rangle \geq 0, \ \forall \ u \in \mathcal{X},$$

pick $u = x_{i+1}^{(k)}$, we have $-\langle \bar{z}_{i+1}, x_{i+1}^{(k)} - \tilde{x}_{i+1} \rangle \geq -(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H(\tilde{x}_{i+1} - \tilde{x}_0)$. Plug this inequality back to the above inequality, we have:

$$\begin{aligned}
&\tilde{\psi}_{i+1}(x_{i+1}^{(k)}) - \tilde{\psi}_{i+1}(\tilde{x}_{i+1}) \\
&\geq -(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H(\tilde{x}_{i+1} - \tilde{x}_0) + \frac{1}{2}(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H(x_{i+1}^{(k)} + \tilde{x}_{i+1} - 2\tilde{x}_0) \\
&\geq \frac{1}{2}(x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H(x_{i+1}^{(k)} - \tilde{x}_{i+1})
\end{aligned}$$

Next, since we have $x_0^{(k)} = \tilde{x}_0$, then by the definition of $\psi_{i+1}^{(k)}(x)$ and $\tilde{\psi}_{i+1}(x)$ we have:

$$\psi_{i+1}^{(k)}(x_{i+1}^{(k)}) - \psi_{i+1}^{(k)}(\tilde{x}_{i+1}) = \tilde{\psi}_{i+1}(x_{i+1}^{(k)}) - \tilde{\psi}_{i+1}(\tilde{x}_{i+1}) - \langle z_{i+1}^{(k)} - \bar{z}_{i+1}, x_{i+1}^{(k)} - \tilde{x}_{i+1} \rangle$$

Next, we combine the above relations and have:

$$||z_{i+1}^{(k)} - \bar{z}_{i+1}|| \times ||x_{i+1}^{(k)} - \tilde{x}_{i+1}|| \geq \langle z_{i+!}^{(k)} - \bar{z}_{i+1}, x_{i+1}^{(k)} - \tilde{x}_{i+1} \rangle$$
$$\geq (x_{i+1}^{(k)} - \tilde{x}_{i+1})^T H (x_{i+1}^{(k)} - \tilde{x}_{i+1}) \geq \rho ||x_{i+1}^{(k)} - \tilde{x}_{i+1}||^2$$

where the first inequality is by the Cauchy-Schwartz inequality and the last inequality is by the positive definiteness of H. This concludes the proof of the first inequality in the lemma. □

8.7.3 State Consensus Error

As each client performs local update, the states *i.e.* $z_{\tau,i}^{(k)}$ and $\nu_{\tau,i}^{(k)}$ drift away, the following lemmas bound this difference. We omit the global epoch number τ in the subscript.

Lemma 176. *For each $0 \leq i \leq I$, and suppose iterates $z_i^{(k)}$, $k \in [K]$ are generated from Algorithm 17, we have:*

$$\sum_{k=1}^{K} \mathbb{E}\|z_i^{(k)} - \bar{z}_i\|^2 \leq (I-1) \sum_{\ell=1}^{i-1} \eta_\ell^2 \sum_{k=1}^{K} \mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2,$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Based on Algorithm 17, we have $z_0^{(k)} = \bar{z}_0 = 0$, the inequality in the lemma holds trivially. Otherwise, we have

$$z_i^{(k)} = -\sum_{\ell=0}^{i-1} \eta_\ell \nu_\ell^{(k)} \quad \text{and} \quad \bar{z}_i = -\sum_{\ell=0}^{i-1} \eta_\ell \bar{\nu}_\ell.$$

So we have:

$$\sum_{k=1}^{K} \|z_i^{(k)} - \bar{z}_i\|^2 = \sum_{k=1}^{K} \left\| \sum_{\ell=1}^{i-1} \left(\eta_\ell \nu_\ell^{(k)} - \eta_\ell \bar{\nu}_\ell \right) \right\|^2 \leq (I-1) \sum_{\ell=1}^{i-1} \eta_\ell^2 \sum_{k=1}^{K} \|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2$$

where the equality uses the fact $\nu_0^{(k)} = \nu_0$ for $k \in [K]$, the inequality uses the Proposition 170 and the fact that we have $i \leq I$. We get the claim in the lemma by taking expectation on both sides of the above inequality. This completes the proof. □

Lemma 177. *For $i \in [I]$, we have:*

$$\sum_{k=1}^{K} ||d_i^{(k)} - \tilde{d}_i||^2 \leq \frac{4^2(I-1)}{\rho^2 \eta_i^2} \sum_{\ell=1}^{i} \eta_\ell^2 \sum_{k=1}^{K} \mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Firstly, when $i = 0$, $x_0^{(k)} = \tilde{x}_0$, $z_1^{(k)} = \bar{z}_1$, so we have $x_1^{(k)} = \tilde{x}_1$ by Line 5 of Algorithm 17, and then we have $\eta_0 d_0^{(k)} = x_0^{(k)} - x_1^{(k)} = \tilde{x}_0 - \tilde{x}_1 = \eta_t \tilde{d}_0$, the inequality in the lemma holds trivially.

Next when $i > 0$, we have:

$$\begin{aligned}
\eta_i^2 ||d_i^{(k)} - \tilde{d}_i||^2 &= ||x_i^{(k)} - x_{i+1}^{(k)} - (\tilde{x}_i - \tilde{x}_{i+1})||^2 \leq 2||x_i^{(k)} - \tilde{x}_i||^2 + 2||x_{i+1}^{(k)} - \tilde{x}_{i+1}||^2 \\
&\leq \frac{2^2}{\rho^2}\Big(||z_i^{(k)} - \bar{z}_i||^2 + ||z_{i+1}^{(k)} - \bar{z}_{i+1}||^2\Big)
\end{aligned}$$

The last inequality uses claim 3 of Lemma 175. Sum over $k \in [K]$ and use Lemma 176, we have:

$$\begin{aligned}
\rho^2 \eta_i^2 \sum_{k=1}^{K} ||d_i^{(k)} - \tilde{d}_i||^2 &\leq 2^2(I-1) \sum_{\ell=1}^{i-1} \eta_\ell^2 \sum_{k=1}^{K} \mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2 + 2^2(I-1) \sum_{\ell=1}^{i} \eta_\ell^2 \sum_{k=1}^{K} \mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2 \\
&\leq 4^2(I-1) \sum_{\ell=1}^{i} \eta_\ell^2 \sum_{k=1}^{K} \mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2
\end{aligned}$$

This completes the proof. □

8.7.4 Descent Lemma

In this subsection, we bound the descent of function value $f(\tilde{x}_{\tau,i})$ over the virtual sequence $\tilde{x}_{\tau,i}$.

Lemma 178. *Suppose that the sequence $\{x_{\tau,i}^{(k)}\}_{i=0}^{I-1}$ be generated from Algorithm 17, then we have*

$$f(\tilde{x}_{\tau+1}) \leq f(\tilde{x}_{\tau}) - \sum_{i=0}^{I-1}\left(\frac{3\rho\eta_{\tau+1,i}}{4} - \frac{\eta_{\tau+1,i}^2 L}{2}\right)\|\tilde{d}_{\tau+1,i}\|^2 + \sum_{i=0}^{I-1}\frac{\eta_{\tau+1,i}}{\rho}\|\bar{e}_{\tau+1,i}\|^2$$

where $\bar{e}_{\tau,i} = \bar{\nu}_{\tau,i} - \frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(x_{\tau,i}^{(k)})$.

Proof. Since the function $f(x)$ is L-smooth, we have (we omit the global epoch number τ for ease of notation):

$$\begin{aligned}
f(\tilde{x}_{i+1}) &\leq f(\tilde{x}_i) + \langle\nabla f(\tilde{x}_i), \tilde{x}_{i+1} - \tilde{x}_i\rangle + \frac{L}{2}\|\tilde{x}_{i+1} - \tilde{x}_i\|^2 = f(\tilde{x}_i) - \eta_i\langle\nabla f(\tilde{x}_i), \tilde{d}_i\rangle + \frac{L\eta_i^2}{2}\|\tilde{d}_i\|^2 \\
&= f(\tilde{x}_i) - \eta_i\langle\bar{\nu}_i, \tilde{d}_i\rangle - \eta_i\langle\nabla f(\tilde{x}_i) - \bar{\nu}_i, \tilde{d}_i\rangle + \frac{L\eta_i^2}{2}\|\tilde{d}_i\|^2 \\
&\overset{(a)}{\leq} f(\tilde{x}_i) - (\rho\eta_i - \frac{L\eta_i^2}{2})\|\tilde{d}_i\|^2 - \eta_i\langle\nabla f(\tilde{x}_i) - \bar{\nu}_i, \tilde{d}_i\rangle \\
&\overset{(b)}{\leq} f(\tilde{x}_i) - \left(\rho\eta_i - \frac{\eta_i^2 L}{2}\right)\|\tilde{d}_i\|^2 + \frac{\rho\eta_i}{4}\|\tilde{d}_i\|^2 + \frac{\eta_i}{\rho}\|\bar{\nu}_i - \nabla f(\tilde{x}_i)\|^2 \\
&\overset{(c)}{\leq} f(\tilde{x}_i) - \left(\frac{3\rho\eta_i}{4} - \frac{\eta_i^2 L}{2}\right)\|\tilde{d}_i\|^2 + \frac{\eta_i}{\rho}\|\bar{e}_i\|^2
\end{aligned}$$

In inequality (a), we use claim 1 of Lemma 175; inequality (b) uses Young's inequality; inequality (c) denotes $\bar{e}_i = \bar{\nu}_i - \nabla f(\tilde{x}_i)$. For $\tilde{e}_i$, we have: For the τ global epoch, we sum

over $i = 0$ to $I-1$, we have:

$$f(\tilde{x}_{\tau+1,I}) \leq f(\tilde{x}_{\tau+1,0}) - \sum_{i=0}^{I-1}\left(\frac{3\rho\eta_{\tau+1,i}}{4} - \frac{\eta_{\tau+1,i}^2 L}{2}\right)\|\tilde{d}_{\tau+1,i}\|^2 + \sum_{i=0}^{I-1}\frac{2\eta_{\tau+1,i}}{\rho}\|\bar{e}_{\tau+1,i}\|^2$$

Follow the update rules in Algorithm 16 and Algorithm 17, we have $\tilde{x}_{\tau+1,0} = x_\tau$ and $\tilde{x}_{\tau+1,I} = x_{\tau+1}$. This completes the proof. □

8.7.5 Gradient Error Contraction

In this subsection, we bound the gradient estimation error $\bar{e}_{\tau,i}$, where we have $\bar{e}_{\tau,i} = \bar{\nu}_{\tau,i} - \nabla f(\tilde{x}_{\tau,i})$ as defined in Lemma 178, additionally, we also define the global gradient estimation error e_τ as $\bar{e}_\tau = \nu_\tau - \nabla f(x_\tau)$. Note we have $e_\tau = \bar{e}_{\tau,I} = \bar{e}_{\tau+1,0}$. We first show a fact about $\bar{e}_0$, the initial gradient estimation error.

Lemma 179. *For $e_0 \coloneqq \nu_0 - \nabla f(x_0)$, suppose we choose mini-batch size of $|\mathcal{B}_0^{(k)}| = b_1, k \in [K]$], we have: $\mathbb{E}\|e_0\|^2 \leq \frac{\sigma^2}{b_1 K}$.*

Proof. By line 1 of Algorithm 16, we have:

$$\begin{aligned}
\mathbb{E}\|e_0\|^2 &= \mathbb{E}\left\|\nu_0 - \frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(x_0)\right\|^2 \\
&= \mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(x_0;\mathcal{B}_0^{(k)}) - \frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(x_0)\right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla f^{(k)}(x_0;\mathcal{B}_0^{(k)}) - \nabla f^{(k)}(x_0)\right\|^2 \overset{(b)}{\leq} \frac{\sigma^2}{b_1 K}.
\end{aligned}$$

where (a) follows from the following: From the unbiased gradient assumption, we have: $\mathbb{E}\left[\nabla f^{(k)}(x_0^{(k)};\mathcal{B}_0^{(k)})\right] = \nabla f^{(k)}(x_0^{(k)})$, for all $k \in [K]$. Moreover, the samples $\mathcal{B}_0^{(k)}$ and $\mathcal{B}_0^{(\ell)}$ at

the k^{th} and the ℓ^{th} clients are chosen uniformly randomly, and independent of each other for all $k, \ell \in [K]]$ and $k \neq \ell$. Inequality (b) results from the bounded variance assumption. This completes the proof. □

Lemma 180. *Define* $\bar{e}_{\tau,i} := \bar{\nu}_{\tau,i} - \frac{1}{K}\sum_{k=1}^{K} \nabla f^{(k)}(\tilde{x}_{\tau,i})$*, then for every* $\tau \geq 1$ *and* $i \geq 1$*, suppose* $\alpha_i < 1$ *and clients use batchsize* b *in the training, then we have:*

$$\mathbb{E}\|\bar{e}_{\tau,i}\|^2 \leq (1-\alpha_{\tau,i})^2 \mathbb{E}\|\bar{e}_{\tau,i-1}\|^2 + \frac{256(I-1)L^2}{\rho^2 K}\sum_{\ell=1}^{i-1}\eta_{\tau,\ell}^2 \sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)} - \bar{\nu}_{\tau,\ell}\|^2 + \frac{8\eta_{\tau,i-1}^2 L^2}{K}\mathbb{E}\|\tilde{d}_{\tau,i-1}\|^2 + \frac{4\alpha_{\tau,i}^2\sigma^2}{bK}$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. Consider the error term $\|\bar{e}_i\|^2$, $i \geq 1$ (we omit the global epoch number τ for ease of notation), we have:

$$\begin{aligned}
\mathbb{E}\|\bar{e}_i\|^2 &= \mathbb{E}\left\|\bar{\nu}_i - \frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(\tilde{x}_i)\right\|^2 \\
&= \mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) + (1-\alpha_i)\left(\bar{\nu}_{i-1} - \frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)})\right) - \frac{1}{K}\sum_{k=1}^{K}\nabla f^{(k)}(\tilde{x}_i)\right\|^2 \\
&= \mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^{K}\Bigg(\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_i)\Big) \right. \\
&\qquad \left. - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\Big)\Bigg) + (1-\alpha_i)\bar{e}_{i-1}\right\|^2 \\
&= (1-\alpha_i)^2\mathbb{E}\|\bar{e}_{i-1}\|^2 + \frac{1}{K^2}\mathbb{E}\left\|\sum_{k=1}^{K}\Bigg[\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_i)\Big) \right. \\
&\qquad \left. - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\Big)\Bigg]\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&\leq (1-\alpha_i)^2\mathbb{E}\|\bar{e}_{i-1}\|^2 + \frac{2}{K^2}\mathbb{E}\Big\|\sum_{k=1}^{K}\Big[\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_i^{(k)})\Big) \\
&\qquad\qquad - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k-1)})\Big)\Big]\Big\|^2 \\
&\quad + \frac{2}{K^2}\mathbb{E}\Big\|\sum_{k=1}^{K}\Big[\Big(\nabla f^{(k)}(x_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_i)\Big) - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\Big)\Big]\Big\|^2 \\
&\leq (1-\alpha_i)^2\mathbb{E}\|\bar{e}_{i-1}\|^2 + \frac{2}{K^2}\sum_{k=1}^{K}\mathbb{E}\Big\|\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_i^{(k)})\Big) \\
&\qquad\qquad - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\Big)\Big\|^2 \\
&\quad + \frac{2}{K}\sum_{k=1}^{K}\mathbb{E}\Big\|\Big(\nabla f^{(k)}(x_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_i)\Big) - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\Big)\Big\|^2
\end{aligned}
$$

where the first equality uses the definition of $\bar{\nu}_i$; last equality follows from expanding the norm using the inner products across $k \in [K]$ and noting that the cross terms of the second term is zero in expectation because of the samples are sampled independently at different workers.

