ABSTRACT

Title of Dissertation: LEARNING STRUCTURED ALIGNMENT:
FROM VISUAL UNDERSTANDING TO ROBOT CONTROL

Shuaiyi Huang
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Abhinav Shrivastava
Department of Computer Science

The rapid growth of visual data and the increasing demand for intelligent robotic systems have created a pressing need for methods that can establish meaningful correspondences and relationships across diverse visual modalities and robotic tasks. This dissertation addresses the fundamental challenge of learning structured alignment, which involves establishing correspondences between different representations, temporal sequences, and task domains to enable more effective visual understanding and robot control.

In the first part of this thesis, we advance visual understanding through three key contributions that demonstrate the power of structured alignment in perception tasks. We begin by tackling semantic correspondence, where we propose a teacher-student learning paradigm that enriches supervision from sparse keypoint annotations, enabling dense correspondence learning through spatial priors and loss-driven dynamic label selection. We then address video instance segmentation through two complementary approaches: UVIS, which leverages foundation models (DINO and CLIP) for unsupervised segmentation without dense annotations, and

PointVIS, which achieves competitive performance using only point-level supervision through class-agnostic proposal generation and spatio-temporal matching. Finally, we develop Trokens for few-shot action recognition, introducing semantic-aware point correspondence sampling and relational motion alignment that captures both intra-trajectory dynamics through Histogram of Oriented Displacements and inter-trajectory spatial relationships, effectively aligning appearance features with motion patterns through trajectory-based token alignment.

While the first part focuses on establishing correspondences within visual data, real-world applications require bridging the gap between visual understanding and robot control. In the second part of this thesis, we present two frameworks that demonstrate how structured alignment can be extended to robotic applications. We introduce ARDuP, a novel method for video-based policy learning that aligns generated visual plans with language instructions for effective control. This innovative framework integrates active region (i.e. potential interaction areas) conditioning with latent diffusion models for video planning and employs latent representations for direct action decoding during inverse dynamic modeling. By utilizing motion cues in videos for automatic active region discovery, our method eliminates the need for manual annotations of active regions. Finally, we present TREND, which addresses robust preference-based reinforcement learning through a tri-teaching framework that filters noisy preference labels while incorporating few-shot expert demonstrations, demonstrating effective alignment between human preferences and robot behaviors even under high noise conditions.

LEARNING STRUCTURED ALIGNMENT: FROM VISUAL UNDERSTANDING TO ROBOT CONTROL

by

Shuaiyi Huang

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 Abhinav Shrivastava, Chair/Advisor
Professor Shuvra S. Bhattacharyya, Dean's Representative
Professor Furong Huang
Professor Pratap Tokekar
Professor Ruohan Gao

© Copyright by
Shuaiyi Huang
2025

Acknowledgments

First and foremost, I want to thank my advisor, Abhinav Shrivastava, for his invaluable guidance and support throughout my PhD. He gave me the freedom to pursue topics I was genuinely passionate about, which allowed me to explore correspondence, segmentation, and robotics. His trust and mentorship helped me adapt quickly and grow across diverse areas. Abhinav is deeply knowledgeable and has a sharp sense for impactful research directions. Our discussions always left me inspired by his clarity and insight. Working with him has shaped the way I think about research and what it means to do meaningful work. More than a research mentor, Abhinav has also been a role model in life. I admire how he balances a successful academic career with a strong commitment to his family. His "family first" mindset and ability to lead with both compassion and strength left a lasting impression on me. He built a lab culture that is supportive, collaborative, and healthy. In moments of stress, his optimism and small gestures, like sending memes during deadlines, helped keep us grounded. Over time, I found my own mindset shifting, shaped by his example. Abhinav's wisdom, character, and mentorship have had a profound impact on me. I am deeply grateful to have had him as my advisor and will carry the lessons I've learned from him throughout my life.

I also appreciate the valuable and insightful suggestions from my committee members, including Professor Furong Huang, Professor Pratap Tokekar, Professor Shuvra S. Bhattacharyya, Professor Ruohan Gao, and Professor Tianyi Zhou. Their valuable insights and suggestions have

immensely contributed to the depth and quality of my dissertation.

I would like to sincerely thank my mentors from the industry and academia for their invaluable guidance during my internships. I am especially grateful to De-An Huang, Zhiding Yu, Shiyi Lan, Jim Fan, Yuke Zhu, Subhashree Radhakrishnan, Jose M. Alvarez, and Anima Anandkumar at NVIDIA, and my advisor Xuming He during my master. Their insights have not only contributed to several successful paper submissions but have also deepened my understanding of potential career paths.

I am grateful to my colleagues and friends in the Perception Intelligence Lab, including Bo He, Yixuan Ren, Hanyu Wang, Luyu Yang, Hao Chen, Pulkit Kumar, Mara Levy, Saksham Suri, Kamal Gupta, Saketh Rambhatla, Anubhav Gupta, Daniel Ekpo, Namitha Padmanabhan, Nirat Saini, Shishira R. Maiya, Khoi Pham, Sharath Girish, Max Enrich, Vinoj Jayasundara, Soumik Mukhopadhyay, Alex Hanson, Matthew Gwilliam, Matthew Walmer, Lillian Huang, Vatsal Agarwal, Chuong Huynh, Archana Swaminathan, Gaurav Shrivastava, Moustafa Meshry, Prateksha Udhayanan, Pravin Nagar, Sonaal Kant, and Bhuvanesh Murali, for their support, collaboration, and for enriching my PhD journey in countless ways. I also feel fortunate to have collaborated with colleagues outside our lab, including Zhenyu Jiang, Tianrui Guan, Xiyang Wu, and Songyang Zhang, whose thoughtful advice and feedback played an important role in shaping our projects.

I am deeply thankful for the friends who accompanied me throughout my PhD journey, especially Peihong Yu, Chen Chen, Yifan Zhu, Zhou He, Suzie, Jiahao Zhan, Yiran Xu, Mingyang Xie, Yue Feng, Chujun Song, Simiao Lin, Jing Qiu, Huanming Li, Jingqi Gao, and Jiaqi Liu. Their companionship, particularly during the challenging times of the pandemic, meant a great deal to me. Their presence and support provided comfort, strength, and encouragement when I

needed it most.

Most of all, I am incredibly grateful to my parents and family. Their constant support has been the foundation of my journey, helping me navigate the challenges of a Ph.D. with strength and peace of mind. I would also like to sincerely thank my boyfriend, Ruijie Zheng, for always being there for me. More than a partner, he has been a teammate and an inspiration. He was always there through the difficult times, offering comfort and reassurance when I needed it most. His companionship has brought joy and meaning to my life.

I acknowledge that it is not possible to individually recognize everyone who has supported me throughout this journey, and I extend my sincere apologies to anyone I may have unintentionally omitted. Your contributions have been deeply meaningful, and I remain truly grateful for each and every one of you.

Table of Contents

Acknowledgements ii

Table of Contents v

List of Tables viii

List of Figures x

Chapter 1: Introduction 1

Chapter 2: Learning Semantic Alignment with Sparse Annotations 5
- 2.1 Introduction . . . 5
- 2.2 Related Work . . . 7
  - 2.2.1 Semantic Correspondence . . . 7
  - 2.2.2 Teacher-Student Learning . . . 8
- 2.3 Model Architecture . . . 9
  - 2.3.1 Efficient Spatial Context Encoder . . . 10
  - 2.3.2 Correlation Map Computation . . . 11
  - 2.3.3 Flow formation and High-resolution loss . . . 11
- 2.4 Learning with Sparse Annotations . . . 12
  - 2.4.1 Sparse Label Densification via Teacher-Student Learning . . . 13
  - 2.4.2 High Quality Pseudo-label Generation . . . 14
  - 2.4.3 Variants of Teacher-Student Learning . . . 16
- 2.5 Experiments . . . 18
  - 2.5.1 Implementation Details . . . 19
  - 2.5.2 Comparison with State-of-the-art Methods . . . 21
  - 2.5.3 Ablation Study . . . 24
- 2.6 Conclusion . . . 26

Chapter 3: Learning Instance Alignment in Videos 27
- 3.1 What is Point Supervision Worth in Video Instance Segmentation? . . . 27
  - 3.1.1 Introduction . . . 27
  - 3.1.2 Related Work . . . 31
  - 3.1.3 Method . . . 33
  - 3.1.4 Experiments . . . 40
  - 3.1.5 Conclusion . . . 47
- 3.2 UVIS: Unsupervised Video Instance Segmentation . . . 49

3.2.1 Introduction . . . 49
3.2.2 Related Work . . . 52
3.2.3 Method . . . 54
3.2.4 Experiments . . . 60
3.2.5 Conclusion . . . 67

Chapter 4: Learning Dense Point Alignment for Video Action Recognition 68
4.1 Introduction . . . 68
4.2 Related Work . . . 71
4.3 Few-shot Action Recognition Setup . . . 73
4.4 Our Approach . . . 74
4.4.1 Overview . . . 74
4.4.2 Semantic-aware Point Trajectories . . . 74
4.4.3 Relational Motion Modeling . . . 76
4.4.4 Motion-aware Space-Time Transformer . . . 78
4.4.5 Few-shot Loss . . . 79
4.5 Experiments . . . 80
4.5.1 Datasets . . . 80
4.5.2 Implementation details . . . 80
4.5.3 Quantitative Results . . . 81
4.5.4 Ablation Analysis . . . 82
4.5.5 Qualitative Analysis . . . 86
4.6 Discussion . . . 87