We first consider the second term above:

$$
\begin{aligned}
&\mathbb{E}\Big\|\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_i^{(k)})\Big) - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\Big)\Big\|^2 \\
&= \mathbb{E}\Big\|(1-a_i)\Big[\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_i^{(k)})\Big) - \Big(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\Big)\Big] \\
&\qquad\qquad + \alpha_i\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_i^{(k)})\Big)\Big\|^2 \\
&\leq 2(1-\alpha_i)^2\mathbb{E}\Big\|\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)})\Big) - \Big(\nabla f^{(k)}(x_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\Big)\Big\|^2 \\
&\qquad\qquad + 2\alpha_i^2\mathbb{E}\Big\|\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_i^{(k)})\Big\|^2 \\
&\leq 2(1-\alpha_i)^2\mathbb{E}\Big\|\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)})\Big\|^2 + 2\alpha_i^2\sigma^2/b \\
&\leq 2(1-\alpha_i)^2L^2\mathbb{E}\|x_i^{(k)} - x_{i-1}^{(k)}\|^2 + 2a_i^2\sigma^2/b \leq 2(1-\alpha_i)^2L^2\eta_{i-1}^2\mathbb{E}\|d_{i-1}^{(k)}\|^2 + 2\alpha_i^2\sigma^2/b
\end{aligned}
$$

$$\leq 4(1-\alpha_i)^2 L^2 \eta_{i-1}^2 \mathbb{E}\|d_{i-1}^{(k)} - \tilde{d}_{i-1}\|^2 + 4(1-\alpha_i)^2 L^2 \eta_{i-1}^2 \mathbb{E}\|\tilde{d}_{i-1}\|^2 + 2\alpha_i^2 \sigma^2 / b$$

where we use Proposition 170 in the first inequality and the bounded variance assumption in the second inequality.

For the third term, we have:

$$\begin{aligned}
&\mathbb{E}\Big\|\nabla f^{(k)}(x_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_i) - (1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\Big)\Big\|^2 \\
&\overset{(a)}{\leq} 2\mathbb{E}\Big\|\nabla f^{(k)}(x_i^{(k)}) - \nabla f^{(k)}(\tilde{x}_i)\Big\|^2 + 2\mathbb{E}\Big\|(1-\alpha_i)\Big(\nabla f^{(k)}(x_{i-1}^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\Big)\Big\|^2 \\
&\leq 2L^2\mathbb{E}\Big\|x_i^{(k)} - \tilde{x}_i\Big\|^2 + 2L^2(1-\alpha_i)^2\mathbb{E}\Big\|x_{i-1}^{(k)} - \tilde{x}_{i-1}\Big\|^2 \\
&\overset{(b)}{\leq} \frac{2L^2}{\rho^2}\mathbb{E}\Big\|z_i^{(k)} - \bar{z}_i\Big\|^2 + \frac{2L^2(1-\alpha_i)^2}{\rho^2}\mathbb{E}\Big\|z_{i-1}^{(k)} - \bar{z}_{i-1}\Big\|^2
\end{aligned}$$

where (a) uses Proposition 170; (b) uses claim 3 of Lemma 175;

Finally, we combine the above inequalities together to get:

$$\begin{aligned}
\mathbb{E}\|\bar{e}_i\|^2 &\leq (1-\alpha_i)^2\mathbb{E}\|\bar{e}_{i-1}\|^2 + \frac{4\alpha_i^2\sigma^2}{bK} + \frac{8\eta_{i-1}^2 L^2}{K^2}\sum_{k=1}^{K}\mathbb{E}\|d_{i-1}^{(k)} - \tilde{d}_{i-1}\|^2 \\
&\quad + \frac{8\eta_{i-1}^2 L^2}{K}\mathbb{E}\|\tilde{d}_{i-1}\|^2 + \frac{8L^2}{K\rho^2}\sum_{k=1}^{K}\left[\mathbb{E}\Big\|z_i^{(k)} - \bar{z}_i\Big\|^2 + \mathbb{E}\Big\|z_{i-1}^{(k)} - \bar{z}_{i-1}\Big\|^2\right] \\
&\leq (1-\alpha_i)^2\mathbb{E}\|\bar{e}_{i-1}\|^2 + \frac{4\alpha_i^2\sigma^2}{bK} + \frac{128(I-1)L^2}{K^2\rho^2}\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2 \\
&\quad + \frac{16\eta_{i-1}^2 L^2}{K}\mathbb{E}\|\tilde{d}_{i-1}\|^2 + \frac{16(I-1)L^2}{K\rho^2}\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2 \\
&\leq (1-\alpha_i)^2\mathbb{E}\|\bar{e}_{i-1}\|^2 + \frac{4\alpha_i^2\sigma^2}{bK} \\
&\quad + \frac{8\eta_{i-1}^2 L^2}{K}\mathbb{E}\|\tilde{d}_{i-1}\|^2 + \frac{256(I-1)L^2}{K\rho^2}\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)} - \bar{\nu}_\ell\|^2
\end{aligned}$$

The first inequality the assumption that $\alpha_i < 1$; the second inequality uses Lemma 176 and

Lemma 177. This completes the proof. □

Lemma 181. *For $\tau \geq 0$ and $i \in [I]$. Suppose we have $\eta_{\tau,i} = \frac{\kappa}{(w_{\tau,i}+i+\tau I)^{1/3}}$, and have $\alpha_i < 1$, $w_{\tau,i} \geq 2$, $\eta_{\tau,i} \leq \frac{\rho}{48LK^{0.5}I^2}$ be satisfied, we have:*

$$\frac{K\rho}{64L^2}\left(\frac{\mathbb{E}\|\bar{e}_{\tau+1}\|^2}{\eta_{\tau+1,I-1}} - \frac{\mathbb{E}\|\bar{e}_{\tau}\|^2}{\eta_{\tau,I-1}}\right) \leq -\sum_{i=0}^{I-1}\frac{3\eta_{\tau+1,i}}{2\rho}\mathbb{E}\|\bar{e}_{\tau+1,i}\|^2 + \sum_{i=0}^{I-1}\frac{\eta_{\tau+1,i}\rho}{8}\mathbb{E}\|\tilde{d}_{\tau+1,i}\|^2 + \sum_{i=0}^{I-1}\frac{\sigma^2c^2\eta_{\tau+1,i}^3\rho}{16bL^2} + \frac{8I(I-1)}{\rho}\sum_{\ell=1}^{I}\eta_{\tau+1,\ell}\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau+1,\ell}^{(k)} - \bar{\nu}_{\tau+1,\ell}\|^2$$

Proof. Using Lemma 180 at the global epoch $\tau-1$, then for $i \geq 0$ (we denote $\eta_{\tau,-1} = \eta_{\tau-1,I-1}$ for all $\tau \geq 1$), we have:

$$\begin{aligned}
&\frac{\mathbb{E}\|\bar{e}_{\tau,i+1}\|^2}{\eta_{\tau,i}} - \frac{\mathbb{E}\|\bar{e}_{\tau,i}\|^2}{\eta_{\tau,i-1}} \\
&\leq \left[\frac{(1-a_{\tau,i+1})^2}{\eta_{\tau,i}} - \frac{1}{\eta_{\tau,i-1}}\right]\mathbb{E}\|\bar{e}_{\tau,i}\|^2 + \frac{256(I-1)L^2}{\rho^2K\eta_{\tau,i}}\sum_{\ell=1}^{i}\eta_{\tau,\ell}^2\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)} - \bar{\nu}_{\tau,\ell}\|^2 \\
&\qquad + \frac{8L^2\eta_{\tau,i}}{K}\mathbb{E}\|\tilde{d}_{\tau,i}\|^2 + \frac{4a_{\tau,i+1}^2\sigma^2}{\eta_{\tau,i}bK} \\
&\overset{(a)}{\leq} \left(\eta_{\tau,i}^{-1} - \eta_{\tau,i-1}^{-1} - c\eta_{\tau,i}\right)\mathbb{E}\|\bar{e}_{\tau,i}\|^2 + \frac{512(I-1)L^2}{\rho^2K}\sum_{\ell=1}^{i}\eta_{\tau,\ell}\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)} - \bar{\nu}_{\tau,\ell}\|^2 \\
&\qquad + \frac{8L^2\eta_{\tau,i}}{K}\mathbb{E}\|\tilde{d}_{\tau,i}\|^2 + \frac{4\sigma^2c^2\eta_{\tau,i}^3}{bK},
\end{aligned}$$

where inequality (a) utilizes the fact that $(1-\alpha_{\tau,i})^2 \leq 1-\alpha_{\tau,i} \leq 1$ and $a_{\tau,i+1} = c\eta_{\tau,i}^2$ for all $i \in [I]$, and the following fact: suppose we choose $\eta_{\tau,i} = \kappa/(w_i+i+\tau I)^{1/3}$, then for $0 \leq l \leq i < I$, we have:

$$\frac{\eta_{\tau,l}}{\eta_{\tau,i}} = \frac{(w_i+i+\tau I)^{1/3}}{(w_l+l+\tau I)^{1/3}} = \left(1 + \frac{w_i+i-w_l-l}{w_l+l+\tau I}\right)^{1/3}$$

$$\leq \left(1+\frac{(I-1)}{w_l+l+\tau I}\right)^{1/3} \leq 1+\frac{(I-1)}{3(w_l+l+\tau I)} \leq 2 \tag{8.17}$$

The first inequality is by the fact that $w_i \leq w_l$ and $0 < i-l < I-1$, the second last inequality uses the concavity of $x^{1/3}$ as: $(x+y)^{1/3} - x^{1/3} \leq y/3x^{2/3}$, while the last inequality uses the fact that $w_l \geq 0$, $I \geq 1$, $l \geq 0$, $\tau \geq 1$.

For the difference $\eta_i^{-1} - \eta_{i-1}^{-1}$, we have:

$$\begin{aligned}\frac{1}{\eta_{\tau,i}} - \frac{1}{\eta_{\tau,i-1}} &= \frac{(w_i+i+\tau I)^{1/3}}{\kappa} - \frac{(w_{i-1}+i-1+\tau I)^{1/3}}{\kappa} \\ &\overset{(a)}{\leq} \frac{(w_{i-1}+i+\tau I)^{1/3}}{\kappa} - \frac{(w_{i-1}+i-1+\tau I)^{1/3}}{\kappa} \\ &\overset{(b)}{\leq} \frac{1}{3\kappa(w_{i-1}+i-1+\tau I)^{2/3}} \overset{(c)}{\leq} \frac{2^{2/3}\kappa^2}{3\kappa^3(w_i+i+\tau I)^{2/3}} \overset{(c)}{=} \frac{2^{2/3}}{3\kappa^3}\eta_i^2 \overset{(d)}{\leq} \frac{\rho}{72\kappa^3 L K^{0.5} I^2}\eta_i,\end{aligned} \tag{8.18}$$

where inequality (a) is because that we choose $w \leq w$, (b) results from the concavity of $x^{1/3}$ as: $(x+y)^{1/3} - x^{1/3} \leq y/(3x^{2/3})$, (c) used the fact that $w \geq 2$, finally, (d) and (e) utilize the definition of $\eta_{\tau,i}$ and the condition that $\eta_{\tau,i} \leq \frac{\rho}{48LK^{0.5}I^2}$, respectively. So if we choose $c = \dfrac{96L^2}{K\rho^2} + \dfrac{\rho}{72\kappa^3 L K^{0.5} I^2}$ we have: $\eta_{\tau,i}^{-1} - \eta_{\tau,i-1}^{-1} - c\eta_{\tau,i} \leq -\frac{96L^2}{K\rho^2}\eta_{\tau,i}$,

Therefore, we have:

$$\begin{aligned}\frac{\mathbb{E}\|\bar{e}_{\tau,i+1}\|^2}{\eta_{\tau,i}} - \frac{\mathbb{E}\|\bar{e}_{\tau,i}\|^2}{\eta_{\tau,i-1}} \leq &-\frac{96L^2\eta_{\tau,i}}{K\rho^2}\mathbb{E}\|\bar{e}_{\tau,i}\|^2 + \frac{512(I-1)L^2}{\rho^2 K}\sum_{\ell=1}^{i}\eta_{\tau,\ell}\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)} - \bar{\nu}_{\tau,\ell}\|^2 \\ &+ \frac{8L^2\eta_{\tau,i}}{K}\mathbb{E}\|\tilde{d}_{\tau,i}\|^2 + \frac{4\sigma^2 c^2 \eta_{\tau,i}^3}{bK},\end{aligned}$$

Multiplying $K\rho/64L^2$ on both sides, we have:

$$\frac{K\rho}{64L^2}\left(\frac{\mathbb{E}\|\bar{e}_{\tau,i+1}\|^2}{\eta_{\tau,i}}-\frac{\mathbb{E}\|\bar{e}_{\tau,i}\|^2}{\eta_{\tau,i-1}}\right)\leq-\frac{3\eta_{\tau,i}}{2\rho}\mathbb{E}\|\bar{e}_{\tau,i}\|^2+\frac{8(I-1)}{\rho}\sum_{\ell=1}^{i}\eta_{\tau,\ell}\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)}-\bar{\nu}_{\tau,\ell}\|^2 + \frac{\eta_{\tau,i}\rho}{8}\mathbb{E}\|\tilde{d}_{\tau,i}\|^2+\frac{\sigma^2c^2\eta_{\tau,i}^3\rho}{16L^2b}.$$

Then we sum the above inequality from 0 to $I-1$ and get:

$$\begin{aligned}\frac{K\rho}{64L^2}\left(\frac{\mathbb{E}\|\bar{e}_{\tau,I}\|^2}{\eta_{\tau,I-1}}-\frac{\mathbb{E}\|\bar{e}_{\tau,0}\|^2}{\eta_{\tau-1,I-1}}\right)&\leq-\sum_{i=0}^{I-1}\frac{3\eta_i}{2\rho}\mathbb{E}\|\bar{e}_{\tau,i}\|^2+\sum_{i=0}^{I-1}\frac{8(I-1)}{\rho}\sum_{\ell=1}^{i}\eta_\ell\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)}-\bar{\nu}_{\tau,\ell}\|^2\\&\quad+\sum_{i=0}^{I-1}\frac{\eta_{\tau,i}\rho}{8}\mathbb{E}\|\tilde{d}_{\tau,i}\|^2+\sum_{i=0}^{I-1}\frac{\sigma^2c^2\eta_{\tau,i}^3\rho}{16L^2b}\\&\leq-\sum_{i=0}^{I-1}\frac{3\eta_i}{2\rho}\mathbb{E}\|\bar{e}_{\tau,i}\|^2+\frac{8I(I-1)}{\rho}\sum_{\ell=1}^{I}\eta_\ell\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)}-\bar{\nu}_{\tau,\ell}\|^2\\&\quad+\sum_{i=0}^{I-1}\frac{\eta_{\tau,i}\rho}{8}\mathbb{E}\|\tilde{d}_{\tau,i}\|^2+\sum_{i=0}^{I-1}\frac{\sigma^2c^2\eta_{\tau,i}^3\rho}{16L^2b}\end{aligned}$$

By definition, we have $\bar{e}_{\tau,0}=e_{\tau-1}$ and $\bar{e}_{\tau,I}=e_\tau$, then we get the results in the lemma by replacing τ by $\tau+1$. □

8.7.6 Descent in Potential Function

We define the potential function as follows:

$$\Phi_\tau\coloneqq f(\tilde{x}_\tau)+\frac{K\rho}{64L^2}\frac{\|e_\tau\|^2}{\eta_{\tau-1,I-1}}. \tag{8.19}$$

Next, we characterize the descent in the potential function.

Lemma 182. *For any $\tau \geq 0$, we have:*

$$\mathbb{E}[\Phi_{\tau+1} - \Phi_\tau] \leq -\sum_{i=0}^{I-1}\left(\frac{5\rho\eta_{\tau+1,i}}{8} - \frac{\eta_{\tau+1,i}^2 L}{2}\right)\mathbb{E}\|\tilde{d}_i\|^2 - \frac{1}{2\rho}\sum_{i=0}^{I-1}\eta_{\tau+1,i}\mathbb{E}\|\bar{e}_{\tau+1,i}\|^2 + \frac{\sigma^2 c^2 \rho}{16L^2 b}\sum_{i=0}^{I-1}\eta_{\tau+1,i}^3 + \frac{8I(I-1)}{\rho}\sum_{i=1}^{I}\eta_{\tau+1,i}\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau+1,i}^{(k)} - \bar{\nu}_{\tau+1,i}\|^2,$$

where the expectation is w.r.t the stochasticity of the algorithm.