Chapter 5: Learning Visual Plan Alignment for Robot Control 89
5.1 Introduction . . . 89
5.2 Related Work . . . 92
5.2.1 Video Diffusion Models for Decision Making . . . 92
5.2.2 Active Region for Vision and Robotics . . . 93
5.3 Problem Formulation . . . 94
5.3.1 Preliminaries . . . 94
5.3.2 Latent Unified Predictive Decision Process conditioned on Active Region (LUPDP-AR) . . . 95
5.4 Method: Active Region Video Diffusion for Universal Policies (ARDuP) . . . 96
5.4.1 Active Region Generator . . . 97
5.4.2 Universal Video Planner conditioned on Active Region . . . 100
5.4.3 Task Specific Action Decoding from Latent Sequence . . . 101
5.5 Experiments . . . 102
5.5.1 Implementation Details . . . 103
5.5.2 Multi-Environment Transfer on CLIPort . . . 105
5.5.3 Real-world Transfer on BridgeData v2 . . . 106
5.5.4 Ablation Study . . . 108
5.5.5 Limitations and Future work . . . 109
5.6 Conclusion . . . 109

Chapter 6: Learning Robust Robot Behavior Alignment through Preferences 111
6.1 Introduction . . . 111
6.2 Related Work . . . 114
6.3 Preliminaries . . . 116
6.4 Method: Tri-teaching for Robust Preference-based Reinforcement Learning with Demonstrations (TREND) . . . 117
6.4.1 Tri-teaching for Preference Data Selection . . . 118
6.4.2 Incorporating Few-shot Expert Demonstrations . . . 119
6.5 Experiments . . . 122
6.5.1 Experimental Setup . . . 122
6.5.2 Main Results . . . 124
6.5.3 Ablation Study . . . 125
6.6 Conclusion . . . 128

Chapter 7: Future Work and Conclusion 129
7.1 Future Work . . . 129
7.2 Conclusion . . . 130

Bibliography 132

List of Tables

2.1 **Comparison with SOTA methods on SPair-71k [1]**. Per-class and overall PCK ($\alpha_{\text{bbox}} = 0.1$) results are shown in the table. Numbers in bold indicate the best performance and underlined ones are the second best. All models in this table use ResNet101 as the backbone. *Sup.* denotes the type of supervision. * means the backbone is finetuned. † means ground truth bbox used. 19

2.2 **Comparison with SOTA methods on PF-PASCAL** [2]. Numbers in bold indicate the best performance and underlined ones are the second best. † means ground truth bbox used. 21

2.3 **Comparison with SOTA methods on PF-WILLOW [3].** Numbers in bold indicate the best performance and underlined ones are the second best. † means ground truth bbox used. 23

2.4 **Effects of each component on SPair-71K [1] test split.** HRLoss refers to high-resolution loss, FT refers to finetuning the backbone, Teacher-Student here refers to the variant with mutual online teacher. 24

2.5 **Comparing single offline teacher and mutual online teacher** setting on SPair-71K [1] test split. 25

2.6 **Effects of kernel size** for label mask dilation on SPair-71K [1] test split. 25

3.1 **Full mask ($\mathcal{M}$) vs. our point supervision ($\mathcal{P}$) on validation set of YouTube-VIS 2019 [4], YouTube-VIS 2021 [4], and OVIS [5].** All results below are based on Swin-L backbone. Our PointVIS results are with self-training. 39

3.2 **Effects of each component on YouTube-VIS 2019 [4] val-dev.** 42

3.3 **Analysis of point selection bias on YouTube-VIS 2019 [4] val-dev.** 42

3.4 **Effects of additional points on YouTube-VIS 2019 [4] val-dev.** "DPPointMatcher" means Dense Pseudo via Point Matcher. "CINeg" means enforcing additional negative point loss. 43

3.5 **Comparison with baselines on YouTube-VIS 2019 [4] validation set.** 44

3.6 **PointVIS (P1N1) with subsampled video frames on YouTube-VIS 2019 validation set (w/o self-training).** 47

3.7 **Mask ($\mathcal{M}$) / Box ($\mathcal{B}$) / category ($\mathcal{C}$) vs. our unsupervised setting on validation set of YouTube-VIS 2019 [4], YouTube-VIS 2021 [4], and OVIS [5].** * indicates training in videos without COCO pretrained model weights as initialization using authors' official code. "I-Sup." and "V-sup." indicate the supervision used in the image dataset and the video dataset, respectively. All results below are based on R50 backbone. Our UVIS achieves decent results in all three datasets without any videos annotations or dense supervision from images. 61

3.8 **Ablation of different components of our pipeline on YTVIS-2019 [4] val set.** . 64

3.9 **Ablation of Prototype Memory Filtering (PMF) on YouTube-VIS 2019 [4] val**. Prototype Memory Filtering improves AP by 0.9. 65
3.10 **Ablation of tracking memory on YouTube-VIS 2019 [4] val**. Our proposed tracking memory can be generalized to different datasets and different supervision settings. 65

4.1 Comparison of few-shot action accuracy (1-5 shots) on SSV2 Full and Kinetics datasets versus contemporary methods. Best results are bolded, second-best underlined, and "-" indicates unavailable data. 81
4.2 Comparison of few-shot action accuracy (1,3 and 5 shots) on SSV2 Small, UCF-101 and HMDB-51 datasets versus contemporary methods. Best results are bolded, second-best underlined, and "-" indicates unavailable data. 83
4.3 Comparative N-way 1-shot classification accuracy (N=5-10) on Kinetics and SSV2 Full datasets versus contemporary methods. Best and second-best results are bolded and underlined, respectively. 84
4.4 Impact of each component on Trokens, demonstrating the relative contribution of individual elements with the final row representing our final setting. 84
4.5 Comparison of few-shot action accuracy (1,3 and 5 shots) on the FineGym dataset. 85
4.6 Comparative analysis of intra-motion module variants independent of inter-motion module. 85
4.7 Trainable parameter analysis. * represents our implementation of [6] with increased parameters for fair comparison. 85

5.1 Success rate in completing tasks across multiple environments in CLIPort [7]. When trained across various multi-task environments, our approach shows strong generalization capabilities to unseen environments, significantly outperforming the UniPi baseline [8]. UniPi* is reproduced based on the reference code provided by the authors. 104
5.2 Effect of Active Region Quality on Performance. The higher-quality active region during training and testing phases, the more improvement in task success rates, highlighting the significance of the active region. 106

List of Figures

1.1 **Overview of structured alignment problems addressed in this thesis.** (a) *Pixel-level semantic alignment between image pairs*, where our SCorrSAN framework learns dense correspondences from sparse keypoint annotations by leveraging spatial context. (b) *Instance-level temporal alignment* in videos, where PointVIS and UVIS enable high-quality video instance segmentation with minimal or no supervision by maintaining spatio-temporal consistency. (c) *Trajectory-level alignment for few-shot action recognition*, where Trokens models semantic and relational correspondences across object trajectories using structured tokens. (d) *Visual plan alignment for robot control*, where ARDuP generates task-aligned trajectories by focusing on active interaction regions to bridge language instructions with physical actions. (e) *Preference-level alignment under noisy supervision*, where TREND employs a tri-teaching framework to learn robust reward models and align robot behavior with human intent. 2

2.1 **Motivation.** Image pairs from SPair-71k dataset [1] training split show sparse annotations for semantic correspondence. 6

2.2 **Model overview.** (a) Illustration of our network. Our network comprises three main modules, including an efficient spatial context encoder, a correlation module, and a flow estimator. (b) Comparison between our proposed efficient spatial context and the full spatial context. Please refer Sec. 2.3 for more details. *Best viewed in color.* 9

2.3 **Sparse Label Densification with Teacher-Student Learning.** (a) Our Teacher-Student Learning Pipeline. Solid lines stand for Single Offline Teacher, with additional dashed lines standing for Mutual Online Teacher. (b) Illustration of Label Mask Dilation. Please refer to Sec. 3.2.3 for more details. *Best viewed in color.* 13

2.4 **Qualitative results of our method on SPair-71k [1].** From left to right are source image, target image, result from CATs [9], and result from ours (MT), resp. 22

3.1 **Point-supervised video instance segmentation in this work (YoutubeVIS-2021)** Top: point-level annotations in the training set (pseudo masks generated from our method overlaid); Bottom: mask predictions in the validation set. 28

3.2 **Method Overview.** Our method consists of class-agnostic spatial-temporal proposal generation, a spatio-temporal point-based matcher to match object queires with point annotations for high-quality pseudo-label generation, and self-training to mitigate the domain gap between images and videos. See text for details. . . . 30

3.3 Visualization of point annotations and pseudo masks obtained by our method on Youtube-VIS 2019 [4] (row1), Youtube-VIS 2021 [4] (row2), and OVIS [5] (row3) training set. 35
3.4 **Failure cases on OVIS [5].** We observe temporal inconsistency (tiger in the top left) or missing instances (person in white) in our pseudo masks. 48
3.5 **Setting Overview.** Previous approaches have tried to use COCO dense annotations in addition to VIS dataset full supervision (a), box supervision (b) and no supervision (c). Additionally, previous works have also used flow information along with frame-level category labels (d). Our approach UVIS works in the unsupervised setting and does not require any dense labels or per frame labels and instead utilizes foundation models. 51
3.6 **We present our approach UVIS.** On the left we show our pseudo-label generation pipeline which involves generating masks and instance labels using CutLER [10] and CLIP [11] followed by Prototype Memory Filtering (PMF). In the center we show our model training which uses and image encoder and a transformer decoder to learn queries to predict per-frame predictions. On the right we show our proposed tracking memory approach which utilizes per frame queries and a memory based update rule to perform matching between frames to track instances and generate temporally consistent predictions. 55
3.7 **Visualizations on YoutubeVIS-2019 [4] with our UVIS**. Each row shows temporal instance mask and class predictions. Our method is able to work for examples containing multiple instances of the same class (rows 1, 3, 4) and also when there are instances from different classes (row 5). UVIS shows promising results when instances of the same class might overlap (row 4). 63
3.8 **Visualizations of failure cases on YoutubeVIS-2019 [4]**. On the left we show CLIP labeling failures where the CLIP model incorrectly classifies to the wrong class. In the center we show prediction inconsistencies where multiple instances are predicted as one. On the right we show temporal inconsistencies in predicted masks. 65