Proof. We can the inequality in the lemma by combining Lemma 178 and Lemma 181

□

8.7.7 Accumulated Gradient Error

In this subsection, we bound the gradient consensus error given by term $\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,i}^{(k)} - \bar{\nu}_{\tau,i}\|^2$.

Lemma 183. *For $i \geq 1$ and $\alpha_i < 1$, we have:*

$$\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,i}^{(k)} - \bar{\nu}_{\tau,i}\|^2 \leq (1+\frac{1}{I})\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{\tau,i-1}^{(k)} - \bar{\nu}_{\tau,i-1}\right\|^2 + 8KIL^2\eta_{\tau,i-1}^2\mathbb{E}\|\tilde{d}_{\tau,i-1}\|^2 + \frac{8KI\sigma^2c^2\eta_{\tau,i-1}^4}{b} + 16KI\zeta^2c^2\eta_{\tau,i-1}^4 + \frac{96I^2L^2}{\rho^2}\sum_{\ell=1}^{i-1}\eta_{\tau,\ell^2}\sum_{k=1}^{K}\mathbb{E}\|\nu_{\tau,\ell}^{(k)} - \bar{\nu}_{\tau,\ell}\|^2$$

where the expectation is w.r.t. the stochasticity of the algorithm.

Proof. By the update rule of $\nu_i^{(k)}$ (we omit the global epoch step for convenience), we have:

$$\begin{aligned}&\mathbb{E}\|\nu_i^{(k)} - \bar{\nu}_i\|^2\\ &= \mathbb{E}\left\|\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) + (1-\alpha_i)\Big(\nu_{i-1}^{(k)} - \nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)})\Big)\right.\end{aligned}$$

$$
\begin{aligned}
&\quad - \left(\frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_i^{(j)};\mathcal{B}_i^{(j)}) + (1-\alpha_i)\Big(\bar{\nu}_{i-1} - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\Big)\right)\Bigg\|^2 \\
&= \mathbb{E}\Bigg\|(1-\alpha_i)\Big(\nu_{i-1}^{(k)} - \bar{\nu}_{i-1}\Big) + \nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_i^{(j)};\mathcal{B}_i^{(j)}) \\
&\qquad\qquad - (1-\alpha_i)\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\right)\Bigg\|^2 \\
&\le (1+\beta)(1-\alpha_i)^2\mathbb{E}\left\|\nu_{i-1}^{(k)} - \bar{\nu}_{i-1}\right\|^2 + \left(1+\frac{1}{\beta}\right)\mathbb{E}\Bigg\|\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_i^{(j)};\mathcal{B}_i^{(j)}) \\
&\qquad\qquad - (1-\alpha_i)\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\right)\Bigg\|^2, \qquad (8.20)
\end{aligned}
$$

where the last inequality uses Proposition 170.

Next, we consider the second term:

$$
\begin{aligned}
&\mathbb{E}\Bigg\|\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_i^{(j)};\mathcal{B}_i^{(j)}) \\
&\qquad\qquad - (1-\alpha_i)\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\right)\Bigg\|^2 \\
&\overset{(a)}{\le} 2\mathbb{E}\Bigg\|\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_i^{(j)};\mathcal{B}_i^{(j)}) \\
&\qquad\qquad - \left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\right)\Bigg\|^2 \\
&\qquad\qquad + 2\alpha_i^2\mathbb{E}\Bigg\|\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\Bigg\|^2 \\
&\overset{(b)}{\le} 2\mathbb{E}\Bigg\|\Big(\nabla f^{(k)}(x_i^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)})\Big)\Bigg\|^2 \\
&\qquad\qquad + 2\alpha_i^2\mathbb{E}\Bigg\|\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\Bigg\|^2 \\
&\overset{(c)}{\le} 2L^2\mathbb{E}\left\|x_i^{(k)} - x_{i-1}^{(k)}\right\|^2 + 2\alpha_i^2\mathbb{E}\Bigg\|\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\Bigg\|^2, \qquad (8.21)
\end{aligned}
$$

where inequality (a) uses Proposition 170; inequality (b) uses Proposition 171; inequality

(c) uses the smoothness assumption.

Next, we consider the second term in (8.21) above, we have

$$
\begin{aligned}
&\mathbb{E}\left\|\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)})\right\|^2 \\
&= \mathbb{E}\Bigg\|\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\right) \\
&\qquad - \frac{1}{K}\sum_{j=1}^{K}\left(\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)}) - \nabla f^{(j)}(x_{i-1}^{(j)})\right) + \nabla f^{(k)}(x_{i-1}^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)})\Bigg\|^2 \\
&\leq 2\mathbb{E}\Bigg\|\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\right) \\
&\qquad\qquad - \frac{1}{K}\sum_{j=1}^{K}\left(\nabla f^{(j)}(x_{i-1}^{(j)};\mathcal{B}_i^{(j)}) - \nabla f^{(j)}(x_{i-1}^{(j)})\right)\Bigg\|^2 \\
&\qquad\qquad\qquad + 2\mathbb{E}\left\|\nabla f^{(k)}(x_{i-1}^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)})\right\|^2 \\
&\overset{(a)}{\leq} 2\mathbb{E}\left\|\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\right)\right\|^2 + 2\mathbb{E}\left\|\nabla f^{(k)}(x_{i-1}^{(k)}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)})\right\|^2 \\
&\leq 2\mathbb{E}\left\|\left(\nabla f^{(k)}(x_{i-1}^{(k)};\mathcal{B}_i^{(k)}) - \nabla f^{(k)}(x_{i-1}^{(k)})\right)\right\|^2 + 4\mathbb{E}\left\|\nabla f^{(k)}(\tilde{x}_{i-1}) - \nabla f(\tilde{x}_{i-1})\right\|^2 \\
&\qquad + 8\mathbb{E}\left\|\nabla f^{(k)}(x_{i-1}^{(k)}) - \nabla f^{(k)}(\tilde{x}_{i-1})\right\|^2 + 8\mathbb{E}\left\|\nabla f(\tilde{x}_{i-1}) - \frac{1}{K}\sum_{j=1}^{K}\nabla f^{(j)}(x_{i-1}^{(j)})\right\|^2 \\
&\overset{(b)}{\leq} \frac{2\sigma^2}{b} + \frac{4}{K}\sum_{j=1}^{K}\mathbb{E}\|\nabla f^{(k)}(\tilde{x}_{i-1}) - \nabla f^{(j)}(\bar{x}_{i-1})\|^2 \\
&\qquad + 8L^2\mathbb{E}\|x_{i-1}^{(k)} - \tilde{x}_{i-1}\|^2 + \frac{8L^2}{K}\sum_{j=1}^{K}\mathbb{E}\|x_{i-1}^{(j)} - \tilde{x}_{i-1}\|^2 \\
&\overset{(c)}{\leq} \frac{2\sigma^2}{b} + 4\zeta^2 + 8L^2\mathbb{E}\|x_{i-1}^{(k)} - \tilde{x}_{i-1}\|^2 + \frac{8L^2}{K}\sum_{j=1}^{K}\mathbb{E}\|x_{i-1}^{(j)} - \tilde{x}_{i-1}\|^2, \qquad (8.22)
\end{aligned}
$$

where inequality (a) uses Proposition 171; inequality (b) utilizes bounded variance assumption; (c) uses the bounded heterogeneity assumption. Finally, substituting (8.22) and (8.21)

into (8.20) and sum over all K workers, we get

$$\begin{aligned}
&\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2\\
&\leq (1-\alpha_i)^2(1+\beta)\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+2L^2\left(1+\frac{1}{\beta}\right)\sum_{k=1}^{K}\mathbb{E}\|x_i^{(k)}-x_{i-1}^{(k)}\|^2\\
&\qquad+\frac{4K\sigma^2}{b}\left(1+\frac{1}{\beta}\right)\alpha_i^2+8K\zeta^2\left(1+\frac{1}{\beta}\right)\alpha_i^2+32L^2\left(1+\frac{1}{\beta}\right)\alpha_i^2\sum_{k=1}^{K}\mathbb{E}\|x_{i-1}^{(k)}-\tilde{x}_{i-1}\|^2
\end{aligned}$$

where the second inequality uses claim 3 of the Lemma 175. For the term $\sum_{k=1}^{K}\mathbb{E}\|x_i^{(k)}-x_{i-1}^{(k)}\|^2$, we have:

$$\begin{aligned}
\sum_{k=1}^{K}\mathbb{E}\|x_i^{(k)}-x_{i-1}^{(k)}\|^2&\leq 2\sum_{k=1}^{K}\mathbb{E}\|x_i^{(k)}-\tilde{x}_i-(x_{i-1}^{(k)}-\tilde{x}_{i-1})\|^2+2\sum_{k=1}^{K}\mathbb{E}\|\tilde{x}_i-\tilde{x}_{i-1}\|^2\\
&\leq 4\sum_{k=1}^{K}\mathbb{E}\|x_i^{(k)}-\tilde{x}_i\|^2+4\sum_{k=1}^{K}\mathbb{E}\|x_{i-1}^{(k)}-\tilde{x}_{i-1}\|^2+2K\eta_{i-1}^2\mathbb{E}\|\tilde{d}_{i-1}\|^2
\end{aligned}$$

So we have:

$$\begin{aligned}
&\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2\\
&\leq (1-\alpha_i)^2(1+\beta)\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+4KL^2\eta_{i-1}^2\left(1+\frac{1}{\beta}\right)\mathbb{E}\|\tilde{d}_{i-1}\|^2\\
&\qquad+8L^2\left(1+\frac{1}{\beta}\right)\sum_{k=1}^{K}\mathbb{E}\|x_i^{(k)}-\tilde{x}_i\|^2\\
&\qquad+\frac{4K\sigma^2}{b}\left(1+\frac{1}{\beta}\right)\alpha_i^2+8K\zeta^2\left(1+\frac{1}{\beta}\right)\alpha_i^2+40L^2\left(1+\frac{1}{\beta}\right)\sum_{k=1}^{K}\mathbb{E}\|x_{i-1}^{(k)}-\tilde{x}_{i-1}\|^2
\end{aligned}$$

Next using Lemma 176, we have:

$$\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2$$

$$
\begin{aligned}
&\leq (1-\alpha_i)^2(1+\beta)\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+4KL^2\eta_{i-1}^2\left(1+\frac{1}{\beta}\right)\mathbb{E}\|\tilde{d}_{i-1}\|^2\\
&\qquad+\frac{8L^2}{\rho^2}\left(1+\frac{1}{\beta}\right)\sum_{k=1}^{K}\mathbb{E}\|z_i^{(k)}-\bar{z}_i\|^2\\
&\qquad+\frac{4K\sigma^2}{b}\left(1+\frac{1}{\beta}\right)\alpha_i^2+8K\zeta^2\left(1+\frac{1}{\beta}\right)\alpha_i^2+\frac{40L^2}{\rho^2}\left(1+\frac{1}{\beta}\right)\sum_{k=1}^{K}\mathbb{E}\|z_{i-1}^{(k)}-\bar{z}_{i-1}\|^2\\
&\leq (1-\alpha_i)^2(1+\beta)\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+4KL^2\eta_{i-1}^2\left(1+\frac{1}{\beta}\right)\mathbb{E}\|\tilde{d}_{i-1}\|^2\\
&\qquad+\frac{4K\sigma^2}{b}\left(1+\frac{1}{\beta}\right)\alpha_i^2+8K\zeta^2\left(1+\frac{1}{\beta}\right)\alpha_i^2\\
&\qquad\quad+\frac{48L^2}{\rho^2}\left(1+\frac{1}{\beta}\right)(I-1)\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2\\
&\leq (1+\frac{1}{I})\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+8KIL^2\eta_{i-1}^2\mathbb{E}\|\tilde{d}_{i-1}\|^2+\frac{8KI\sigma^2c^2\eta_{i-1}^4}{b}\\
&\qquad+16KI\zeta^2c^2\eta_{i-1}^4+\frac{96I^2L^2}{\rho^2}\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2 \qquad (8.23)
\end{aligned}
$$

In the last inequality, we choose $\beta = 1/I$, then we have $(1+1/\beta) \leq (1+I) \leq 2I$, we also use the fact that $(1-\alpha_i)^2 < 1$ and $a_i = c\eta_{i-1}^2 < 1$. This completes the proof. □

Lemma 184. *For $\eta_i \leq \frac{\rho}{48LK^{0.5}I^2}$, then we have*

$$\frac{I^2}{\rho}\sum_{i=1}^{I}\eta_i\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2 \leq \frac{\rho}{84}\sum_{i=0}^{I-1}\eta_i\mathbb{E}\|\tilde{d}_i\|^2+\left(\frac{\rho\sigma^2c^2}{84bL^2}+\frac{\rho\zeta^2c^2}{42L^2}\right)\sum_{i=0}^{I-1}\eta_i^3$$

Proof. By Lemma 183 (we omit the global epoch number for convenience) we have:

$$\begin{aligned}
\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2 &\leq (1+\frac{1}{I})\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+8KIL^2\eta_{i-1}^2\mathbb{E}\|\tilde{d}_{i-1}\|^2+\frac{8KI\sigma^2c^2\eta_{i-1}^4}{b}\\
&\quad+16KI\zeta^2c^2\eta_{i-1}^4+\frac{96I^2L^2}{\rho^2}\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2\\
&\leq (1+\frac{1}{I})\sum_{k=1}^{K}\mathbb{E}\left\|\nu_{i-1}^{(k)}-\bar{\nu}_{i-1}\right\|^2+\frac{\sqrt{K}L\rho\eta_{i-1}}{6I}\mathbb{E}\|\tilde{d}_{i-1}\|^2+\frac{\sqrt{K}\rho\sigma^2c^2\eta_{i-1}^3}{6ILb}\\
&\quad+\frac{\sqrt{K}\rho\zeta^2c^2\eta_{i-1}^3}{3IL}+\frac{96I^2L^2}{\rho^2}\sum_{\ell=1}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2, \qquad (8.24)
\end{aligned}$$

where in the second inequality, we use the condition that $\eta_i \leq \frac{\rho}{48LK^{0.5}I^2}$. Applying (8.24) recursively from 1 to i. We have:

$$\begin{aligned}
\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2 &\leq \frac{\sqrt{K}L\rho}{6I}\sum_{\ell=0}^{i-1}\left(1+\frac{1}{I}\right)^{i-1-\ell}\eta_\ell\mathbb{E}\|\tilde{d}_\ell\|^2+\frac{\sqrt{K}\rho\sigma^2c^2}{6ILb}\sum_{\ell=0}^{i-1}\left(1+\frac{1}{I}\right)^{i-1-\ell}\eta_\ell^3\\
&\quad+\frac{\sqrt{K}\rho\zeta^2c^2}{3IL}\sum_{\ell=0}^{i-1}\left(1+\frac{1}{I}\right)^{i-1-\ell}\eta_\ell^3\\
&\quad+\frac{96L^2I^2}{\rho^2}\sum_{\ell=0}^{i-1}\left(1+\frac{1}{I}\right)^{i-1-\ell}\sum_{\bar{\ell}=0}^{\ell}\eta_{\bar{\ell}}^2\sum_{k=1}^{K}\mathbb{E}\|\nu_{\bar{\ell}}^{(k)}-\bar{\nu}_{\bar{\ell}}\|^2\\
&\overset{(a)}{\leq}\frac{\sqrt{K}L\rho}{6I}\left(1+\frac{1}{I}\right)^{I}\sum_{\ell=0}^{i-1}\eta_\ell\mathbb{E}\|\tilde{d}_\ell\|^2+\frac{\sqrt{K}\rho\sigma^2c^2}{6ILb}\left(1+\frac{1}{I}\right)^{I}\sum_{\ell=0}^{i-1}\eta_\ell^3\\
&\quad+\frac{\sqrt{K}\rho\zeta^2c^2}{3IL}\left(1+\frac{1}{I}\right)^{I}\sum_{\ell=0}^{i-1}\eta_\ell^3+\frac{96L^2I^3}{\rho^2}\left(1+\frac{1}{I}\right)^{I}\sum_{\bar{\ell}=0}^{i-1}\eta_{\bar{\ell}}^2\sum_{k=1}^{K}\mathbb{E}\|\nu_{\bar{\ell}}^{(k)}-\bar{\nu}_{\bar{\ell}}\|^2\\
&\overset{(b)}{\leq}\frac{\sqrt{K}L\rho}{2I}\sum_{\ell=0}^{i-1}\eta_\ell\mathbb{E}\|\tilde{d}_\ell\|^2+\frac{\sqrt{K}\rho\sigma^2c^2}{2ILb}\sum_{\ell=0}^{i-1}\eta_\ell^3+\frac{\sqrt{K}\rho\zeta^2c^2}{IL}\sum_{\ell=0}^{i-1}\eta_\ell^3
\end{aligned}$$

$$+\frac{288L^2I^3}{\rho^2}\sum_{\ell=0}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2, \tag{8.25}$$

where inequality (a) is by the fact that $1+1/I>1$ and $i-1-\ell\leq I$ for $i\in[I]$ and $\ell\in[i]$ and inequality (b) is because that $(1+1/I)^I\leq \mathrm{e}<3$.