4.1 **Our motivation.** (a) Our semantic-aware points adapts better to object scale and semantic relevance while existing methods w/ grid sampling missed small objects w/ important motion (e.g., knife). (b) We explicitly model relational motions within a trajectory and across trajectories. 69
4.2 **Method Overview.** (A) Given an input video, we extract appearance tokens using DINOv2. (B) We then cluster these tokens and sample semantic-aware points in the initial frame, which are tracked using Co-tracker [12] to obtain point trajectories. (C) We compute intra- and inter-motion features, reorder appearance tokens via token alignment [6], and fuse them with motion features via element-wise addition to form semantic-aware relational trajectory tokens. (D) Finally, we input these tokens into a Decoupled Space-Time Transformer for few-shot action classification. 73
4.3 Class-wise accuracy comparison between MoLo [13], TATs [6] and our method. Left: classes where our approach shows performance gains. Right: classes without improvement. 83

4.4 Few-shot accuracy by variyng the number of points on SSV2 Small and SSV2 Full for 1-shot setting. 86
4.5 Visualization of action trajectory similarities across four classes, where semantic-based sampling enables object-focused trajectories. Each quadrant demonstrates intra-class motion consistency while maintaining inter-class discriminative features. 87

5.1 Given the task text and initial frame, we aim to generate a video as a planner. With active region conditioning (bottom left), our method ARDuP generates frames where the robot arm successfully picks up the white block (bottom), unlike the incorrect targeting of a purple block when w/o active region input (top), showing ARDuP's effectiveness in producing task-aligned video sequences. 90
5.2 **Overview of our Active Region Video Diffusion for Universal Policies (ARDuP).** Starting with a video, we use Co-Tracker to identify moving points at the initial frame, which SAM then uses to generate pseudo masks of active regions. These pseudo masks delineate the pseudo active region, serving as supervision for training our Latent Active Region Diffusion Model. The generated latent active region informs the Latent Video Diffusion Model, resulting in a synthesized video latent sequence. Finally, the Latent Inverse Dynamic Model decodes the generated latent sequence into a corresponding action sequence. 93
5.3 **Qualitative comparison of the generated video plans on CLIPort [7] unseen test tasks.** Our method successfully generates a video that packs the white block, while the counterpart w/o active region generates frames that pick the wrong object (the red block) or fails to correctly generate the arm position for grasping objects. We demonstrate higher visual quality, especially around objects of interest, thanks to the guidance provided by our active regions. 100
5.4 **Qualitative comparison of the generated video plans on BridgeData v2 [14].** Our method successfully picks up the correct object (sushi) or places the object in the appropriate location (in the cup or colander), while the UniPi* baseline selects the wrong object (cucumber) or places it incorrectly (on the desk), showing the advantage of our method. 103
5.5 **Task Loss on CLIPort [7] test set.** Our method (green) shows significantly lower task loss compared to that without the active region (red), underscoring the active region's role in enhancing generation quality. 107
5.6 **Visualization of pseudo active region in BridgeData v2.** We visualize the dense point trajectories and the obtained pseudo masks of the active region in BridgeData v2 [14]. 108

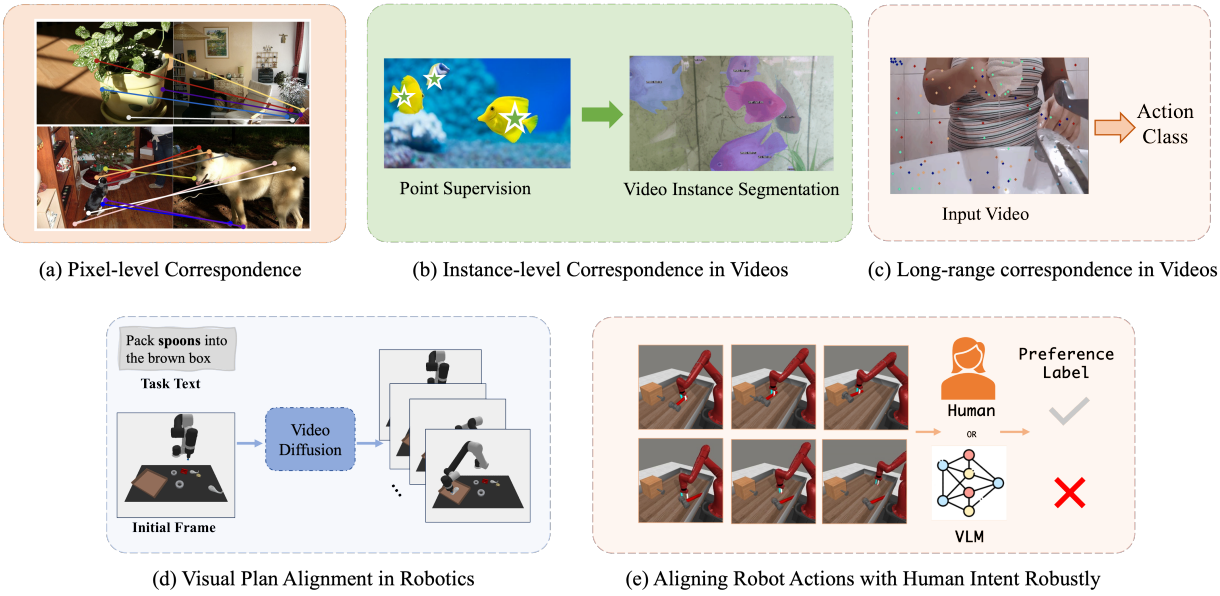
6.1 **Overview of our method TREND.** First, we pretrain the policy network using behavior cloning (BC) with few-shot expert demonstrations for effective exploration (A). In the online training phase, noisy preferences are collected from human annotators or a vision-language model (B1). We then apply our tri-teaching strategy for denoised reward learning, where three collaborative reward models identify clean preference samples for each other (B2). Finally, the learned reward model is used to guide the agent's training (B3), ensuring robust performance despite noisy labels. 112

6.2 **Learning curves for robot manipulation tasks on Meta-world.** Each row represents results for a specific task and each column corresponds to a different error rate ϵ. Results are averaged over five seeds. Shaded Areas represent standard deviation across seeds. . . . 120
6.3 **Contribution of each component on Button-Press ($\epsilon = 40\%$).** Our tri-teaching boosts performance significantly, and adding a single expert demonstration further enhances success, emphasizing the need for both components. 123
6.4 **Results on Drawer-Open using VLM (Gemini-1.5-flash) to generate preference feedback.** Our TREND-VLM-F achives the best result (left) under the high noise rate of VLM labels (right). 125
6.5 **(Left):** Comparison between our method (w. Tri-teach) and a baseline denoising strategy (w. Self-teach) on Hammer ($\epsilon = 40\%$). **(Right):** Comparison of clean label ratio under different noise levels between our method (w. Tri-teach) and the baseline (w/o. Tri-teach) on Hammer. 126
6.6 **Hyperparameter analysis on Hammer** (ϵ=40%). 127

Chapter 1: Introduction

The ability to establish meaningful correspondences and alignments between different forms of information lies at the heart of both computer vision and robotics. Whether matching pixels across images, tracking objects through video frames, or connecting language instructions to physical actions, alignment is fundamental to understanding and interacting with the world. Despite significant advances in deep learning over the past decade, developing computational methods that can reliably perform alignment remains challenging, particularly when limited supervision is available.

This thesis addresses structured alignment challenges across five progressively complex domains: pixel-level semantic alignment between image pairs, instance-level temporal alignment in videos, trajectory-level alignment for action recognition, task-level alignment between language instructions and robot actions, and preference-level alignment under noisy supervision, as shown in Figure 1.1. We develop novel frameworks that effectively learn these alignments with limited supervision by leveraging carefully designed architectures and training strategies. Our work begins with fundamental visual understanding through semantic correspondence (Chapter 2), progresses to dynamic video understanding through instance-level alignment in segmentation tasks (Chapter 3), and further captures temporal correspondence across object trajectories for few-shot action recognition in Trokens (Chapter 4). Building on these perceptual alignment capabilities,



(a) Pixel-level Correspondence (b) Instance-level Correspondence in Videos (c) Long-range correspondence in Videos

(d) Visual Plan Alignment in Robotics (e) Aligning Robot Actions with Human Intent Robustly

Figure 1.1: **Overview of structured alignment problems addressed in this thesis.** (a) *Pixel-level semantic alignment between image pairs*, where our SCorrSAN framework learns dense correspondences from sparse keypoint annotations by leveraging spatial context. (b) *Instance-level temporal alignment* in videos, where PointVIS and UVIS enable high-quality video instance segmentation with minimal or no supervision by maintaining spatio-temporal consistency. (c) *Trajectory-level alignment for few-shot action recognition*, where Trokens models semantic and relational correspondences across object trajectories using structured tokens. (d) *Visual plan alignment for robot control*, where ARDuP generates task-aligned trajectories by focusing on active interaction regions to bridge language instructions with physical actions. (e) *Preference-level alignment under noisy supervision*, where TREND employs a tri-teaching framework to learn robust reward models and align robot behavior with human intent.

we transition to robotic decision-making by introducing a visual planning framework that aligns video predictions with task goals in ARDuP (Chapter 5), and conclude with TREND (Chapter 6), which addresses preference-level alignment under noisy supervision through a tri-teaching strategy. Together, these contributions demonstrate how structured alignment and correspondence can be leveraged across modalities and domains to enhance both perception and control in embodied systems.

At the most fundamental level, alignment begins with understanding pixel-level correspondences between images. This problem, known as semantic correspondence, requires matching

semantically similar regions across images despite variations in appearance, viewpoint, and context. In Chapter 2, we address this challenge through SCorrSAN, a novel teacher-student framework that effectively learns dense semantic alignment from sparse keypoint annotations. Our approach leverages spatial context and cross-instance constraints to densify sparse supervision signals, enabling robust matching across diverse scenarios.

While pixel-level alignment provides foundational understanding, many real-world applications require tracking and understanding object instances over time. This necessitates moving from static image pairs to dynamic video sequences, where we must maintain consistent instance-level alignment across frames. Chapter 3 presents two complementary approaches to this challenge: PointVIS, which achieves high-quality video instance segmentation using minimal point supervision, and UVIS, which eliminates the need for manual annotations entirely through self-supervised learning. These methods demonstrate how structured constraints and temporal consistency can compensate for limited supervision in video understanding.

Beyond object tracking, video action understanding often requires modeling higher-order relational and temporal structure. In Chapter 4, we introduce Trokens, a framework that constructs semantic-aware relational trajectory tokens for few-shot action recognition. By integrating dense point-based motion cues with object-level relational modeling and temporal attention, Trokens captures fine-grained dynamics and significantly improves recognition performance in data-scarce settings. This contribution highlights how structured alignment at the trajectory level enables sample-efficient generalization in temporal tasks.

The ultimate goal of computer vision is often to enable intelligent interaction with the physical world through robotics. This requires aligning visual understanding with concrete actions - a significantly more complex form of alignment that must bridge perception and control.

In Chapter 5, we introduce ARDuP, a framework for video-based policy learning that generates task-aligned visual plans for robot manipulation. By focusing on active regions where interaction occurs, ARDuP effectively aligns language instructions with appropriate action sequences, demonstrating how structured alignment principles can enhance robot control.

Finally, Chapter 6 tackles the challenge of aligning robot behavior with human preferences, especially under noisy or ambiguous feedback. We present TREND, a tri-teaching framework for robust preference-based reinforcement learning. TREND employs three collaborative reward models to identify and share clean preference samples while incorporating a small number of expert demonstrations to provide a grounding supervisory signal. This design ensures effective learning even in high-noise scenarios, and enables scalable preference alignment without exhaustive labeling.

Throughout this thesis, we demonstrate how carefully designed training frameworks can enable effective alignment learning with limited supervision across different domains and scales, from pixels to objects to actions. Our work takes important steps toward more practical and deployable vision and robotics systems that can establish meaningful correspondences across diverse forms of information. The methods we develop advance the theoretical understanding of alignment learning and demonstrate strong empirical performance on real-world tasks.

Chapter 2: Learning Semantic Alignment with Sparse Annotations

2.1 Introduction

Paper List: [1, 9, 15–22]

Estimating pixel-wise correspondence between images is a fundamental task in computer vision applications. Correspondences like stereo disparities [23] and optical flow [24] are widely used for applications such as surface reconstruction and video analysis [25, 26]. Recently, such instance-level dense correspondence has been generalized to semantic correspondence, which, given a pair of images, aligns the object instance from the first image to the one of the same category in the second image [15, 16, 18, 20, 27–30]. It has attracted growing attention due to its practical use in segmentation, style-transfer, and image editing [31–36]. However, background clutter, intra-class variations, viewpoint changes, and particularly the severe lack of annotations make it an extremely challenging task.

Due to the high cost of dense annotation, the semantic correspondence task only provides sparse keypoint annotations in the supervised setting [20, 21, 37, 38] as shown in Fig. 2.1. In this paper, we are motivated by how to better utilize the limited supervision. Specifically, we explore the techniques to generate pseudo-labels. However, due to the inevitably noisy effect of pseudo-labels, filtering out noisy pseudo-labels remains a challenging problem. Our key observation is that sparse keypoint annotations and their neighborhood encode rich semantic information.

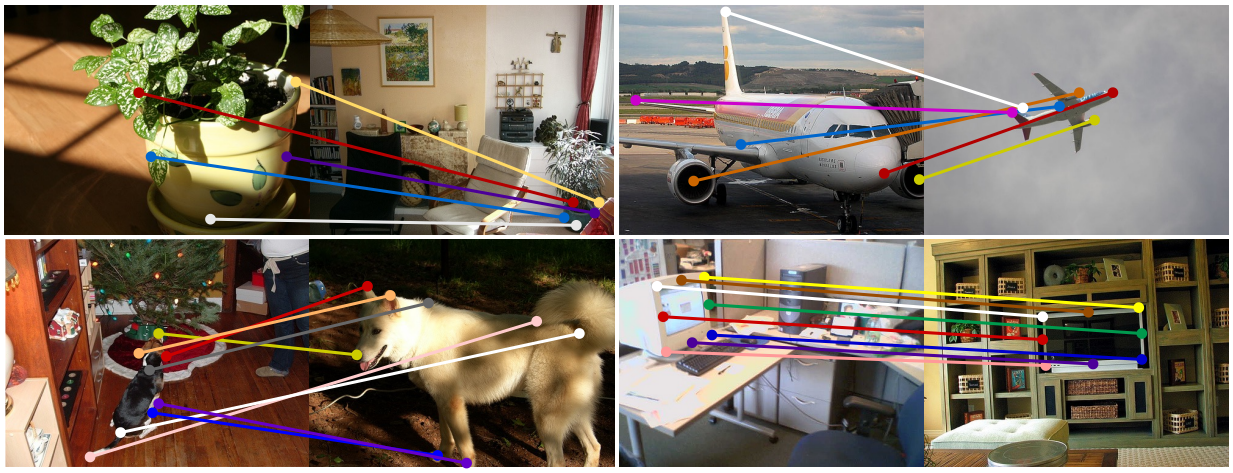


Figure 2.1: **Motivation.** Image pairs from SPair-71k dataset [1] training split show sparse annotations for semantic correspondence.

By utilizing this spatial prior, one can seek reliable pseudo-labels that are more likely in the foreground region of interest.

To this end, we propose a novel teacher-student framework to cope with label sparsity. The teacher model is trained with sparse keypoint annotations to generate dense pseudo-labels. To improve pseudo-labels quality, we propose (a) using the sparse annotations as spatial prior to suppress the noisy pseudo-labels, and (b) loss-driven dynamic label selection. To train the models, we propose two variants of our strategy: (1) a single offline teacher with an online student, and (2) two online teachers that learn from each other. Both variants lead to substantial performance improvements over the state-of-the-art.

We instantiate our novel learning strategy based on our proposed simple, yet effective network architecture for semantic correspondence. The proposed network comprises three modules: (a) a feature extractor equipped with our efficient spatial context encoder, (b) a parameter-free correlation map module, and (c) a flow estimator with our designed high-resolution loss.

The contributions are summarized as follows:

- We propose a simple, yet effective model for semantic correspondence without any transformer or 4D-conv for correlation refinement. The key ingredients are an efficient spatial context encoder and a high-resolution loss.

- We introduce a novel teacher-student learning paradigm to enrich the supervision guidance when only sparse annotations are available. Two key techniques are a novel spatial-prior based label filtering and a loss-driven dynamic label selection strategy for high-quality pseudo-label generation.

- Our novel learning strategy is simple to implement, and achieves state-of-the-art results with good generalization performance on three semantic correspondence benchmarks, demonstrating the effectiveness of our method.

2.2 Related Work

2.2.1 Semantic Correspondence

Conventional approaches for semantic correspondence mostly employ hand-crafted features together with geometric models [39–41]. These methods establish correspondences across images via energy minimization. SIFT Flow [39] pioneers the idea of finding correspondences across similar scenes with SIFT descriptors. Ham *et al.* [2] utilize object proposals as the matching primitives and establish correspondence via HOG descriptors. Those methods often have difficulty dealing with background clutter, intra-class variations, and large viewpoint changes due to the lack of semantics in features.

Recently, deep CNN-based methods have been widely used in semantic correspondences

due to their powerful representations. Early methods formulate semantic correspondence as a geometric alignment problem, with a major focus on developing robust geometric models [15, 42, 43]. Rocco *et al.* [15, 16] propose a two-stage CNN architecture for regressing image-level transformation parameters, while other efforts regress local translation fields [35, 43, 44]. More recent works tend to formulate semantic correspondence as a pixel-wise matching problem and cast it as a classification problem. Among these works, there are techniques focusing on developing powerful feature representations [18, 20, 28], correlation map filtering with 4D/6D-conv or transformers [9, 36, 37, 45, 46], effective correspondence readout [19], and different levels of supervision [21, 47, 48]. However, none of these aforementioned methods have explicitly approached the task of dense semantic correspondence from the perspective of sparse annotations.

2.2.2 Teacher-Student Learning

Teacher-student framework has been widely used in semi-supervised learning (SSL) [49–53], where the predictions of the teacher model on unlabeled samples serve as pseudo-labels to guide the student model. Teacher-student framework also plays an important role in knowledge distillation [54–58], where knowledge from a larger teacher model can be transferred into a smaller student model without loss of validity. Recently, Xin *et al.* [21] extend teacher-student to semantic correspondence, where they distill knowledge learned from a probabilistic teacher model on synthetic data to a static student model with unlabeled real image pairs. In contrast, we directly learn from real image pairs labeled with sparse keypoints, and focus on addressing the label sparsity challenge via Siamese teacher-student network design [59]. Note that we tailor teacher-student learning specifically for the dense prediction task of semantic correspondence,

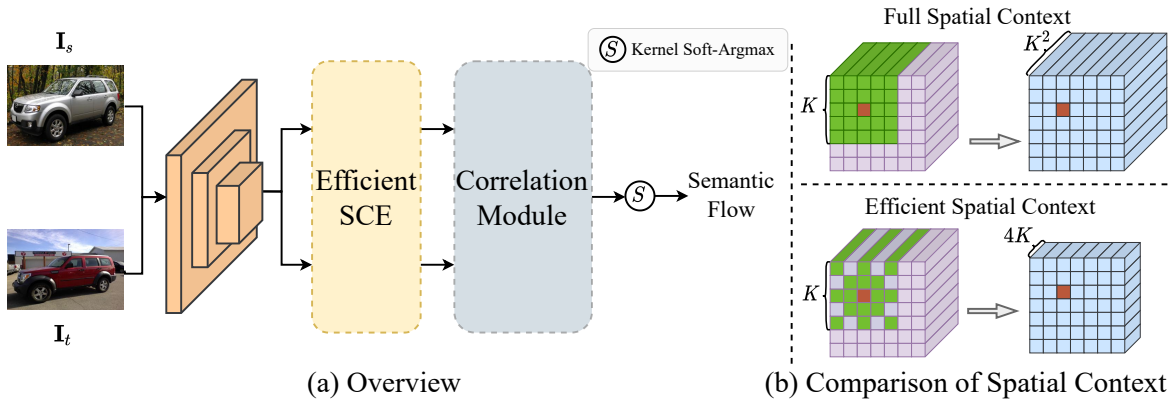


Figure 2.2: **Model overview.** (a) Illustration of our network. Our network comprises three main modules, including an efficient spatial context encoder, a correlation module, and a flow estimator. (b) Comparison between our proposed efficient spatial context and the full spatial context. Please refer Sec. 2.3 for more details. *Best viewed in color.*

where we conduct pixel-level semi-supervised learning within an image and generate pseudo-labels for unlabeled pixels, while most existing work focus on image-level semi-supervised learning.