Next, multiplying both sides of (8.25) by η_i and summing over $i=1$ to I:

$$\begin{aligned}
\sum_{i=1}^{I}\eta_i\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2 &\leq \frac{\sqrt{K}L\rho}{2I}\sum_{i=1}^{I}\eta_i\sum_{\ell=0}^{i-1}\eta_\ell\mathbb{E}\|\tilde{d}_\ell\|^2+\frac{\sqrt{K}\rho\sigma^2c^2}{2ILb}\sum_{i=1}^{I}\eta_i\sum_{\ell=0}^{i-1}\eta_\ell^3\\
&\quad+\frac{\sqrt{K}\rho\zeta^2c^2}{IL}\sum_{i=1}^{I}\eta_i\sum_{\ell=0}^{i-1}\eta_\ell^3+\frac{288L^2I^3}{\rho^2}\sum_{i=1}^{I}\eta_i\sum_{\ell=0}^{i-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2\\
&\overset{(a)}{\leq}\frac{\sqrt{K}L\rho}{2I}\left(\sum_{i=1}^{I}\eta_i\right)\sum_{\ell=0}^{I-1}\eta_\ell\mathbb{E}\|\tilde{d}_\ell\|^2+\left(\frac{\sqrt{K}\rho\sigma^2c^2}{2ILb}+\frac{\sqrt{K}\rho\zeta^2c^2}{IL}\right)\left(\sum_{i=1}^{I}\eta_i\right)\sum_{\ell=0}^{I-1}\eta_\ell^3\\
&\quad+\frac{288L^2I^3}{\rho^2}\left(\sum_{i=1}^{I}\eta_i\right)\sum_{\ell=0}^{I-1}\eta_\ell^2\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2\\
&\overset{(b)}{\leq}\frac{\rho^2}{96I^2}\sum_{i=0}^{I-1}\eta_i\mathbb{E}\|\tilde{d}_i\|^2+\left(\frac{\rho^2\sigma^2c^2}{96I^2L^2b}+\frac{\rho^2\zeta^2c^2}{48I^2L^2}\right)\sum_{i=0}^{I-1}\eta_i^3+\frac{1}{8}\sum_{\ell=1}^{I-1}\eta_\ell\sum_{k=1}^{K}\mathbb{E}\|\nu_\ell^{(k)}-\bar{\nu}_\ell\|^2
\end{aligned}$$

where inequality (a) uses the fact that $i\leq I$ and (b) uses that we choose $\eta_i\leq\rho/(48LK^{0.5}I^2)$. Rearranging the terms we have:

$$\frac{7}{8}\sum_{i=1}^{I}\eta_i\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2\leq\frac{\rho^2}{96I^2}\sum_{i=0}^{I-1}\eta_i\mathbb{E}\|\tilde{d}_i\|^2+\left(\frac{\rho^2\sigma^2c^2}{96I^2L^2b}+\frac{\rho^2\zeta^2c^2}{48I^2L^2}\right)\sum_{i=0}^{I-1}\eta_i^3$$

Multiplying $8I^2/(7K\rho)$ on both sides, we have:

$$\frac{I^2}{\rho}\sum_{i=1}^{I}\eta_i\sum_{k=1}^{K}\mathbb{E}\|\nu_i^{(k)}-\bar{\nu}_i\|^2\leq\frac{\rho}{84}\sum_{i=0}^{I-1}\eta_i\mathbb{E}\|\tilde{d}_i\|^2+\left(\frac{\rho\sigma^2c^2}{84L^2b}+\frac{\rho\zeta^2c^2}{42L^2}\right)\sum_{i=0}^{I-1}\eta_i^3$$

This completes the proof. □

8.7.8 Proof of the Main Convergence Theorem

In this subsection, we prove Theorem 168 and Corollary 5.7. To prove Theorem 168, we firstly show the following theorem hold:

Theorem 185. *Choosing the parameters as* $\kappa = \frac{K^{2/3}\rho}{L}$, $c = \frac{96L^2}{K\rho^2} + \frac{\rho}{72\kappa^3 LK^{0.5}I^2}$, $w_t = \max\{2, 48^3 I^6 K^{7/2} - t\}$, *and choose* $\eta_t = \frac{\kappa}{(w_t+t)^{1/3}}$, *then we have:*

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbb{E}\|\tilde{d}_t\|^2 + \frac{1}{\rho^2}\mathbb{E}\|\bar{e}_t\|^2\right) \\
&\leq \left[\frac{96LI^2}{\rho^2 T} + \frac{2L}{\rho^2 K^{2/3}T^{2/3}}\right](f(x_0) - f^*) + \left[\frac{72I^4}{b_1\rho^2 T} + \frac{3I^2}{2b_1\rho^2 K^{2/3}T^{2/3}}\right]\sigma^2 \\
&\quad + \frac{192^2}{\rho^2} \times \left(\frac{48I^2}{T} + \frac{1}{K^{2/3}T^{2/3}}\right) \times \left(\frac{\sigma^2}{4b} + \frac{2\zeta^2}{21}\right)\log(T+1).
\end{aligned}$$

Proof. By definition, we have $\eta_t \leq \eta_0 < \kappa/w_0^{1/3} = \rho/48LK^{0.5}I^2$, then $c = \frac{L^2}{K\rho^2}\left(96 + \frac{1}{72K^{1.5}I^2}\right) \leq \frac{192L^2}{K\rho^2}$ and: $c\eta_t^2 \leq c\eta_0^2 = \frac{192L^2}{K\rho^2} * \frac{\rho^2}{48^2L^2KI^4} < 1$, so we have $\alpha_t < 1$, then the conditions of Lemma 182-Lemma 184 are satisfied.

Firstly, substitute the gradient consensus error in Lemma 184 to Lemma 182, we can write the descent of potential function as:

$$\begin{aligned}
\mathbb{E}[\Phi_{\tau+1} - \Phi_\tau] &\leq -\sum_{i=0}^{I-1}\left(\frac{5\rho\eta_{\tau+1,i}}{8} - \frac{\eta_{\tau+1,i}^2 L}{2}\right)\mathbb{E}\|\tilde{d}_{\tau+1,i}\|^2 - \frac{1}{2\rho}\sum_{i=0}^{I-1}\eta_{\tau+1,i}\mathbb{E}\|\bar{e}_{\tau+1,i}\|^2 \\
&\quad + \frac{\sigma^2c^2\rho}{16L^2b}\sum_{i=0}^{I-1}\eta_{\tau+1,i}^3 + \frac{2\rho}{21}\sum_{i=0}^{I-1}\eta_{\tau+1,i}\mathbb{E}\|\tilde{d}_{\tau+1,i}\|^2 + \left(\frac{2\rho\sigma^2c^2}{21L^2b} + \frac{4\rho\zeta^2c^2}{21L^2}\right)\sum_{i=0}^{I-1}\eta_{\tau+1,i}^3 \\
&\overset{(a)}{\leq} -\sum_{i=0}^{I-1}\frac{\rho\eta_i}{2}\mathbb{E}\|\tilde{d}_{\tau+1,i}\|^2 - \frac{1}{2\rho}\sum_{i=0}^{I-1}\eta_{\tau+1,i}\mathbb{E}\|\bar{e}_{\tau+1,i}\|^2 + \left(\frac{\rho\sigma^2c^2}{4L^2b} + \frac{\rho\zeta^2c^2}{4L^2}\right)\sum_{i=0}^{I-1}\eta_{\tau+1,i}^3,
\end{aligned}$$

where (a) follows from the fact that $\eta_i \leq \frac{\rho}{48LK^{0.5}I^2} \leq \frac{\rho}{48L}$.

Suppose we denote $T = EI$, and $t = \tau I + i$ for $t \geq 0$ and $\tau \geq 0$. Then we have $\eta_t = \eta_{\tau+1,i}$, $\tilde{d}_t = \tilde{d}_{\tau+1,i}$, $\bar{e}_t = \bar{e}_{\tau+1,i}$. In particular, we denote $\eta_{-1} = \eta_0$ for convenience.

Then we sum the above inequality for τ from 0 to $E-1$, and get:

$$\mathbb{E}[\Phi_E - \Phi_0] \leq -\sum_{t=0}^{T-1}\left(\frac{\rho\eta_t}{2}\right)\mathbb{E}\|\tilde{d}_t\|^2 - \sum_{t=0}^{T-1}\frac{\eta_t}{2\rho}\mathbb{E}\|\bar{e}_t\|^2 + \left(\frac{\rho\sigma^2c^2}{4L^2b} + \frac{\rho\zeta^2c^2}{4L^2}\right)\sum_{t=0}^{T}\eta_t^3,$$

Rearranging terms, we get:

$$\begin{aligned}\sum_{t=1}^{T}\left(\frac{\rho\eta_t}{2}\mathbb{E}\|\tilde{d}_t\|^2 + \frac{\eta_t}{2\rho}\mathbb{E}\|\bar{e}_t\|^2\right) &\leq \mathbb{E}[\Phi_0 - \Phi_E] + \left(\frac{\rho\sigma^2c^2}{4L^2b} + \frac{\rho\zeta^2c^2}{4L^2}\right)\sum_{t=0}^{T-1}\eta_t^3 \\ &\overset{(a)}{\leq} f(x_0) - f^* + \frac{K\rho}{64L^2}\frac{\mathbb{E}\|e_0\|^2}{\eta_0} + \left(\frac{\rho\sigma^2c^2}{4L^2b} + \frac{\rho\zeta^2c^2}{4L^2}\right)\sum_{t=0}^{T-1}\eta_t^3 \\ &\overset{(b)}{\leq} f(x_0) - f^* + \frac{\sigma^2\rho}{64b_1L^2\eta_0} + \left(\frac{\rho\sigma^2c^2}{4L^2b} + \frac{\rho\zeta^2c^2}{4L^2}\right)\sum_{t=0}^{T-1}\eta_t^3, \quad (8.26)\end{aligned}$$

where (a) follows from the fact that $f^* \leq \Phi_E$ and (b) results from application of Lemma 179 and b is the minibatch size at the first iteration.

Next for the last term of the (8.26) above, we have:

$$\sum_{t=0}^{T-1}\eta_t^3 = \sum_{t=0}^{T-1}\frac{\kappa^3}{w_t + t} \overset{(a)}{\leq} \sum_{t=0}^{T-1}\frac{\kappa^3}{1+t} \overset{(b)}{\leq} \kappa^3\ln(T+1). \quad (8.27)$$

where inequality (a) above follows from the fact that we have $w_t > 1$ and inequality (b) follows from the application of Proposition 172.

Substituting (8.27) in (8.26), multiplying both sides by $2/(\rho\eta_T T)$ and using the fact

that η_t is non-increasing in t we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbb{E}\|\tilde{d}_t\|^2+\frac{1}{\rho^2}\mathbb{E}\|\bar{e}_t\|^2\right)\leq\frac{2(f(x_0)-f^*)}{\rho\eta_T T}+\frac{1}{\eta_T T}\frac{\sigma^2}{32b_1L^2\eta_0}+\frac{\kappa^3}{\eta_T T}\left(\frac{\sigma^2c^2}{4bL^2}+\frac{2\zeta^2c^2}{21L^2}\right)\ln(T+1). \tag{8.28}$$

Now considering each term of (8.28) above separately. For the first term:

$$\frac{1}{\eta_T T}=\frac{(w_T+T)^{1/3}}{\kappa T}\overset{(a)}{\leq}\frac{w_T^{1/3}}{\kappa T}+\frac{1}{\kappa T^{2/3}}\overset{(b)}{\leq}\frac{48LI^2K^{0.5}}{\rho T}+\frac{L}{\rho K^{2/3}T^{2/3}}. \tag{8.29}$$

where inequality (a) follows from identity $(x+y)^{1/3}\leq x^{1/3}+y^{1/3}$ and inequality (b) follows from the definition of κ and w_T Similarly, for the second term of (8.28), we have from the definition of η_0 and η_T:

$$\begin{aligned}\frac{1}{\eta_T T}\frac{\sigma^2}{32bL^2\eta_0}&\leq\left(\frac{48LK^{0.5}I^2}{\rho T}+\frac{L}{\rho K^{2/3}T^{2/3}}\right)\times\frac{\sigma^2}{32b_1L^2}\times\frac{w_0^{1/3}}{\kappa}\\&\leq\left(\frac{48LK^{0.5}I^2}{\rho T}+\frac{L}{\rho K^{2/3}T^{2/3}}\right)\times\frac{\sigma^2}{32b_1L^2}\times\frac{48LK^{0.5}I^2}{\rho}\\&\leq\frac{72KI^4}{b_1\rho^2T}\sigma^2+\frac{3K^{0.5}I^2}{2b_1\rho^2K^{2/3}T^{2/3}}\sigma^2.\end{aligned} \tag{8.30}$$

Finally, for the last term in (8.28) above, we have from the definition of the stepsize, η_t,

$$\begin{aligned}&\frac{\kappa^3c^2}{\eta_T TL^2}\left(\frac{\sigma^2}{4b}+\frac{2\zeta^2}{21}\right)\ln(T+1)\\&\leq\left(\frac{48LK^{0.5}I^2}{\rho T}+\frac{L}{\rho K^{2/3}T^{2/3}}\right)\times\frac{192^2}{L\rho}\times\left(\frac{\sigma^2}{4b}+\frac{2\zeta^2}{21}\right)\log(T+1)\\&\leq\frac{192^2}{\rho^2}\times\left(\frac{48K^{0.5}I^2}{T}+\frac{1}{K^{2/3}T^{2/3}}\right)\times\left(\frac{\sigma^2}{4b}+\frac{2\zeta^2}{21}\right)\log(T+1).\end{aligned} \tag{8.31}$$

Finally, substituting the bounds obtained in (8.29), (8.30) and (8.31) into (8.28), we get

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbb{E}\|\tilde{d}_t\|^2+\frac{1}{\rho^2}\mathbb{E}\|\bar{e}_t\|^2\right)\\
&\leq\left[\frac{96LK^{0.5}I^2}{\rho^2T}+\frac{2L}{\rho^2K^{2/3}T^{2/3}}\right](f(x_0)-f^*)+\left[\frac{72KI^4}{b_1\rho^2T}+\frac{3K^{0.5}I^2}{2b_1\rho^2K^{2/3}T^{2/3}}\right]\sigma^2\\
&\quad+\frac{192^2}{\rho^2}\times\left(\frac{48K^{0.5}I^2}{T}+\frac{1}{K^{2/3}T^{2/3}}\right)\times\left(\frac{\sigma^2}{4b}+\frac{2\zeta^2}{21}\right)\log(T+1).
\end{aligned}$$

This completes the proof of the theorem. □

Now we are ready to show Theorem 168. Firstly notice that:

$$\frac{\mathcal{G}_t}{\rho^2}=\frac{1}{\eta_t^2}||\tilde{x}_t-\tilde{x}_{t+1}||^2+\frac{1}{\rho^2}||\bar{\nu}_t-\nabla f(\tilde{x}_t)||^2=||\tilde{d}_t||^2+\frac{1}{\rho^2}||\bar{e}_t||^2$$

Combine with Theorem 185, we have:

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathcal{G}_t]\leq&\left[\frac{96LK^{0.5}I^2}{T}+\frac{2L}{K^{2/3}T^{2/3}}\right](f(x_0)-f^*)+\left[\frac{72KI^4}{b_1T}+\frac{3K^{0.5}I^2}{2b_1K^{2/3}T^{2/3}}\right]\sigma^2\\
&+192^2\times\left(\frac{48K^{0.5}I^2}{T}+\frac{1}{K^{2/3}T^{2/3}}\right)\times\left(\frac{\sigma^2}{4b}+\frac{2\zeta^2}{21}\right)\log(T+1).
\end{aligned}$$

Remark 186. *For the measure $\mathcal{G}_t$, we discuss its intuition under both the unconstrained and constrained case. First, for unconstrained case,* i.e. *when $\mathcal{X}=R^d$, we have:*

$$\begin{aligned}
&||\nabla f(\tilde{x}_{\tau,i})||/||H_\tau||=||H_\tau\times H_\tau^{-1}\nabla f(\tilde{x}_{\tau,i})||/||H_\tau||\leq||H_\tau^{-1}\nabla f(\tilde{x}_{\tau,i})||\\
&=||H_\tau^{-1}\nabla f(\tilde{x}_{\tau,i})-H_\tau^{-1}\bar{\nu}_{\tau,i}+H_\tau^{-1}\bar{\nu}_{\tau,i}||\leq||H_\tau^{-1}\nabla f(\tilde{x}_{\tau,i})-H_\tau^{-1}\bar{\nu}_{\tau,i}||+||H_\tau^{-1}\bar{\nu}_{\tau,i}||
\end{aligned}$$

$$\leq \frac{1}{\rho}||\bar{\nu}_{\tau,i} - \nabla f(\tilde{x}_{\tau,i})|| + \frac{1}{\eta_{\tau,i}}||\tilde{x}_{\tau,i} - \tilde{x}_{\tau,i+1}|| \leq \sqrt{2}\sqrt{\mathcal{G}_{\tau,i}}/\rho$$

In the last inequality, we use Jensen inequality, and in the second last inequality, we use Assumption 166 and the fact that $\tilde{x}_{\tau,i+1} = x_{\tau,0} + H_\tau^{-1}\bar{z}_{\tau,i+1}$ *and* $\tilde{x}_{\tau,i} = x_{\tau,0} + H_\tau^{-1}\bar{z}_{\tau,i}$ *and* $\eta_{\tau,i}\bar{\nu}_{\tau,i} = \bar{z}_{\tau,i+1} - \bar{z}_{\tau,i}$ *in the unconstrained case. In other words, we have* $||\nabla f(\tilde{x}_t)||^2 \leq \frac{2||H_\tau||^2}{\rho^2}\mathcal{G}_\tau$. *Note the coefficient of the right-side is an upper bound of the square condition number of* H_τ. *It is common assumption in the analysis of adaptive gradient methods that* H_t *has a finite condition number [31]. In sum, the convergence of our measure* $\mathcal{G}_t$ *means the convergence to a first order stationary point in the unconstrained case.*

Next, for the constrained case, our measure upper bounds the gradient mapping $\frac{1}{\eta_{\tau+1,i}}||x_\tau - x^*_{\tau+1,i}||$, x^*_t *is defined as follows:*

$$x^*_{\tau+1,i} = \arg\min_{x\in\mathcal{X}}\{-\langle x, z^*_{\tau+1,i})\rangle + \frac{1}{2}(x - x_\tau)^T H_\tau (x - x_\tau)\}$$

where $z^*_{\tau+1,i} = \sum_{\ell=0}^{i} -\eta_\ell \nabla f(\tilde{x}_{\tau+1,i})$ *is the accumulation of true gradient. Next follow Lemma 175, we have:*

$$\begin{aligned}
&\|x^*_{\tau+1,i} - \tilde{x}_{\tau+1,i}\| \leq \frac{1}{\rho}\|z^*_{\tau+1,i} - \bar{z}_{\tau+1,i}\| \\
&= \frac{1}{\rho}\|\sum_{l=0}^{i-1} -\eta_{\tau+1,\ell}(\nabla f(\tilde{x}_{\tau+1,\ell}) - \bar{\nu}_{\tau+1,\ell}))\| \overset{(a)}{\leq} \sum_{l=0}^{i-1}\frac{\eta_{\tau+1,\ell}}{\rho}\|\nabla f(\tilde{x}_{\tau+1,\ell}) - \bar{\nu}_{\tau+1,\ell}\|
\end{aligned}$$

where inequality (a) *is due to the triangle inequality. Next we have:*

$$\|x_\tau - x^*_{\tau+1,i}\| = \|x_\tau - \tilde{x}_{\tau+1,i} + \tilde{x}_{\tau+1,i} - x^*_{\tau+1,i}\| \leq \|x_\tau - \tilde{x}_{\tau+1,i}\| + \|\tilde{x}_{\tau+1,i} - x^*_{\tau+1,i}\|$$

$$\leq \|\sum_{l=0}^{i-1}\tilde{d}_{\tau+1,i}\| + \|\tilde{x}_{\tau+1,i} - x^*_{\tau+1,i}\| \leq \sum_{l=0}^{i-1}\left(\|\tilde{d}_{\tau+1,\ell}\| + \frac{\eta_{\tau+1,\ell}}{\rho}\|\nabla f(\tilde{x}_{\tau+1,\ell}) - \bar{\nu}_{\tau+1,\ell}\|\right)$$

By Jensen inequality and the definition of the measure (8.9), we have

$$\|\tilde{d}_t\| + \frac{\eta_t}{\rho}\|\nabla f(\tilde{x}_t) - \bar{\nu}_t\| \leq \frac{\sqrt{2}\eta_t}{\rho}\sqrt{\mathcal{G}_t},$$

So we have

$$\frac{1}{\eta_{\tau+1,i}}\|x_\tau - x^*_{\tau+1,i}\| \leq \frac{\sqrt{2}}{\rho}\sum_{l=0}^{i-1}\frac{\eta_{\tau+1,l}}{\eta_{\tau+1,i}}\sqrt{\mathcal{G}_{\tau+1,l}} \leq \frac{2\sqrt{2}}{\rho}\sum_{l=0}^{i-1}\sqrt{\mathcal{G}_{\tau+1,\ell}},$$

the last inequality is because of Eq. (8.17). In all, when the measure $\mathcal{G}_{\tau+1,\ell} \to 0$, the gradient mapping $\frac{1}{\eta_{\tau+1,i}}\|x_\tau - x^*_{\tau+1,i}\|$ converges to 0.

Corollary 187. *With the hyper-parameters chosen as in Theorem 185. Suppose we set $I = O((T/K^{3.5})^{1/6})$ and use sample minibatch of size $b_1 = O(K^{0.5}I^2)$ in the first step, Then we have:*

$$\mathbb{E}[\mathcal{G}_t] = O\left(\frac{f(x_0) - f^*}{K^{2/3}T^{2/3}}\right) + \tilde{O}\left(\frac{\sigma^2}{K^{2/3}T^{2/3}}\right) + \tilde{O}\left(\frac{\zeta^2}{K^{2/3}T^{2/3}}\right).$$

and to reach an ϵ-stationary point, we need to make $\tilde{O}(K^{-1}\epsilon^{-1.5})$ number of steps and need $\tilde{O}(K^{-0.25}\epsilon^{-1.25})$ number of communication rounds.

Proof. It is straightforward to verify the expression for $\mathbb{E}[\mathcal{G}_t]$ in the corollary by applying Theorem 185 and choosing I and b as corresponding values. As for the gradient and communication complexity of the algorithm. We have the following results: The number of total steps T needed to achieve an ϵ-stationary point, *i.e.* $\tilde{O}(\frac{1}{K^{2/3}T^{2/3}}) = \epsilon$, then the

gradient complexity is $\mathcal{O}(K^{-1}\epsilon^{-3/2})$. Total rounds of communication steps to achieve an ϵ-stationary point is $E = T/I$, as we have $I = \mathrm{O}((T/K^{3.5})^{1/6})$, then $T/I = \tilde{O}(K^{7/12}T^{5/6})$. By the fact that $T = K^{-1}\epsilon^{-3/2}$, we have $E = O(K^{-1/4}\epsilon^{-5/4})$. □

8.8 Experimental Details and Results

In this section, we add additional experiments. In Section A.1, we consider more variants of **FedDA** besides **FedDA-MVR**. More specifically, we consider four variants of **FedDA**. We introduce two cases for the update of the adaptive matrix H_τ in (8.7) and (8.8) and we denote them as case 1 and case 2, similarly, we denote (8.5) and (8.6) as case 1 and case 2 of gradient estimation respectively. So we have four different variants, we denote them as **FedDA-i-j**, for $i, j \in \{1, 2\}$, where i shows the choice of gradient estimation and j shows the choice of adaptive matrix update rule. Note **FedDA-MVR** corresponds to **FedDA-1-1** as we choose Case 1 of gradient estimation and Case 1 of adaptive matrix update in Algorithm 16. We also introduce more details such as the hyper-parameter choices. Then in Section A.2, we perform some ablation studies and compare our FedDA with other baselines in more detail; In Section A.3, we include experiments when we construct heterogeneous dataset from CIFAR10; Finally in Section A.4, we show the form of our FedDA when $I = 1$, i.e. no local steps.

8.8.1 Experimental Results for More Variants of **FedDA**

8.8.1.1 Colorrectal Cancer Survival Prediction with Biomarker Identification

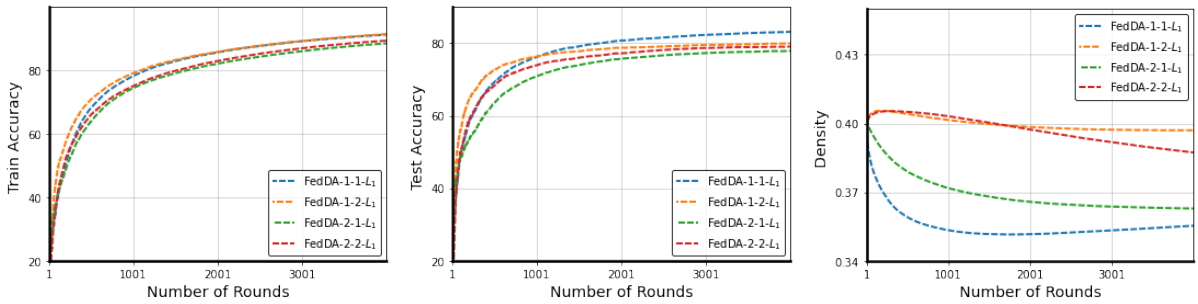


Figure 8.4: Results for the PATHMNIST dataset. Plots show the Train Accuracy, Test Accuracy, Density vs Number of Rounds (E in Algorithm 16) respectively. The post-fix of L_1 means we consider the L_1 constraints.

In this task, for L_1 constraint, we consider the constraint set of $|x|_1 \leq \epsilon$, where x is the model parameter and ϵ is some constant. We use a 4-layer convolutional neural network with 32 filters at each layer. We have 10 clients and run 20000 steps (T), average states with interval 5 (I) and use mini-batch size of 16. Besides, we calculate density with threshold 0.01. For other hyper-parameters, we perform grid search and choose the best setting for each method. More specifically, for the SGD method, we use learning rate 0.01; for the FedDualAvg algorithm, we use local learning rate 0.1, global learning rate 0.1, L_1 constraint 0.01; for our FedDA-MVR, we use learning rate 0.01, w as 100000, c as 5000000, β as 0.999 and τ as 0.01, for the L_1 regularized version FedDA-MVR-L_1, we also add L_1 constraint $\epsilon = 0.01$. For other variants of FedDA: for FedDA-2-1, we use learning rate 0.001, α as 0.9, β as 0.999, τ as 0.01; for FedDA-1-2, we use learning rate 1, w as 10000, c

as 200, β as 0.999, τ as 0.001, L_1 constraint 0.01; for FedDA-2-2, we use learning rate 0.01, α 0.9, β as 0.999, τ as 0.01, L_1 constraint 0.01.

The experimental results for different variants of FedDA is summarized in Figure 8.4. As shown by the plots, all variants of FedDA get good performance, but we find FedDA-MVR (FedDA-1-1) gets most sparse model as measured by the density metric.

8.8.1.2 Splice Site Detection with Biomarker Identification

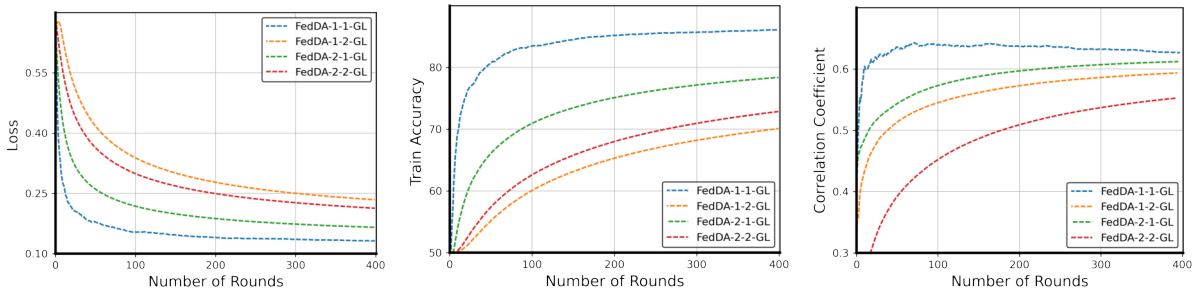


Figure 8.5: Results for the MEMset Donar Dataset. Plots show the Train Loss, Train Accuracy and the correlation coefficient, respectively.

In this task, for group lasso constraint, we consider the constraint set of $\sum_{i=1}^{q} |x_q|_2 \leq \epsilon$, where x is the model parameter, q is the number of groups, x_q denotes a subset of parameters of group q, ϵ is some constant. The MEMset donor data set consists of a training set of 8415 true and 179,438 false human donor sites, and a test set of 4208 true and 89,717 false donor sites. A sequence of a real splice site consists of the last three bases of the exon and the first six bases of the intron. A training sample is a sequence of length of 7 with values in $\{A, C, G, T\}$. The data are available at `http://hollywood.mit.edu/burgelab/maxent/ssdata/`. We follow [3] to create a balanced training set with 5610 true/false samples and an unbalanced validation set with 2805 true/59804 false samples. We have 10 clients

and randomly evenly distribute the training data over 10 clients. We consider the logistic regression model with group lasso constraint and include all the three-way and lower order interactions. In the experiments, we run 2000 steps (T), average states with interval 5 (I) and use mini-batch size of 16. For other hyper-parameters, we perform grid search and choose the best setting for each method. More specifically, for the FedAvg method, we use learning rate 0.1; for the FedDualAvg algorithm with L_1 constraint, we use local learning rate 0.1, global learning rate 0.2, L_1 constraint 0.01; for the FedDualAvg algorithm with group lasso constraint, we use local learning rate 0.1, global learning rate 0.5, L_1 constraint 0.01; for our FedDA-MVR, we use learning rate 0.1, w as 10000, c as 40000, β as 0.999 and τ as 0.01, for the L_1 constrained version FedDA-MVR-L_1, we also add L_1 constraint 0.01, for the group lasso version FedDA-MVR-GL, we also add group lasso constraint of coefficient 0.01. For other variants of FedDA: for FedDA-1-2, we use learning rate 1, w as 10000, c as 200, β as 0.999, τ as 0.001, group lasso constraint 0.01; for FedDA-2-1, we use learning rate 0.001, α as 0.9, β as 0.999, τ as 0.01, group lasso constraint 0.01; for FedDA-2-2, we use learning rate 0.01, α 0.9, β as 0.999, τ as 0.01, group lasso constraint 0.01.

The experimental results for different variants of FedDA is summarized in Figure 8.5. As shown by the plots, all variants of FedDA get good performance, but we find FedDA-MVR (FedDA-1-1) gets the highest test correlation coefficient among all variants. Note that the correlation coefficient is the maximum (among all possible threshold, and we find 0.95 is the best value for all methods) Pearson correlation between the binary random variable of the true class membership and the binary random variable of the predicted class membership.