2.3 Model Architecture

Semantic correspondence establishes dense correspondences between a source image $\mathbf{I}_a$ and a target image $\mathbf{I}_b$. We adopt a typical CNN-based method which computes a correlation map between the convolution features of two images, based on which a dense flow field is predicted as the final output. We additionally encode spatial context efficiently to compute high-quality correlation map and develop a novel teacher-student learning strategy to cope with label sparsity.

This section introduces our simple and powerful semantic correspondence framework. As depicted in Fig. 2.2, our framework comprises of three main modules: (1) a sparse spatial context feature extractor that encodes context information efficiently (Sec. 2.3.1), (2) a correlation operator to compute the correlation map between two convolution features (Sec. 2.3.2), and (3) a

flow estimation operator with high-resolution loss (Sec. 2.3.3).

2.3.1 Efficient Spatial Context Encoder

Taking the conv features of the image pairs as the input, the first component of our network is an efficient spatial context encoder that incorporates spatial context into conv features. Recent methods adopt the self-similarity based descriptor to encode spatial context [18, 37]. However, the time complexity of the self-similarity grows quadratically with respect to the kernel size of the self-similarity descriptor due to dense sampling patterns they used [18,37]. Inspired by the recent success of sparse attention on reducing the computational cost of non-local operation [60–63], we propose a spatial context encoder based on sparse sampling patterns, which efficiently encodes context information and reduces the time complexity from quadratic to linear.

As shown in Fig. 2.2(b), at location (i, j), its spatial context descriptor $\mathbf{s}_{(i,j)} \in \mathbb{R}^{4K}$ is a self-similarity vector, where K is the self-similarity operator kernel size. It is computed between its own feature vector $\mathbf{z}_{(i,j)} \in \mathbb{R}^{d_z}$ and its $4K$ neighboring feature vectors, where the neighbors are in its criss-cross and diagonal directions in an fixed ordered. In contrast to dense sampling patterns [18, 37], our sparse sampling patterns reduce the time and space complexity for computing spatial descriptors from $O(K^2)$ to $O(K)$.

To combine spatial context and conv features, we employ a simple fusion step to generate the final context-aware semantic feature map $\mathbf{G}$ following [18]. Concretely, we concatenate $\mathbf{z}_{(i,j)}$ and $\mathbf{s}_{(i,j)}$ and feed the result into a linear transformation with parameter $\mathbf{W} \in \mathbb{R}^{(d_z+4K)\times d_g}$ followed by a ReLU operation, resulting in a context-aware semantic feature vector $\mathbf{g}_{(i,j)} \in \mathbb{R}^{d_g}$. We add subscripts to represent the context-aware semantic feature map $\mathbf{G}_a \in \mathbb{R}^{d_g \times h_a \times w_a}$ and

$\mathbf{G}_b \in \mathbb{R}^{d_g \times h_b \times w_b}$ for the image $\mathbf{I}_a$ and $\mathbf{I}_b$, resp., where h_b, w_b (resp. h_a, w_a) is the spatial size of $\mathbf{G}_b$ (resp. $\mathbf{G}_a$).

2.3.2 Correlation Map Computation

We compute a 4D correlation map from the context-aware semantic feature maps $\mathbf{G}_a$, $\mathbf{G}_b$ and filter it with the mutual nearest neighbor module [17]. We denote the resulting 4D correlation map as $\mathbb{C} \in \mathcal{R}^{h_a \times w_a \times h_b \times w_b}$.

We propose to learn a high-resolution correlation map for high-quality dense matching in contrast to learning correspondence in stride16 [16, 18]. We upsample the correlation map $\mathbb{C}$ (4 times) instead of upsampling the feature maps for memory efficiency. We denote the resulting upsampled correlation map as

$$\mathbf{C} = \mathcal{U}(\mathbb{C}), \quad \mathbf{C} \in \mathbb{R}^{H_a \times W_a \times H_b \times W_b}, \tag{2.1}$$

where H_a, W_a, H_b, and W_b are the upsampled spatial sizes, $\mathcal{U}$ is the upsample operation. Note that we achieve high performance with single layer feature, while DHPF [20] requires multi-layer features with higher complexity.

2.3.3 Flow formation and High-resolution loss

To obtain differentiable flow, we adopt the kernel soft-argmax operator [19] to transform the upsampled correlation map $\mathbf{C}$ into dense semantic flow $\hat{f}$ as below:

$$\hat{f} = \mathcal{S}(\mathbf{C}), \quad \hat{f} \in \mathbb{R}^{2 \times H_b \times W_b} \tag{2.2}$$

where $\mathcal{S}$ is the kernel soft-argmax operator without any learnable parameters, $\hat{f}$ is the predicted semantic flow in the direction of the target to source.

During training, as we only have sparse keypoint labels, the ground-truth flow $f^{\text{gt}} \in \mathbb{R}^{2\times H_b\times W_b}$ have valid values only at labeled positions. We use a sparse binary label mask $\mathbf{M} \in \mathbb{R}^{H_b\times W_b}$ to indicate valid positions with ground-truth labels as below:

$$\mathbf{M}(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p} \text{ is labeled,} \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where $\mathbf{p}$ is the position index in f^{gt}.

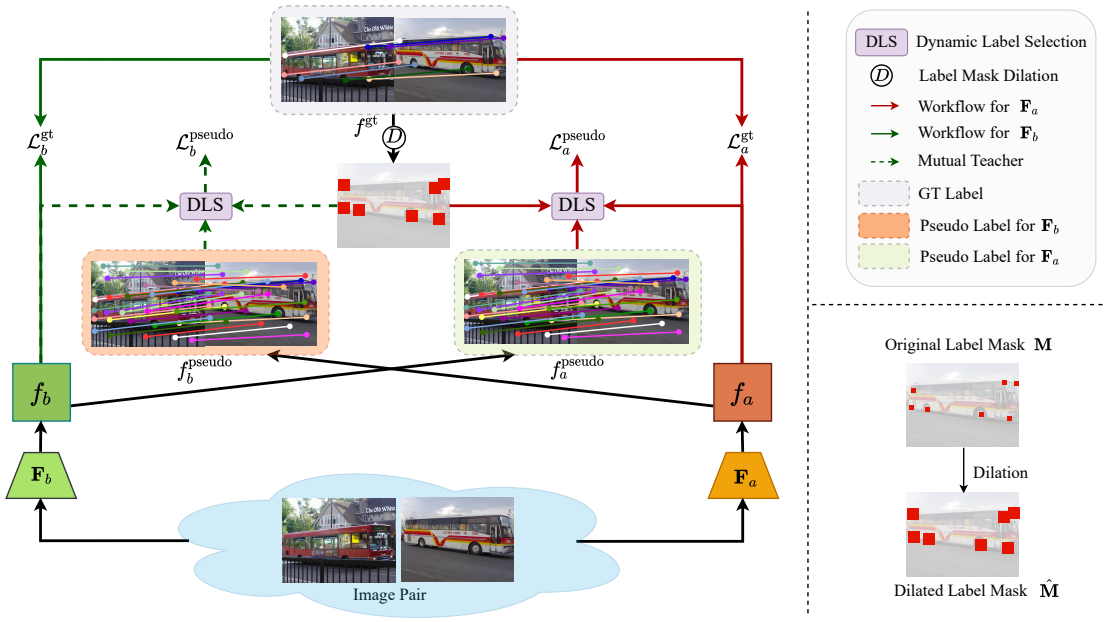
Given $\mathbf{M}$, the objective is then defined as the L2 norm between the predicted flow and the ground-truth flow at labeled subpixel positions:

$$L^{\text{gt}}(\mathbf{p}) = \|\hat{f}(\mathbf{p}) - f^{\text{gt}}(\mathbf{p})\|_2 \cdot \mathbf{M}(\mathbf{p}) \tag{2.4}$$

where $L^{\text{gt}}(\mathbf{p})$ is the ground-truth loss at position $\mathbf{p}$. It is worth noting that our network does not involve any 4D-conv or transformer for correlation refinement [9, 18], but as shown later it achieves high performance thanks to our efficient spatial context encoder and high-resolution design.

2.4 Learning with Sparse Annotations

While our network design enables us to encode spatial context efficiently and utilize high-resolution correlation maps, the sparsely-annotated keypoint pairs (8 on average on PF-PASCAL [2])



(a) Teacher-Student Learning with sparse annotations

(b) Label Mask Dilation

Figure 2.3: **Sparse Label Densification with Teacher-Student Learning.** (a) Our Teacher-Student Learning Pipeline. Solid lines stand for Single Offline Teacher, with additional dashed lines standing for Mutual Online Teacher. (b) Illustration of Label Mask Dilation. Please refer to Sec. 3.2.3 for more details. *Best viewed in color.*

greatly hinder the learning of the dense matching model. We address this with a novel teacher-student learning framework which we will elaborate below.