8.8.1.3 Image Classification Task with CIFAR10 and FEMNIST

In this unconstrained federated image classification task, we use a 4-layer convolutional neural network with 64 filters at each layer. For the FEMNIST dataset, we randomly sample 50 users at each global round. We run 20000 steps (T), average states with interval 5 (I) and use mini-batch size of 16. For other hyper-parameters, we perform grid search and choose the best setting for each method. In the CIFAR10 related experiments, for the SGD method, we use learning rate 0.005; for the FedCM algorithm, we use learning rate 0.01, momentum coefficient α as 0.9; for the FedAdam algorithm, we use local learning rate 0.001, global learning rate 0.002, momentum coefficient 0.9, coefficient for adaptive matrix β as 0.999; for the Local-Adapt algorithm, we use local learning rate 0.001, global learning rate 0.002, momentum coefficient 0.9, coefficient for adaptive matrix β as 0.999; for the Local-AMSGrad algorithm, we use learning rate 0.001, momentum coefficient 0.9, adaptive matrix coefficient 0.999; for the MIME-MVR algorithm, we use learning rate 0.1, w 100, c as 2000; for the STEM algorithm, we use learning rate 0.1, w 100 and c 2000; for our FedDA-MVR, we use learning rate 0.02, w as 10000, c as 1000000, β as 0.999 and τ as 0.01. For other variants of FedDA: for FedDA-2-1, we use learning rate 0.001, α as 0.9, β as 0.999, τ as 0.01; for FedDA-1-2, we use learning rate 1, w as 5000, c as 100, β as 0.999, τ as 0.01; for FedDA-2-2, we use learning rate 0.01, α 0.9, β as 0.999, τ as 0.01.

Then in the FEMNIST experiments, for the SGD method, we use learning rate 0.1; for the FedCM algorithm, we use learning rate 0.1, momentum coefficient α as 0.9; for the FedAdam algorithm, we use local learning rate 0.02, global learning rate 0.04, momentum coefficient 0.9, coefficient for adaptive matrix β as 0.999; for the Local-Adapt algorithm, we

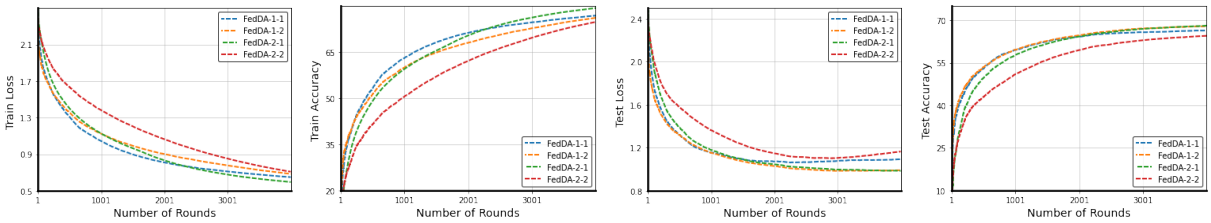


Figure 8.6: Results for CIFAR10 dataset. From left to right, we show Train Loss, Train Accuracy, Test Loss, Test Accuracy $w.r.t$ the number of global rounds (E in Algorithm 16), respectively.

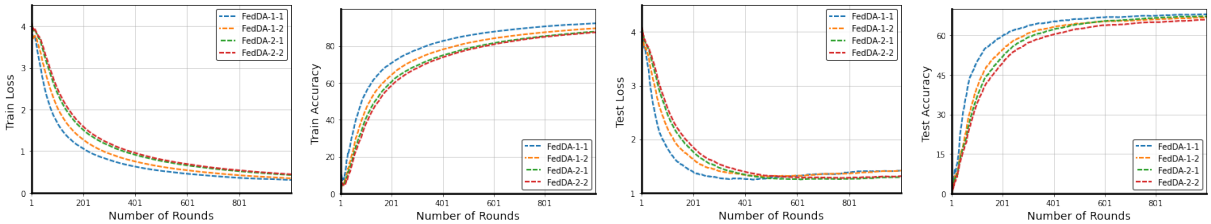


Figure 8.7: Results for FEMNIST dataset. From left to right, we show Train Loss, Train Accuracy, Test Loss, Test Accuracy $w.r.t$ the number of global rounds (E in Algorithm 16), respectively.

use local learning rate 0.02, global learning rate 0.02, momentum coefficient 0.9, coefficient for adaptive matrix β as 0.999; for the Local-AMSGrad algorithm, we use learning rate 0.0005, momentum coefficient 0.9, adaptive matrix coefficient 0.999; for the MIME-MVR algorithm, we use learning rate 1, w 10000, c as 400; for the STEM algorithm, we use learning rate 1, w 10000 and c 400; for our FedDA-MVR, we use learning rate 0.02, w as 10000, c as 1000000, β as 0.999 and τ as 0.01. For other variants of FedDA: for FedDA-2-1, we use learning rate 0.001, α as 0.9, β as 0.999, τ as 0.01; for FedDA-1-2, we use learning rate 1, w as 5000, c as 100, β as 0.999, τ as 0.01; for FedDA-2-2, we use the learning rate 0.01, α 0.9, β as 0.999, τ as 0.01.

The experimental results for different variants of FedDA is summarized in Figure 8.6 and 8.7. As shown by plots, all variants of FedDA get good performance. FedDA-MVR

(FedDA-1-1) gets the best performance in most metrics, we observe that its test loss show some extent of overfitting in the late training stage.

8.8.2 More discussion of Experimental Results

In this subsection, we make more detailed comparison between our FedDA and other baselines (The experiments are over homogeneous CIFAR10 dataset). In Figure 8.8, we compare FedCM with FedDA-2-1 and FedDA-2-2 for different values of local steps I. Since FedDA-2-1 and FedDA-2-2 do not use variance reduction acceleration, the superior performance shows the effectiveness of using adaptive gradients in our framework. Next, In Figure 8.9, we compare Local-AMSGrad vs FedDA-2-1 for different values of I, FedDA-2-1 outperforms Local-AMSGrad for all I and with a greater margin for larger I. Note both Local-AMSGrad and FedDA-2-1 use Adam-style adaptive gradients ((8.6) and (8.7)) and have same communication cost per epoch. In Figure 8.10, we compare FedAdam and Local-Adapt with FedDA-2-1. All methods use Adam-style adaptive gradients. FedAdam only performs adaptive gradients over the server, Local-Adapt performs both local and global adaptive gradients, but the state of the local adaptive gradient is refreshed per epoch. We have two observations: First, the Local-Adapt method has very marginal improvement over FedAdam, which shows the restarted strategy used by Local-Adapt is less effective than our method; Second, both FedAdam and Local-Adapt benefit little from increasing the I value (compared to our FedDA-2-1). For FedAdam, this shows the limitation of only applying adaptive gradients at the server level. Finally, in Figure 8.11, we change I for all four variants of our FedDA. As shown by the figure, our framework can benefit from more

local steps.

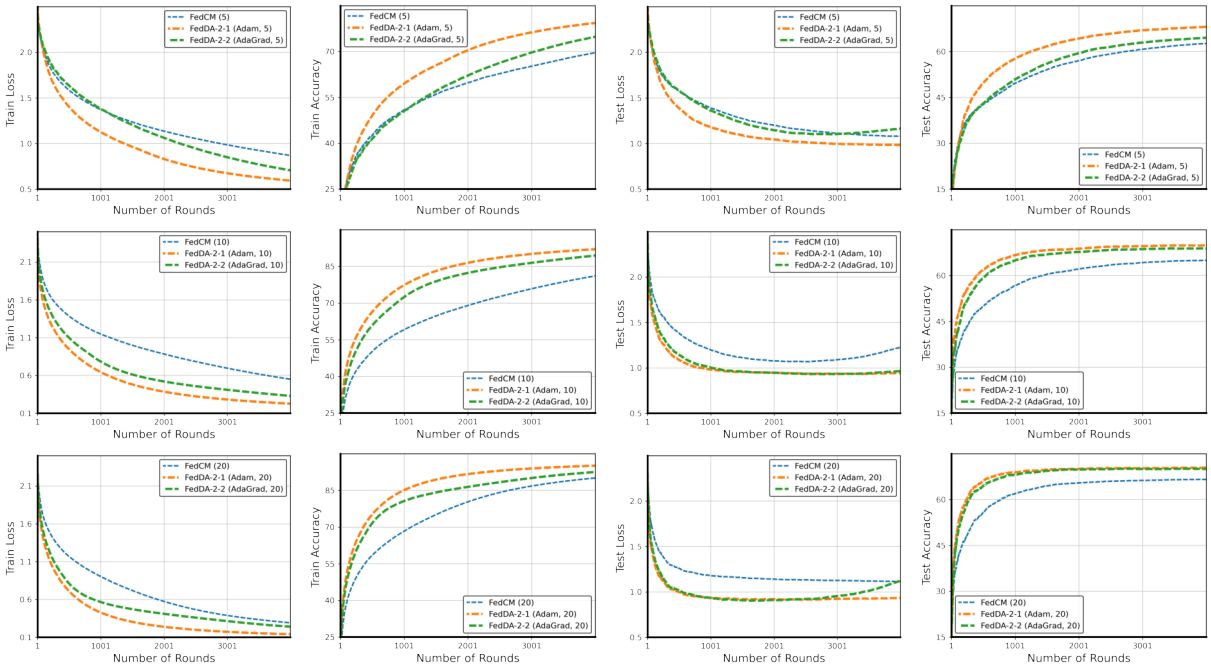


Figure 8.8: Comparison between FedCM vs FedDA-2-1 and FedDA-2-2. From top to bottom, we show $I = 5, 10, 20$ respectively. The number inside the parentheses is the value of I.

8.8.3 Image Classification Task with Heterogeneous CIFAR10

For the heterogeneous case, we create heterogeneity in the training set as follows: Suppose we have 10 clients, for i_{th} client, we distribute ρ-percent samples of i_{th} class, and $(1-\rho)/9$-percent samples of other classes, where $0 < \rho \leq 1$. Note for ρ close to 1, the i_{th} client will be dominated by images of i_{th} class, thus the data distribution among clients will be very different. In our experiments, we choose $\rho = 0.8$. This means the i_{th} client has 4000 images of i_{th} class and 111 images of other classes. This creates a high level of heterogeneity.

For hyper-parameters, we perform grid search and choose the best setting for each

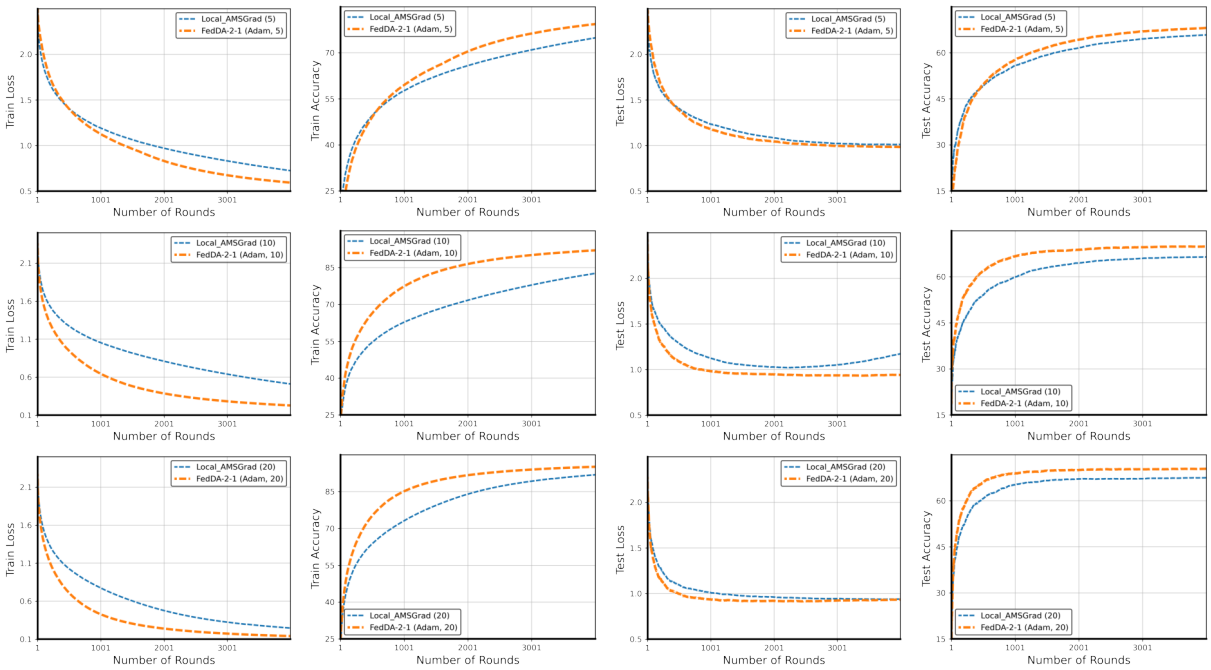


Figure 8.9: Comparison between Local-AMSGrad vs FedDA-2-1. From top to bottom, we show $I = 5, 10, 20$ respectively. The number inside the parentheses is the value of I.

method. For the SGD method, we use learning rate 0.01; for the FedCM algorithm, we use learning rate 0.01, momentum coefficient α as 0.9; for the FedAdam algorithm, we use local learning rate 0.001, global learning rate 0.002, momentum coefficient 0.9, coefficient for adaptive matrix β as 0.999; for the Local-Adapt algorithm, we use local learning rate 0.001, global learning rate 0.002, momentum coefficient 0.9, coefficient for adaptive matrix β as 0.999; for the Local-AMSGrad algorithm, we use learning rate 0.001, momentum coefficient 0.9, adaptive matrix coefficient 0.999; for the MIME-MVR algorithm, we use learning rate 0.1, w 100, c as 2000; for the STEM algorithm, we use learning rate 0.1, w 100 and c 2000; for our FedDA-MVR/FedDA-1-1, we use learning rate 0.02, w as 10000, c as 1000000, β as 0.999 and τ as 0.01. For other variants of FedDA: for FedDA-2-1, we use learning rate 0.001, α as 0.9, β as 0.999, τ as 0.01; for FedDA-1-2, we use learning rate 1,

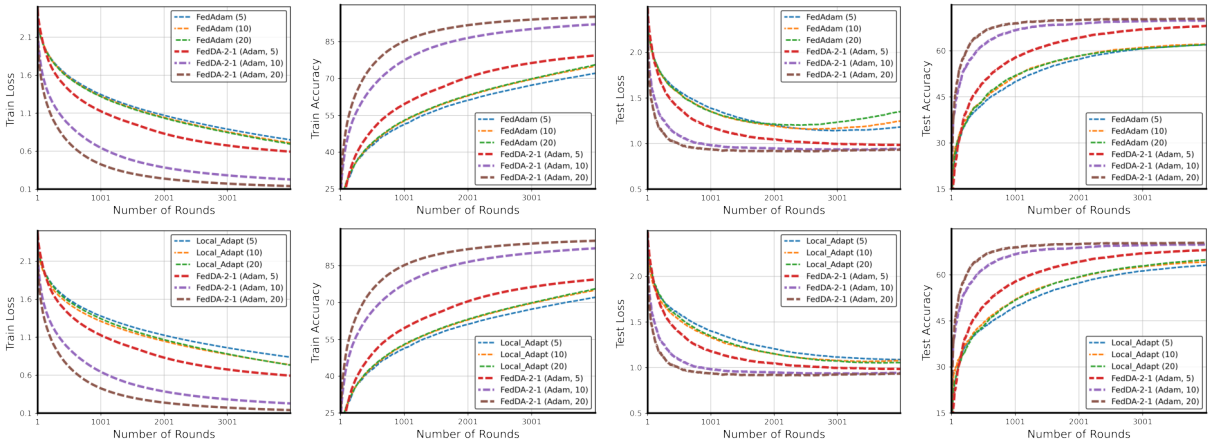


Figure 8.10: Comparison between FedAdam and Local-Adapt vs FedDA-2-1. The number inside the parentheses is the value of I.

w as 5000, c as 100, β as 0.999, τ as 0.01; for FedDA-2-2, we use learning rate 0.01, α 0.9, β as 0.999, τ as 0.01.