Our goal is to enrich the supervision when only sparse annotations are provided, as shown in Fig. 2.3. We first densify the sparse labels with a teacher-student paradigm (Sec. 2.4.1). Then we introduce two novel techniques to denoise the generated pseudo-labels (Sec. 2.4.2): (a) leveraging spatial-priors and (b) loss-driven dynamic label selection. Finally, we investigate two variants of the proposed learning paradigm (Sec. 2.4.3).

2.4.1 Sparse Label Densification via Teacher-Student Learning

To enrich the sparse supervision signals, we generate dense pseudo-labels for unlabeled region via a teacher-student paradigm, which consists of a student model $\mathbf{F}_s$ and a teacher model

$\mathbf{F}_t$. The teacher model $\mathbf{F}_t$ trained with sparse annotations generates dense flows $\hat{f}_t$, providing pseudo-labels f_s^{pseudo} for the student model $\mathbf{F}_s$. Formally,

$$f_s^{\text{pseudo}} = \mathbf{F}_t(\mathbf{I}_a, \mathbf{I}_b), \quad f_s^{\text{pseudo}} \in \mathbb{R}^{2\times H_b \times W_b}. \tag{2.5}$$

Then, the optimization objective $L_s(\mathbf{p})$ for the student $\mathbf{F}_s$ is a combination of the ground-truth loss $L_s^{\text{gt}}(\mathbf{p})$ and a dense pseudo-label loss $L_s^{\text{pseudo}}(\mathbf{p})$ calculated as follows:

$$L_s(\mathbf{p}) = L_s^{\text{gt}}(\mathbf{p}) + \lambda L_s^{\text{pseudo}}(\mathbf{p}) \tag{2.6}$$

$$L_s^{\text{gt}}(\mathbf{p}) = \|\hat{f}_s(\mathbf{p}) - f^{\text{gt}}(\mathbf{p})\|_2 \cdot \mathbf{M}(\mathbf{p}) \tag{2.7}$$

$$L_s^{\text{pseudo}}(\mathbf{p}) = \|\hat{f}_s(\mathbf{p}) - f_s^{\text{pseudo}}(\mathbf{p})\|_2 \tag{2.8}$$

where $\hat{f}_s \in \mathbb{R}^{2\times H_t \times W_t}$ is the predicted flow of the student model $\mathbf{F}_s$ given $\mathbf{I}_a$ and $\mathbf{I}_b$, λ is the scale hyper-parameter, $\mathbf{p}$ indexes the positions in $\hat{f}_s$.

2.4.2 High Quality Pseudo-label Generation

The dense pseudo-labels generated by the teacher model are inevitably unreliable and inaccurate for supervision. To filter out erroneous pseudo-labels, we use: (a) label filtering based on spatial priors, and (b) loss-driven dynamic label selection.

2.4.2.1 Spatial-prior Based Label Filtering.

Our key insight is that, as the annotated keypoints are in the object foreground region, we are able to suppress noisy background pseudo-labels by exploiting the spatial-smoothness prior

of the semantic correspondence in the neighborhood of the sparse keypoints. Motivated by this, we generate a densified binary label mask $\hat{\mathbf{M}}$ via dilating the sparse label mask $\mathbf{M}$ as follows, which will be used for label filtering:

$$\bar{\mathbf{M}} = \mathbf{M} * \mathcal{K} \tag{2.9}$$

$$\hat{\mathbf{M}}(\mathbf{p}) = \begin{cases} 1 & \text{if } \bar{\mathbf{M}}(\mathbf{p}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

where $*$ is a convolution operator with zero padding, $\mathcal{K} \in \mathcal{R}^{k \times k}$ is a kernel filled with one with k as the kernel size. Note that dilation here refers to expanding the existing foreground region in $\mathbf{M}$. Compared with using CAM [64] or uncertainty estimation [21], our proposed label filtering technique is easy to implement and utilizes the spatial prior around the sparse annotations.

Given the dilated label mask $\hat{\mathbf{M}}$, the pseudo-loss $\hat{L}_s^{\text{pseudo}}(\mathbf{p})$ for the student model is calculated as below:

$$\hat{L}_s^{\text{pseudo}}(\mathbf{p}) = \|\hat{f}_s(\mathbf{p}) - f_s^{\text{pseudo}}(\mathbf{p})\|_2 \cdot \hat{\mathbf{M}}(\mathbf{p}). \tag{2.11}$$

where $\mathbf{p}$ indexes the positions. In this way, we are able to significantly suppress noisy background pseudo-labels as shown in Sec 3.2.4.3.

2.4.2.2 Loss-driven Dynamic Label Selection.

While many background pseudo-labels can be filtered out by our dilated label mask, some noisy labels still exist due to inaccurate predictions from the teacher model. To further filter

out inaccurate labels, we introduce a loss-driven label selection strategy following the small-loss principle [65]. Denoting R as the ratio of pixels being selected, we choose the pixel set $\mathcal{P}$ on the foreground region in $\hat{\mathbf{M}}$ with the smallest loss as below:

$$\mathcal{P} = \underset{\bar{\mathcal{D}}:|\bar{\mathcal{D}}|\geq R(T)N_{\hat{\mathbf{M}}}\wedge\bar{\mathcal{D}}\subseteq\hat{\mathcal{D}}}{\arg\min} \sum_{\mathbf{p}\in\bar{\mathcal{D}}} \hat{L}_s^{\text{pseudo}}(\mathbf{p}) \tag{2.12}$$

$$\hat{\mathcal{D}} = \{\mathbf{p} \mid \hat{\mathbf{M}}(\mathbf{p}) = 1\} \tag{2.13}$$

where $R(T)$ controls the selection percentage in training epoch T, $\mathbf{p}$ indexes the positions, $\hat{\mathcal{D}}$ is a collection of foreground positions in the dilated label mask $\hat{\mathbf{M}}$, $N_{\hat{\mathbf{M}}}$ is the total number of non-zero positions in $\hat{\mathbf{M}}$.

Hence the final optimization objective $\mathcal{L}_s$ for the student model $\mathbf{F}_s$ over an image pair is a combination of the sparse ground-truth loss $\mathcal{L}_s^{\text{gt}}$ at labeled positions $\mathcal{G} = \{\mathbf{p} \mid \mathbf{M}(\mathbf{p}) = 1\}$ and the dense pseudo loss $\mathcal{L}_s^{\text{pseudo}}$ at selected positions $\mathcal{P}$ as below:

$$\mathcal{L}_s = \mathcal{L}_s^{\text{gt}} + \lambda\mathcal{L}_s^{\text{pseudo}} \tag{2.14}$$

$$\mathcal{L}_s^{\text{pseudo}} = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{p}\in\mathcal{P}} \hat{L}_s^{\text{pseudo}}(\mathbf{p}) \tag{2.15}$$

$$\mathcal{L}_s^{\text{gt}} = \frac{1}{|\mathcal{G}|}\sum_{\mathbf{p}\in\mathcal{G}} L_s^{\text{gt}}(\mathbf{p}) \tag{2.16}$$

2.4.3 Variants of Teacher-Student Learning

To investigate the optimization strategy of our proposed teacher-student learning, we propose two variants of our learning strategy, including (a) single offline teacher, and (b) mutual

online teachers, which are detailed below.

2.4.3.1 Single offline Teacher (ST).

This variant consists of two learning stages. Specifically, we first learn a baseline network, which acts as the teacher model, given spare ground-truth annotation only. In the second stage, the pseudo-labels are generated by the fixed teacher network as described in Sec. 2.4.1 and Sec. 2.4.2. Given the enriched supervision, we then train the student model from scratch, which is used for the inference stage finally.

2.4.3.2 Mutual online Teacher (MT).

Inspired by the recent advances in multi-view learning [66, 67], we additionally explore a one-stage variant with two mutual online teachers which learn from scratch. We simultaneously train two networks of the same architecture, each of which takes predictions from the other network as the pseudo-labels for optimization. These two networks can learn knowledge of correspondence with enriched pseudo-labels from each other. The one with a higher validation performance is selected for the inference stage.

Specifically, we maintain two networks $\mathbf{F}_s$ and $\mathbf{F}_t$ of the same architecture. The network $\mathbf{F}_t$ (resp. $\mathbf{F}_s$) provides its predicted flow $\hat{f}_t$ (resp. $\hat{f}_s$) as the pseudo-label f_s^{pseudo} (resp. f_t^{pseudo}) for the peer network $\mathbf{F}_s$ (resp. $\mathbf{F}_t$). Both networks use the shared dilated label mask $\hat{\mathbf{M}}$ for label

filtering. For each model, the pseudo-loss filtered by dilated label masks is described as below:

$$\hat{L}_s^{\text{pseudo}}(\mathbf{p}) = \|\hat{f}_s(\mathbf{p}) - f_s^{\text{pseudo}}(\mathbf{p})\|_2 \cdot \hat{\mathbf{M}}(\mathbf{p}) \tag{2.17}$$

$$\hat{L}_t^{\text{pseudo}}(\mathbf{p}) = \|\hat{f}_t(\mathbf{p}) - f_t^{\text{pseudo}}(\mathbf{p})\|_2 \cdot \hat{\mathbf{M}}(\mathbf{p}), \tag{2.18}$$

where $\mathbf{p}$ indexes the position, f_s^{pseudo} (resp. f_t^{pseudo}) equals to $\hat{f}_t$ (resp. $\hat{f}_s$). $\hat{L}_s^{\text{pseudo}}(\mathbf{p})$ and $\hat{L}_t^{\text{pseudo}}(\mathbf{p})$ will then go through the dynamic label selection procedure as described in Sec. 2.4.2 to compute pseudo-label loss $\mathcal{L}_s^{\text{pseudo}}$ and $\mathcal{L}_t^{\text{pseudo}}$, respectively. The final optimization objective for each model is a combination of sparse ground-truth loss and pseudo loss as below:

$$\mathcal{L}_s = \mathcal{L}_s^{\text{gt}} + \lambda \mathcal{L}_s^{\text{pseudo}} \tag{2.19}$$

$$\mathcal{L}_t = \mathcal{L}_t^{\text{gt}} + \lambda \mathcal{L}_t^{\text{pseudo}} \tag{2.20}$$

2.5 Experiments

We evaluate our method on the supervised semantic correspondence task by conducting comprehensive experiments on three public benchmarks: PF-PASCAL [2], PF-WILLOW [3], and SPair-71k [1]. In the following sections, we first elaborate on the implementation details of our proposed method in Sec. 2.5.1, and follow that with the quantitative and qualitative comparison with prior state-of-the-art (SOTA) competitors in Sec. 2.5.2. Then, we provide ablation studies and comprehensive analysis in Sec. 3.2.4.3. For more detailed results and analysis, we refer readers to the supplementary material.