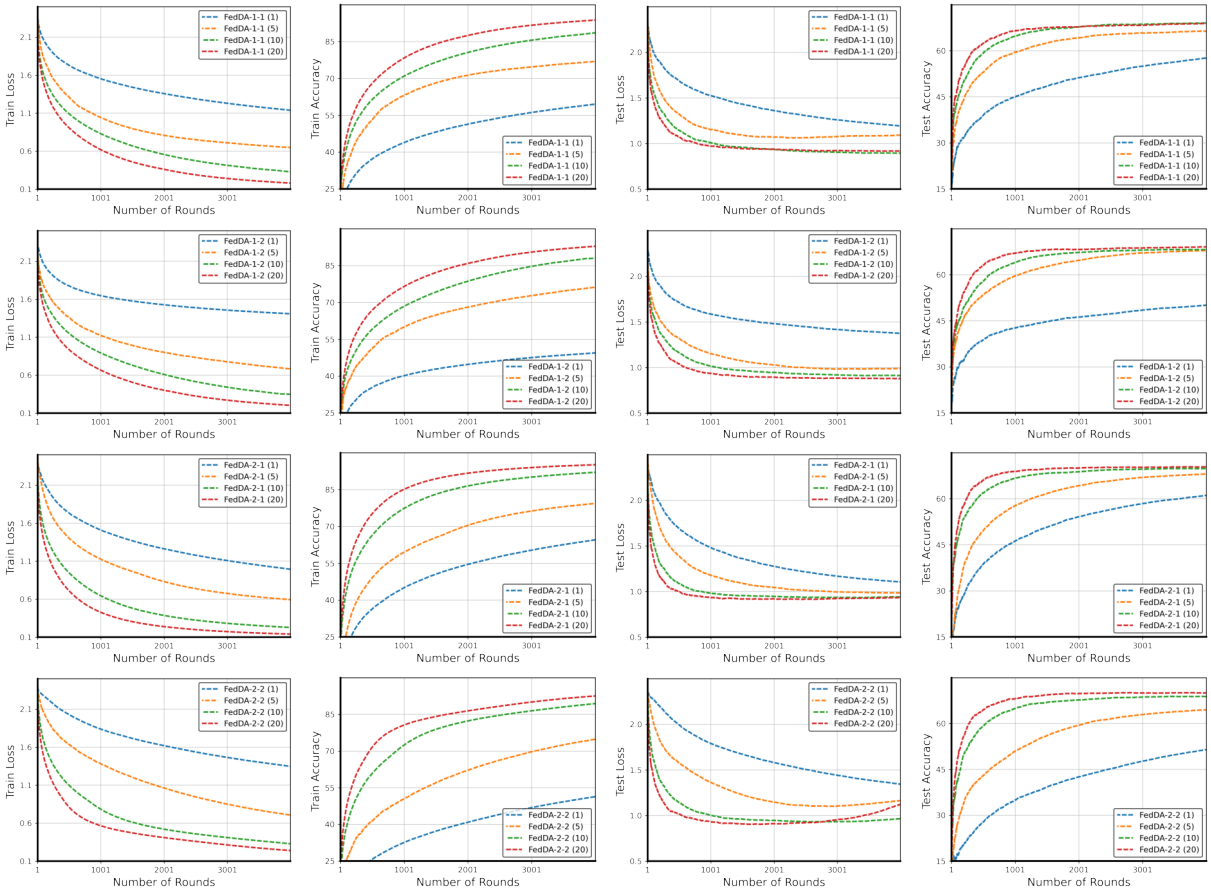


Figure 8.11: Ablation study of local steps I. From top row to the bottom row, we show results for FedDA-1-1, FedDA-1-2, FedDA-2-1 and FedDA-2-2. The number inside the parentheses is the value of I.

Bibliography

[1] Yucheng Lu, Wentao Guo, and Christopher M De Sa. Grab: Finding provably better data permutations than random reshuffling. *Advances in Neural Information Processing Systems*, 35:8969–8981, 2022.

[2] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021.

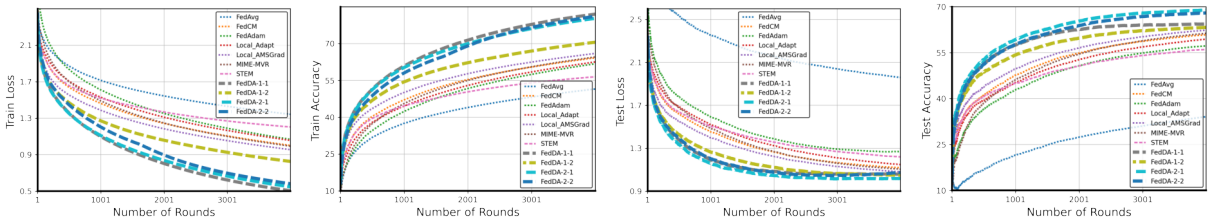


Figure 8.12: Results for heterogeneous CIFAR10 dataset. From left to right, we show Train Loss, Train Accuracy, Test Loss, Test Accuracy *w.r.t* the number of rounds (E in Algorithm 16), respectively. I is chosen as 5.

[3] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[4] Jonathan Lorraine and David Duvenaud. Stochastic hyperparameter optimization through hypernetworks. *arXiv preprint arXiv:1802.09419*, 2018.

[5] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. *arXiv preprint arXiv:1806.04910*, 2018.

[6] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

[7] Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR, 2020.

[8] Michael C Ferris and Olvi L Mangasarian. Finite perturbation of convex programs. *Applied Mathematics and Optimization*, 23(1):263–273, 1991.

[9] Renjie Liao, Yuwen Xiong, Ethan Fetaya, Lisa Zhang, KiJung Yoon, Xaq Pitkow, Raquel Urtasun, and Richard Zemel. Reviving and improving recurrent back-propagation. In *International Conference on Machine Learning*, pages 3082–3091. PMLR, 2018.

[10] Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*, 2021.

[11] Tianyi Chen, Yuejiao Sun, and Wotao Yin. A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671*, 2021.

[12] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Provably faster algorithms for bilevel optimization and applications to meta-learning. *arXiv preprint arXiv:2010.07962*, 2020.

[13] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *arXiv preprint arXiv:2102.07367*, 2021.

[14] Feihu Huang and Heng Huang. Biadam: Fast adaptive bilevel optimization methods. *arXiv preprint arXiv:2106.11396*, 2021.

[15] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

[16] Ralph A Willoughby. Solutions of ill-posed problems (an tikhonov and vy arsenin). *SIAM Review*, 21(2):266, 1979.

[17] Mikhail Solodov. An explicit descent method for bilevel convex optimization. *Journal of Convex Analysis*, 14(2):227, 2007.

[18] Isao Yamada, Masahiro Yukawa, and Masao Yamagishi. Minimizing the moreau envelope of nonsmooth convex functions over the fixed point set of certain quasi-nonexpansive mappings. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 345–390. Springer, 2011.

[19] Shoham Sabach and Shimrit Shtern. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660, 2017.

[20] Jan Larsen, Lars Kai Hansen, Claus Svarer, and M Ohlsson. Design and regularization of neural networks: the optimal use of a validation set. In *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*, pages 62–71. IEEE, 1996.

[21] Dingding Chen and Martin T Hagan. Optimal use of regularization and cross-validation in neural network modeling. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 2, pages 1275–1280. IEEE, 1999.

[22] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

[23] Chuong B Do, Chuan-Sheng Foo, and Andrew Y Ng. Efficient multiple hyperparameter learning for log-linear models. In *NIPS*, volume 2007, pages 377–384. Citeseer, 2007.

[24] Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pages 318–326. PMLR, 2012.

[25] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122, 2015.

[26] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1165–1173. JMLR. org, 2017.

[27] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. *arXiv preprint arXiv:1602.02355*, 2016.

[28] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. *arXiv preprint arXiv:1810.10667*, 2018.

[29] Kaiyi Ji and Yingbin Liang. Lower bounds and accelerated algorithms for bilevel optimization. *arXiv preprint arXiv:2102.03926*, 2021.

[30] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*, 2019.

[31] Feihu Huang, Junyi Li, and Heng Huang. Super-adam: Faster and universal framework of adaptive gradients. *arXiv preprint arXiv:2106.08208*, 2021.

[32] Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *arXiv preprint arXiv:2106.04692*, 2021.

[33] Feihu Huang and Heng Huang. Enhanced bilevel optimization via bregman distance. *arXiv preprint arXiv:2107.12301*, 2021.

[34] Akshay Mehra and Jihun Hamm. Penalty method for inversion-free deep bilevel optimization. *arXiv preprint arXiv:1911.03432*, 2019.

[35] Junyi Li, Bin Gu, and Heng Huang. Improved bilevel model: Fast and optimal algorithm with theoretical guarantee. *arXiv preprint arXiv:2009.00690*, 2020.

[36] Daouda Sow, Kaiyi Ji, Ziwei Guan, and Yingbin Liang. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.

[37] Takayuki Okuno, Akiko Takeda, and Akihiro Kawana. Hyperparameter learning via bilevel nonsmooth optimization. *arXiv preprint arXiv:1806.01520*, 2018.

[38] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.

[39] Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. Es-maml: Simple hessian-free meta learning. *arXiv preprint arXiv:1910.01215*, 2019.

[40] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3516–3525, 2020.

[41] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[42] Catherine Wong, Neil Houlsby, Yifeng Lu, and Andrea Gesmundo. Transfer learning with neural automl. *arXiv preprint arXiv:1803.02780*, 2018.

[43] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. *arXiv preprint arXiv:1907.05737*, 2019.

[44] Yuesong Tian, Li Shen, Guinan Su, Zhifeng Li, and Wei Liu. Alphagan: Fully differentiable architecture search for generative adversarial networks. *arXiv preprint arXiv:2006.09134*, 2020.

[45] Chen Gao, Yunpeng Chen, Si Liu, Zhenxiong Tan, and Shuicheng Yan. Adversarialnas: Adversarial neural architecture search for gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5680–5689, 2020.

[46] Sebastian Tschiatschek, Ahana Ghosh, Luis Haug, Rati Devidze, and Adish Singla. Learner-aware teaching: Inverse reinforcement learning with preferences and constraints. *arXiv preprint arXiv:1906.00429*, 2019.

[47] Runxue Bao, Bin Gu, and Heng Huang. Efficient approximate solution path algorithm for order weight l_1-norm with accuracy guarantee. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 958–963. IEEE, 2019.

[48] Runxue Bao, Bin Gu, and Heng Huang. Fast oscar and owl regression via safe screening rules. In *International Conference on Machine Learning*, pages 653–663. PMLR, 2020.

[49] Clarice Poon and Gabriel Peyré. Smooth bilevel programming for sparse regularization. *Advances in Neural Information Processing Systems*, 34, 2021.

[50] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *arXiv preprint arXiv:2101.11517*, 2021.

[51] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[52] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[53] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[54] Chhavi Yadav and Léon Bottou. Cold case: The lost mnist digits. In *Advances in Neural Information Processing Systems*, pages 13443–13452, 2019.

[55] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[56] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9489–9502. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/shamsian21a.html`.

[57] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[58] Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *arXiv preprint arXiv:2201.13409*, 2022.

[59] Yifan Hu, Jie Wang, Yao Xie, Andreas Krause, and Daniel Kuhn. Contextual stochastic bilevel optimization, 2023.

[60] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436. Springer, 2012.

[61] Benjamin Recht and Christopher Re. Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 11.1–11.24, Edinburgh, Scotland, 25–27 Jun 2012. PMLR. URL `https://proceedings.mlr.press/v23/recht12.html`.

[62] Itay Safran and Ohad Shamir. How good is sgd with random shuffling? In *Conference on Learning Theory*, pages 3250–3284. PMLR, 2020.

[63] Yucheng Lu, Si Yi Meng, and Christopher De Sa. A general analysis of example-selection for stochastic gradient descent. In *International Conference on Learning Representations (ICLR)*, volume 10, 2022.

[64] Mao Ye, Bo Liu, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach, 2022.

[65] Junyi Li, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7426–7434, 2022.

[66] Feihu Huang, Junyi Li, Shangqian Gao, and Heng Huang. Enhanced bilevel optimization via bregman distance. *Advances in Neural Information Processing Systems*, 35:28928–28939, 2022.

[67] Yifan Yang, Peiyao Xiao, and Kaiyi Ji. Achieving $O(\epsilon^{-1.5})$ complexity in hessian/jacobian-free stochastic bilevel optimization, 2023.

[68] Peiran Yu, Junyi Li, and Heng Huang. Dropout enhanced bilevel training. In *The Twelfth International Conference on Learning Representations*, 2023.

[69] Risheng Liu, Zhu Liu, Wei Yao, Shangzhi Zeng, and Jin Zhang. Moreau envelope for nonconvex bi-level optimization: A single-loop and hessian-free solution strategy, 2024.

[70] Yifan Yang, Peiyao Xiao, and Kaiyi Ji. Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] Junyi Li, Feihu Huang, and Heng Huang. Communication-efficient federated bilevel optimization with global and local lower level problems. *Advances in Neural Information Processing Systems*, 36, 2024.

[72] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

[73] Mert Gürbüzbalaban, Asu Ozdaglar, and Pablo A Parrilo. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, 186:49–84, 2021.

[74] Lam M Nguyen, Quoc Tran-Dinh, Dzung T Phan, Phuong Ha Nguyen, and Marten Van Dijk. A unified convergence analysis for shuffling-type gradient methods. *The Journal of Machine Learning Research*, 22(1):9397–9440, 2021.

[75] Jeff Haochen and Suvrit Sra. Random shuffling beats SGD after finite epochs. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2624–2633. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/haochen19a.html`.

[76] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33:17309–17320, 2020.

[77] Dami Choi, Alexandre Passos, Christopher J Shallue, and George E Dahl. Faster neural network training with data echoing. *arXiv preprint arXiv:1907.05550*, 2019.

[78] Alexander Buchholz, Florian Wenzel, and Stephan Mandt. Quasi-monte carlo variational inference. In *International Conference on Machine Learning*, pages 668–677. PMLR, 2018.

[79] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128, 2009.

[80] Mengdi Wang, Ethan X. Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1–2):419–449, May 2016. ISSN 1436-4646. doi: 10.1007/s10107-016-1017-3. URL http://dx.doi.org/10.1007/s10107-016-1017-3.

[81] Aniket Das, Bernhard Schölkopf, and Michael Muehlebach. Sampling without replacement leads to faster rates in finite-sum minimax optimization, 2022.

[82] Hanseul Cho and Chulhee Yun. Sgda with shuffling: faster convergence for nonconvex-p {\L} minimax optimization. *arXiv preprint arXiv:2210.05995*, 2022.

[83] Amirkeivan Mohtashami, Sebastian Stich, and Martin Jaggi. Characterizing & finding good data orderings for fast convergence of sequential gradient methods, 2022.

[84] Lie He and Shiva Prasad Kasiviswanathan. Debiasing conditional stochastic optimization. *arXiv preprint arXiv:2304.10613*, 2023.

[85] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[86] Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. *arXiv preprint arXiv:2111.14580*, 2021.

[87] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[88] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2017.

[89] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[90] Mathieu Dagréou, Thomas Moreau, Samuel Vaiter, and Pierre Ablin. A lower bound and a near-optimal algorithm for bilevel empirical risk minimization. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 82–90. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/dagreou24a.html.

[91] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[92] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[93] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6): 1205–1221, 2019.

[94] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

[95] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[96] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[97] Ahmed Khaled Ragab Bayoumi, Konstantin Mishchenko, and Peter Richtarik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529, 2020.

[98] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 2021.

[99] Junyi Li, Jian Pei, and Heng Huang. Communication-efficient robust federated learning with noisy labels. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 914–924, 2022.

[100] Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. Fednest: Federated bilevel, minimax, and compositional optimization. *arXiv preprint arXiv:2205.02215*, 2022.

[101] Peiyao Xiao and Kaiyi Ji. Communication-efficient federated hypergradient computation via aggregated iterative differentiation. *arXiv preprint arXiv:2302.04969*, 2023.