Table 2.1: **Comparison with SOTA methods on SPair-71k [1]**. Per-class and overall PCK ($\alpha_{\text{bbox}} = 0.1$) results are shown in the table. Numbers in bold indicate the best performance and underlined ones are the second best. All models in this table use ResNet101 as the backbone. *Sup.* denotes the type of supervision. * means the backbone is finetuned. † means ground truth bbox used.

Sup.	Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	mbike	person	plant	sheep	train	tv	**all**
self	CNNGeo [15]	23.4	16.7	40.2	14.3	36.4	27.7	26.0	32.7	12.7	27.4	22.8	13.7	20.9	21.0	17.5	10.2	30.8	34.1	20.6
	A2Net [42]	22.6	18.5	42.0	16.4	37.9	30.8	26.5	35.6	13.3	29.6	24.3	16.0	21.6	22.8	20.5	13.5	31.4	36.5	22.3
weak	WeakAlign [16]	22.2	17.6	41.9	15.1	38.1	27.4	27.2	31.8	12.8	26.8	22.6	14.2	20.0	22.2	17.9	10.4	32.2	35.1	20.9
	NCNet [17]	17.9	12.2	32.1	11.7	29.0	19.9	16.1	39.2	9.9	23.9	18.8	15.7	17.4	15.9	14.8	9.6	24.2	31.1	20.1
trn-none /	HPF [28]	25.2	18.9	52.1	15.7	38.0	22.8	19.1	52.9	17.9	33.0	32.8	20.6	24.4	27.9	21.1	15.9	31.5	35.6	28.2
val-strong	SCOT [45]	34.9	20.7	63.8	21.1	43.5	27.3	21.3	63.1	20.0	42.9	42.5	31.1	29.8	35.0	27.7	24.4	48.4	40.8	35.6
strong	DHPF [20]	38.4	23.8	68.3	18.9	42.6	27.9	20.1	61.6	22.0	46.9	46.1	33.5	27.6	40.1	27.6	28.1	49.5	46.5	37.3
	PMD [21]	38.5	23.7	60.3	18.1	42.7	39.3	27.6	60.6	14.0	54.0	41.8	34.6	27.0	25.2	22.1	29.9	70.1	42.8	37.4
	MMNet* [68]	43.5	27.0	62.4	27.3	40.1	50.1	37.5	60.0	21.0	56.3	50.3	41.3	30.9	19.2	30.1	33.2	64.2	43.6	40.9
	CHM [46]	49.6	29.3	68.7	29.7	45.3	48.4	39.5	64.9	20.3	60.5	56.1	46.0	33.8	44.3	38.9	31.4	72.2	55.5	46.3
	CATs†* [9]	52.0	34.7	72.2	34.3	49.9	57.5	43.6	66.5	24.4	63.2	56.5	**52.0**	42.6	41.7	43.0	33.6	72.6	58.0	49.9
	PMNC* [38]	54.1	35.9	74.9	36.5	42.1	48.8	40.0	72.6	21.1	67.6	58.1	50.5	40.1	**54.1**	43.3	35.7	74.5	59.9	50.4
	Ours (ST)*	56.9	37.0	76.2	33.9	50.1	51.7	42.4	68.2	22.4	70.7	61.0	47.7	43.6	47.8	47.8	38.6	77.0	67.1	52.4
	Ours (MT)*	**57.1**	**40.3**	**78.3**	**38.1**	**51.8**	**57.8**	**47.1**	**67.9**	**25.2**	**71.3**	**63.9**	49.3	**45.3**	49.8	**48.8**	**40.3**	**77.7**	**69.7**	**55.3**

2.5.1 Implementation Details

2.5.1.1 Datasets.

SPair-71k is a newly-released challenging and largest-scale benchmark [1]. There are keypoint-annotated 70,958 image pairs with large viewpoint and scale variation in diverse scenes. SPair-71k [1] is a reliable test bed for studying real problems of semantic matching. *PF-PASCAL* dataset [2] contains 1351 image pairs with limited variability and scale, which is approximately split into 700, 300, and 300 pairs for train, val, and test set, resp. *PF-WILLOW* [3] dataset consists of 900 image pairs of 4 categories, which is a widely-used benchmark for the verification of generalization ability.

2.5.1.2 Evaluation Metric.

In line with prior work, we report the percentage of correct keypoints (PCK) [69]. The predicted keypoints are considered to be correct if they lie within $\alpha \cdot \max(h, w)$ pixels from the ground-truth keypoints for $\alpha \in [0, 1]$, where h and w are the height and width of either an image (α_{img}) or an object bounding box (α_{bbox}).

2.5.1.3 Experimental Configuration.

For the feature extractor, we use ResNet-101 [70] pre-trained on ImageNet with a single feature at stride 16. Learnable parameters are randomly initialized. For our base model, we set Efficient SCE kernel size $K = 7$ and $d_g = 2048$ for SPair-71k; $K = 13$ and $d_g = 1024$ for PF-PASCAL, resp. We upsample the correlation map to stride 4 for high-resolution loss. For label mask dilation, dilation kernel size $k = 7$ is set for both SPair-71k and PF-PASCAL by validation search. For dynamic label selection, we set $R(T)$ linearly increases from the ratio of 20% to 90% in a duration of 10 epochs for both SPair-71k and PF-PASCAL. λ is 10.0 for weighting pseudo-loss. We strictly follow previous work for data augmentation [9] (e.g., color jittering) except that [9] uses ground truth box for random crop while we do not. An AdamW optimizer with a learning rate of 3e-6 for the backbone and 3e-5 for the remaining parameters are used. All the implementations are in PyTorch [71].

Images of all three datasets are resized to 256×256. Our model is trained on PF-PASCAL and SPair-71k, resp. Following the previous work [9,18,20], we validate the generalization ability of our method by testing on PF-WILLOW with our model trained on PF-PASCAL without any finetuning.

Table 2.2: **Comparison with SOTA methods on PF-PASCAL** [2]. Numbers in bold indicate the best performance and underlined ones are the second best. † means ground truth bbox used.

Sup.	Methods	PCK@α_{img}			α_{bbox}
		$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$	$\alpha = 0.1$
none	PF-LOM$_{\text{HOG}}$ [2]	31.4	62.5	79.5	45.0
self	CNNGeo$_{\text{ResNet-101}}$ [15]	41.0	69.5	89.4	68.0
weak	WeakAlign$_{\text{ResNet-101}}$ [16]	49.0	74.8	84.0	72.0
	NC-Net$_{\text{ResNet-101}}$ [17]	54.3	78.9	86.0	70.0
	DCCNet$_{\text{ResNet-101}}$ [18]	55.6	82.3	90.5	-
	GSF$_{\text{ResNet-101}}$ [72]	62.8	84.5	93.7	-
trn-none val-strong	HPF$_{\text{ResNet-101}}$ [28]	60.1	84.8	92.7	78.5
	SCOT$_{\text{ResNet-101}}$ [45]	63.1	85.4	92.7	-
strong	SCNet$_{\text{VGG-16}}$ [73]	36.2	72.2	82.0	48.2
	ANCNet$_{\text{ResNet-101}}$ [37]	-	86.1	-	-
	DHPF$_{\text{ResNet-101}}$ [20]	75.7	90.7	95.0	87.8
	PMD$_{\text{ResNet-101}}$ [21]	-	90.7	-	-
	MMNet$_{\text{ResNet-101}}$ [68]	77.6	89.1	94.3	-
	CHM$_{\text{ResNet-101}}$ [46]	80.1	91.6	-	-
	CATs† $_{\text{ResNet-101}}$ [9]	75.4	92.6	<u>96.4</u>	89.2
	PMNC$_{\text{ResNet-101}}$ [38]	**82.4**	90.6	-	-
	Ours (ST)$_{\text{ResNet-101}}$	81.4	<u>92.9</u>	96.1	<u>90.5</u>
	Ours (MT)$_{\text{ResNet-101}}$	<u>81.5</u>	**93.3**	**96.6**	**91.2**

2.5.2 Comparison with State-of-the-art Methods

2.5.2.1 SPair-71k.

We compare our method with the most recent work [9, 20, 28, 38, 45, 68] on SPair-71k in Table 2.1. Our two variant settings (ST and MT) both achieve an overall SOTA results, with our method (MT) achieving an overall PCK ($\alpha_{\text{img}} = 0.1$) of 55.3%, outperforming the previous SOTA [38] by a large margin (4.9%). Note that our method does not involve any parameterized correlation refinement compared with [38], clearly illustrating the power of the proposed pipeline. Fig. 2.4 shows qualitative results on SPair-71k. We observe that our method is robust to diverse