[102] Minhui Huang, Dewei Zhang, and Kaiyi Ji. Achieving linear speedup in non-iid federated bilevel learning. *arXiv preprint arXiv:2302.05412*, 2023.

[103] Yifan Yang, Peiyao Xiao, and Kaiyi Ji. Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning. *arXiv preprint arXiv:2305.19442*, 2023.

[104] Hongchang Gao. On the convergence of momentum-based algorithms for federated stochastic bilevel optimization problems. *arXiv preprint arXiv:2204.13299*, 2022.

[105] Feihu Huang. On momentum-based gradient methods for bilevel optimization with nonconvex lower-level. *arXiv preprint arXiv:2303.03944*, 2023.

[106] Zhishuai Guo, Quanqi Hu, Lijun Zhang, and Tianbao Yang. Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021.

[107] Xuxing Chen, Minhui Huang, and Shiqian Ma. Decentralized bilevel optimization. *arXiv preprint arXiv:2206.05670*, 2022.

[108] Shuoguang Yang, Xuezhou Zhang, and Mengdi Wang. Decentralized gossip-based stochastic bilevel optimization over communication networks. *arXiv preprint arXiv:2206.10870*, 2022.

[109] Songtao Lu, Siliang Zeng, Xiaodong Cui, Mark Squillante, Lior Horesh, Brian Kingsbury, Jia Liu, and Mingyi Hong. A stochastic linearized augmented lagrangian method for decentralized bilevel optimization. *Advances in Neural Information Processing Systems*, 35:30638–30650, 2022.

[110] Hongchang Gao, Bin Gu, and My T Thai. On the convergence of distributed stochastic bilevel optimization algorithms over a network. In *International Conference on Artificial Intelligence and Statistics*, pages 9238–9281. PMLR, 2023.

[111] Yang Jiao, Kai Yang, Tiancheng Wu, Dongjin Song, and Chengtao Jian. Asynchronous distributed bilevel optimization. *arXiv preprint arXiv:2212.10048*, 2022.

[112] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[113] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3, 2018.

[114] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[115] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. *arXiv preprint arXiv:1902.00146*, 2019.

[116] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[117] Pengwei Xing, Songtao Lu, Lingfei Wu, and Han Yu. Big-fed: Bilevel optimization enhanced graph-aided federated learning. *IEEE Transactions on Big Data*, 2022.

[118] Blake Woodworth. The minimax complexity of distributed optimization. *arXiv preprint arXiv:2109.00534*, 2021.

[119] Prashant Khanduri, Pranay Sharma, Haibo Yang, Mingyi Hong, Jia Liu, Ketan Rajawat, and Pramod Varshney. Stem: A stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning. *Advances in Neural Information Processing Systems*, 34:6050–6061, 2021.

[120] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020.

[121] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[122] Karthik Nandakumar, Nalini Ratha, Sharath Pankanti, and Shai Halevi. Towards deep neural network training on encrypted data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[123] Sameer Wagh, Divya Gupta, and Nishanth Chandran. Securenn: 3-party secure computation for neural network training. *Proc. Priv. Enhancing Technol.*, 2019(3): 26–49, 2019.

[124] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

[125] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

[126] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *arXiv preprint arXiv:1809.07599*, 2018.

[127] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR, 2019.

[128] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed sgd with sketching. *arXiv preprint arXiv:1903.04488*, 2019.

[129] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.

[130] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pages 8253–8265. PMLR, 2020.

[131] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[132] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 22(1):191–205, 2021.

[133] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *arXiv preprint arXiv:2204.12703*, 2022.

[134] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

[135] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[136] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *arXiv preprint arXiv:2212.01548*, 2022.

[137] Junyi Li, Feihu Huang, and Heng Huang. Communication-efficient federated bilevel optimization with local and global lower level problems. *arXiv preprint arXiv:2302.06701*, 2023.

[138] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[139] Sara Babakniya, Souvik Kundu, Saurav Prakash, Yue Niu, and Salman Avestimehr. Federated sparse training: Lottery aware model compression for resource constrained edge. *arXiv preprint arXiv:2208.13092*, 2022.

[140] Shangqian Gao, Feihu Huang, Weidong Cai, and Heng Huang. Network pruning via performance maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9270–9280, 2021.

[141] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[142] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[143] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

[144] Xinchi Qiu, Javier Fernandez-Marques, Pedro PB Gusmao, Yan Gao, Titouan Parcollet, and Nicholas Donald Lane. Zerofl: Efficient on-device training for federated learning with local sparsity. *arXiv preprint arXiv:2208.02507*, 2022.

[145] Sameer Bibikar, Haris Vikalo, Zhangyang Wang, and Xiaohan Chen. Federated dynamic sparse training: Computing less, communicating less, yet learning better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6080–6088, 2022.

[146] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[147] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

[148] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[149] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

[150] Yiqiang Chen, Xiaodong Yang, Xin Qin, Han Yu, Biao Chen, and Zhiqi Shen. Focus: Dealing with label quality disparity in federated learning. *arXiv preprint arXiv:2001.11359*, 2020.

[151] Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. Robust federated learning with noisy labels. *arXiv preprint arXiv:2012.01700*, 2020.

[152] Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K Leung. Overcoming noisy and irrelevant data in federated learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5020–5027. IEEE, 2021.

[153] Lloyd S Shapley. *Notes on the N-person Game–I: Characteristic-point Solutions of the Four-person Game*. Rand Corporation, 1951.

[154] Junyi Li, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. *arXiv preprint arXiv:2112.04660*, 2021.

[155] Feihu Huang, Junyi Li, and Heng Huang. Compositional federated learning: Applications in distributionally robust averaging and meta learning. *arXiv preprint arXiv:2106.11264*, 2021.

[156] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International conference on machine learning*, pages 125–134. PMLR, 2015.

[157] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

[158] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5552–5560, 2018.

[159] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

[160] Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2021.

[161] Runxue Bao, Xidong Wu, Wenhan Xian, and Heng Huang. Distributed dynamic safe screening algorithms for sparse regularization. *arXiv preprint arXiv:2204.10981*, 2022.

[162] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and system sciences*, 58(1):137–147, 1999.

[163] Junyi Li, Feihu Huang, and Heng Huang. Local stochastic bilevel optimization with momentum-based variance reduction. *arXiv preprint arXiv:2205.01608*, 2022.

[164] David P Woodruff. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014.

[165] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[166] Jan Philipp Albrecht. How the gdpr will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

[167] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[168] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[169] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[170] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[171] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[172] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[173] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[174] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[175] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[176] Rulin Shao, Hui Liu, and Dianbo Liu. Privacy preserving stochastic channel-based federated learning with neural network pruning. *arXiv preprint arXiv:1910.02115*, 2019.

[177] Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: empower edge intelligence with personalized and communication-efficient federated learning. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 68–79. IEEE, 2021.

[178] Muhammad Tahir Munir, Muhammad Mustansar Saeed, Mahad Ali, Zafar Ayyub Qazi, and Ihsan Ayyub Qazi. Fedprune: Towards inclusive federated learning. *arXiv preprint arXiv:2110.14205*, 2021.

[179] Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassiulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[180] Hong Huang, Lan Zhang, Chaoyue Sun, Ruogu Fang, Xiaoyong Yuan, and Dapeng Wu. Fedtiny: Pruned federated learning towards specialized tiny models. *arXiv preprint arXiv:2212.01977*, 2022.

[181] Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 304–320, 2018.

[182] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 2130–2141, 2019.

[183] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[184] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*, 2020.

[185] Junyi Li and Heng Huang. Faster secure data mining via distributed homomorphic encryption. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2706–2714, 2020.

[186] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[187] Shangqian Gao, Feihu Huang, Jian Pei, and Heng Huang. Discrete model compression with resource constraint for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1899–1908, 2020.

[188] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML, pages 448–456. JMLR.org, 2015. URL `http://dl.acm.org/citation.cfm?id=3045118.3045167`.

[189] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[190] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

[191] Jaedeok Kim, Chiyoun Park, Hyun-Joo Jung, and Yoonsuck Choe. Plug-in, trainable gate for streamlining arbitrary neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[192] Shangqian Gao, Feihu Huang, Yanfu Zhang, and Heng Huang. Disentangled differentiable network pruning. In *European Conference on Computer Vision*, pages 328–345. Springer, 2022.

[193] Shangqian Gao, Burak Uzkent, Yilin Shen, Heng Huang, and Hongxia Jin. Learning to jointly share and prune weights for grounding based vision and language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[194] Alireza Ganjdanesh*, Shangqian Gao*, and Heng Huang. Interpretations steered network pruning via amortized inferred saliency maps. In *European Conference on Computer Vision*, pages 278–296. Springer, 2022.

[195] Shangqian Gao, Zeyu Zhang, Yanfu Zhang, Feihu Huang, and Heng Huang. Structural alignment for network pruning through partial regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17402–17412, 2023.

[196] Alireza Ganjdanesh, Shangqian Gao, and Heng Huang. Effconv: efficient learning of kernel sizes for convolution layers of cnns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7604–7612, 2023.

[197] Alireza Ganjdanesh*, Shangqian Gao*, Hirad Alipanah, and Heng Huang. Compressing image-to-image translation gans using local density structures on their learned manifold. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[198] Dongfang Xu, Zeyu Zhang, and Steven Bethard. A generate-and-rank framework with semantic type regularization for biomedical concept normalization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8452–8464, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.748. URL `https://aclanthology.org/2020.acl-main.748`.

[199] Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. Joint models for answer verification in question answering systems. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3252–3262, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.252. URL `https://aclanthology.org/2021.acl-long.252`.

[200] Zeyu Zhang, Thuy Vu, Sunil Gandhi, Ankit Chadha, and Alessandro Moschitti. Wdrass: A web-scale dataset for document retrieval and answer sentence selection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4707–4711, New York, NY, USA, 2022. Association for

Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557678. URL https://doi.org/10.1145/3511808.3557678.

[201] Hannah Smith, Zeyu Zhang, John Culnan, and Peter Jansen. ScienceExamCER: A high-density fine-grained science-domain corpus for common entity recognition. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4529–4546, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.558.

[202] Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. Double retrieval and ranking for accurate question answering. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1751–1762, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.130. URL https://aclanthology.org/2023.findings-eacl.130.

[203] Zeyu Zhang and Steven Bethard. Improving toponym resolution with better candidate generation, transformer-based reranking, and two-stage resolution. In Alexis Palmer and Jose Camacho-collados, editors, *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 48–60, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.starsem-1.6. URL https://aclanthology.org/2023.starsem-1.6.

[204] Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. In situ answer sentence selection at web-scale. *arXiv preprint arXiv:2201.05984*, 2022.

[205] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.496. URL https://aclanthology.org/2020.emnlp-main.496.

[206] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[207] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[208] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020.

[209] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3, 2015.

[210] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[211] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[212] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[213] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[214] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

[215] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[216] Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pages 10334–10343. PMLR, 2020.

[217] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[218] Yubei Chen Chun-Hsiao Yeh. IN100pytorch: Pytorch implementation: Training resnets on imagenet-100. `https://github.com/danielchyeh/ImageNet-100-Pytorch`, 2022.

[219] M Moshawrab, Adda M, Bouzouane A, Ibrahim H, and Raad A. Reviewing federated machine learning and its use in diseases prediction. *Sensors*, 23(4):2112, 2023.

[220] Madhura Joshi, Ankit Pal, and Malaikannan Sankarasubbu. Federated learning for healthcare domain - pipeline, applications and challenges. *ACM Trans. Comput. Healthcare*, 3(4), 2022.

[221] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Trans. Intell. Syst. Technol.*, 13(4), 2022.

[222] J. Xu, B.S. Glicksberg, C. Su, et al. Federated learning for healthcare informatics. *J Healthc Inform Res*, 5:1–19, 2021.

[223] A Rahman, Hossain MS, Muhammad G, Kundu D, Debnath T, Rahman M, Khan MSI, Tiwari P, and Band SS. Federated learning-based ai approaches in smart healthcare: concepts, taxonomies, challenges and open issues. *Cluster Comput.*, pages 1–41, 2022.

[224] S Pati, Baid U, Edwards B, et al. Federated learning enables big data for rare cancer boundary detection. *Nat Commun.*, 13(1):7346, 2022.

[225] M.J. Sheller, B. Edwards, G.A. Reina, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Sci Rep*, 10: 12598, 2020.

[226] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[227] Xiangyi Chen, Xiaoyun Li, and Ping Li. Toward communication efficient adaptive gradient method. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, pages 119–128, 2020.

[228] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

[229] Honglin Yuan, Manzil Zaheer, and Sashank Reddi. Federated composite optimization. In *International Conference on Machine Learning*, pages 12253–12266. PMLR, 2021.

[230] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021.

[231] Rudrajit Das, Anish Acharya, Abolfazl Hashemi, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Faster non-convex federated learning via global and local momentum. *arXiv preprint arXiv:2012.04061*, 2020.

[232] Xidong Wu, Feihu Huang, Zhengmian Hu, and Heng Huang. Faster adaptive federated learning. *arXiv preprint arXiv:2212.00974*, 2022.

[233] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[234] Olvi L Mangasarian and Mikhail V Solodov. Backpropagation convergence via deterministic nonmonotone perturbed minimization. *Advances in Neural Information Processing Systems*, 6, 1993.

[235] Aymeric Dieuleveut and Kumar Kshitij Patel. Communication trade-offs for local-sgd with large step size. *Advances in Neural Information Processing Systems*, 32, 2019.

[236] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[237] Margalit R Glasgow, Honglin Yuan, and Tengyu Ma. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 9050–9090. PMLR, 2022.

[238] Qianqian Tong, Guannan Liang, and Jinbo Bi. Effective federated adaptive gradient methods with non-iid decentralized data. *arXiv preprint arXiv:2009.06557*, 2020.

[239] Yujia Wang, Lu Lin, and Jinghui Chen. Communication-efficient adaptive federated learning. In *International Conference on Machine Learning*, pages 22802–22838. PMLR, 2022.

[240] Hanlin Tang, Shaoduo Gan, Samyam Rajbhandari, Xiangru Lian, Ji Liu, Yuxiong He, and Ce Zhang. Apmsqueeze: A communication efficient adam-preconditioned momentum sgd algorithm. *arXiv preprint arXiv:2008.11343*, 2020.

[241] Hanlin Tang, Shaoduo Gan, Ammar Ahmad Awan, Samyam Rajbhandari, Conglong Li, Xiangru Lian, Ji Liu, Ce Zhang, and Yuxiong He. 1-bit adam: Communication efficient large-scale training with adam's convergence speed. In *International Conference on Machine Learning*, pages 10118–10129. PMLR, 2021.

[242] Yucheng Lu, Conglong Li, Minjia Zhang, Christopher De Sa, and Yuxiong He. Maximizing communication efficiency for large-scale training via 0/1 adam. *arXiv preprint arXiv:2202.06009*, 2022.

[243] Congliang Chen, Li Shen, Haozhi Huang, Wei Liu, and Zhi-Quan Luo. Efficient-adam: Communication-efficient distributed adam with complexity analysis. 2020.

[244] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[245] Mahesh Chandra Mukkamala and Matthias Hein. Variants of rmsprop and adagrad with logarithmic regret bounds. In *International Conference on Machine Learning*, pages 2545–2553. PMLR, 2017.

[246] Zaiyi Chen, Yi Xu, Enhong Chen, and Tianbao Yang. Sadagrad: Strongly adaptive stochastic gradient methods. In *International Conference on Machine Learning*, pages 913–921. PMLR, 2018.

[247] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, pages 9793–9803, 2018.

[248] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[249] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[250] Dongruo Zhou, Jinghui Chen, Yuan Cao, Yiqi Tang, Ziyan Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

[251] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[252] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[253] Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

[254] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

[255] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33, 2020.

[256] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686. PMLR, 2019.

[257] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

[258] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

[259] Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

[260] Jianyu Wang, Zheng Xu, Zachary Garrett, Zachary Charles, Luyang Liu, and Gauri Joshi. Local adaptivity in federated learning: Convergence and consistency. *arXiv preprint arXiv:2106.02305*, 2021.