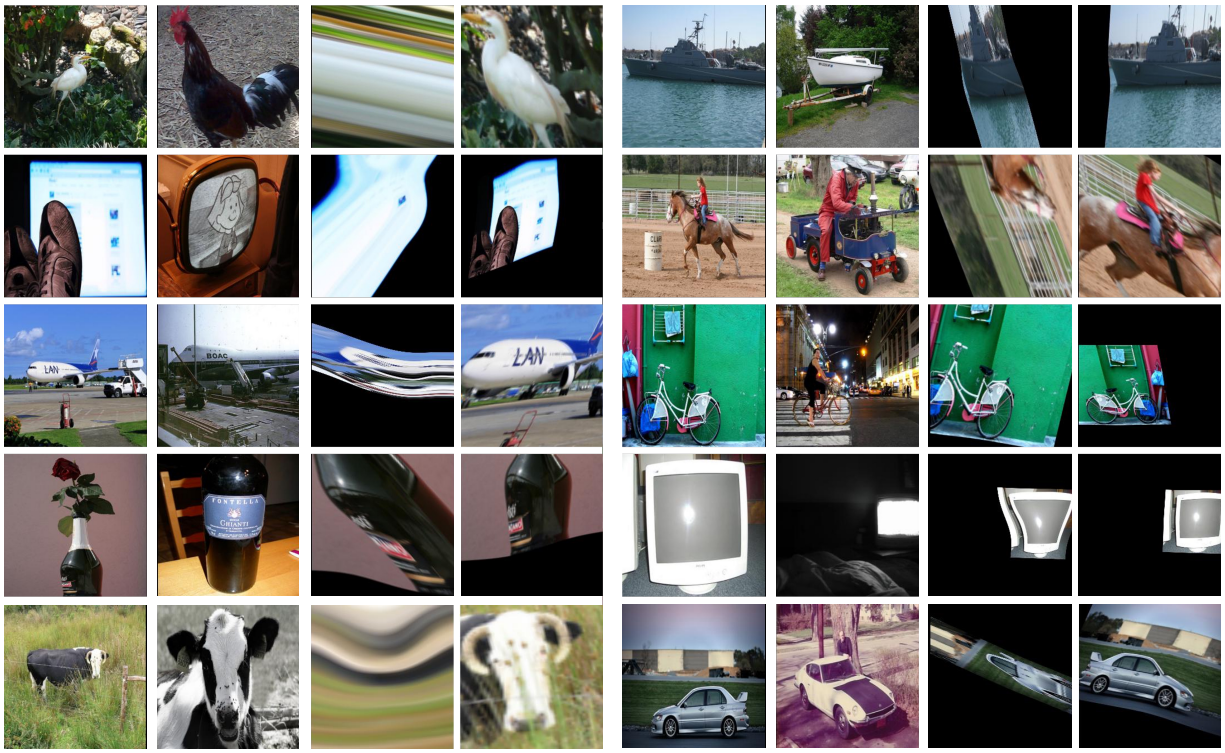


Figure 2.4: **Qualitative results of our method on SPair-71k [1].** From left to right are source image, target image, result from CATs [9], and result from ours (MT), resp.

variations in scale and viewpoint thanks to our enriched training signals.

2.5.2.2 PF-PASCAL.

Our results on PF-PASCAL are summarized in Table 2.2. Our method outperforms the previous state-of-the-art [9, 38] on almost all thresholds even if the performance on PF-PASCAL is near saturated, reaching a new SOTA of 93.3% PCK ($\alpha = 0.1$). Note that even if we did not use sophisticated parameterized correlation map refinement as in PMNC [38], we can still achieve comparable PCK at $\alpha = 0.05$.

Table 2.3: **Comparison with SOTA methods on PF-WILLOW [3].** Numbers in bold indicate the best performance and underlined ones are the second best. † means ground truth bbox used.

Sup.	Methods	PCK@α_{bbox}		
		$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$
none	PF-LOM$_{\text{HOG}}$ [2]	28.4	56.8	68.2
self	CNNGeo$_{\text{ResNet-101}}$ [15]	36.9	69.2	77.8
weak	WeakAlign$_{\text{ResNet-101}}$ [16]	37.0	70.2	79.9
	NC-Net$_{\text{ResNet-101}}$ [17]	44.0	72.7	85.4
	DCCNet$_{\text{ResNet-101}}$ [18]	43.6	73.8	86.5
	GSF$_{\text{ResNet-101}}$ [72]	47.0	75.8	88.9
trn-none	HPF$_{\text{ResNet-101}}$ [28]	45.9	74.4	85.6
val-strong	SCOT$_{\text{ResNet-101}}$ [45]	47.8	76.0	87.1
strong	SCNet$_{\text{VGG-16}}$ [73]	38.6	70.4	85.3
	DHPF$_{\text{ResNet-101}}$ [20]	49.5	77.6	89.1
	PMD$_{\text{ResNet-101}}$ [21]	-	75.6	-
	CHM$_{\text{ResNet-101}}$ [46]	52.7	<u>79.4</u>	-
	CATs† $_{\text{ResNet-101}}$ [9]	50.3	79.2	**90.3**
	Ours (ST)$_{\text{ResNet-101}}$	<u>53.5</u>	<u>79.4</u>	89.5
	Ours (MT)$_{\text{ResNet-101}}$	**54.1**	**80.0**	<u>89.8</u>

2.5.2.3 PF-WILLOW.

We test on PF-WILLOW [3] using our model trained on PF-PASCAL [2] to verify dataset generalization ability of our method. As shown in Table 2.3, our method (MT) outperforms the prior SOTA [9, 46] in $\alpha = 0.05, 0.1$ by 1.4% and 0.6%, resp, indicating superior dataset generalization ability of our learning method. Note that our method (MT) are 0.5% behind at $\alpha = 0.15$ compared with [9], we argue that CATs [9] used ground truth bounding box during training while we did not.

Table 2.4: **Effects of each component on SPair-71K [1] test split.** HRLoss refers to high-resolution loss, FT refers to finetuning the backbone, Teacher-Student here refers to the variant with mutual online teacher.

Model ID	Efficient-SCE	HRLoss	Finetune	Teacher-Student	Label Denoise		PCK
					Dynamic Selection	Mask Dilation	
A0	-	-	-	-	-	-	14.7
A1	✓	-	-	-	-	-	33.4
A2	✓	✓	-	-	-	-	40.6
A3	✓	✓	✓	-	-	-	49.8
A4	✓	✓	✓	✓	-	-	49.7
A5	✓	✓	✓	✓	✓	-	51.5
A6	✓	✓	✓	✓	✓	✓	**55.3**

2.5.3 Ablation Study

In this section, we conduct ablation studies to verify the effectiveness of each individual module of the proposed model. We train all the variants on the training split of SPair-71k [1] and report PCK ($\alpha_{\text{bbox}} = 0.1$) on the test split. Each ablation experiment is conducted under the same experimental setting for a fair comparison.

2.5.3.1 Effect of individual modules.

Table 3.8 summarizes the ablation results of each individual module. First, we note that applying the proposed Efficient-SCE (ID A1) yields significant gain over the baseline (ID A0), showing the effectiveness of the proposed feature enhancement module. Second, enforcing high-resolution loss improves to a remarkable 49.8% after finetuning (ID A3). Our proposed network achieves competitive results without any Conv4D or transformer modules for correlation map refinement. Third, densifying labels combined with our two denoising techniques achieves 5.5% boost further and promotes the performance to 55.3%, showing the effectiveness of our proposed

Table 2.5: **Comparing single offline teacher and mutual online teacher** setting on SPair-71K [1] test split.

Variant Setting	PCK
none	49.8
single offline teacher	52.4
mutual online teacher	**55.3**

Table 2.6: **Effects of kernel size** for label mask dilation on SPair-71K [1] test split.

Dilation Kernel Size	PCK
none	49.8
3	54.0
7	**55.3**
15	52.0

learning strategy. In contrast, teacher-student learning alone without any denoising provides little boost as the dense pseudo-labels might be too noisy (ID A4).

2.5.3.2 Single offline Teacher vs. Mutual online Teacher.

Table 2.5 shows the comparison between our proposed two variants of teacher-student learning. Both settings have greatly surpassed the performance of the base network (ID A3), showing the effectiveness of our proposed label densification strategy. We note that the mutual online teacher setting is 2.9% higher than the single offline teacher setting. The reason could be that performance is bounded by the fixed teacher model while the mutual online teacher setting could improve each other over the training process.

2.5.3.3 Effects of kernel size for label mask dilation.

Table 2.6 summarizes the results of different kernel size for label mask dilation. When increasing the kernel size, the performance rises first but then drops, with kernel size 7 being the best, which demonstrates the necessity of restricting pseudo-labels in a meaningful local neighborhood.

2.6 Conclusion

In this work, we propose a novel teacher-student learning paradigm in order to address the challenge of label sparsity for semantic correspondence task. In our teacher-student paradigm, we generate dense pseudo-labels by the teacher networks which are trained with sparse annotations. To improve quality of pseudo-labels, we develop two novel techniques to denoise pseudo-labels. Specifically, we first dilate the sparse label masks derived from the sparse keypoint annotations to suppress background pseudo-labels. A dynamic label selection strategy is then introduced to further filter noisy labels. We investigate two variants of the proposed learning paradigm, a single offline teacher setting, and a mutual online teacher setting. Our method achieves state-of-the-art performances on three standard datasets. The effectiveness of our method provides new insight into the problem, and is one step closer towards a more realistic application of semantic correspondence.

Chapter 3: Learning Instance Alignment in Videos

3.1 What is Point Supervision Worth in Video Instance Segmentation?

3.1.1 Introduction

Paper List: [4, 5, 31, 74–80]

Video instance segmentation (VIS) is emerging as a challenging vision task which aims to detect, segment, track objects in continuous videos [4]. It has achieved increasing attention recently [81–87] due to its wide real-world applications such as video editing, 3D reconstruction [88–90], and view point estimation [91, 92].

Annotating per-frame object masks in videos is time-consuming and even more challenging than annotating image instance segmentation masks. Due to the limited video annotations, a common strategy to train a video instance segmentation model is to first train on image instance segmentation datasets with ground truth mask and category annotations (COCO [93]), and then finetune on video instance segmentation datasets with ground truth masks, category and tracking annotations [94, 95]. However, the categories in image datasets do not necessarily fully overlap with the video datasets, and hence adapting models from the image domain to the video domain has challenging generalization issues due to the emergence of new categories.

There are some recent efforts to reduce video annotation cost for video instance segmen-

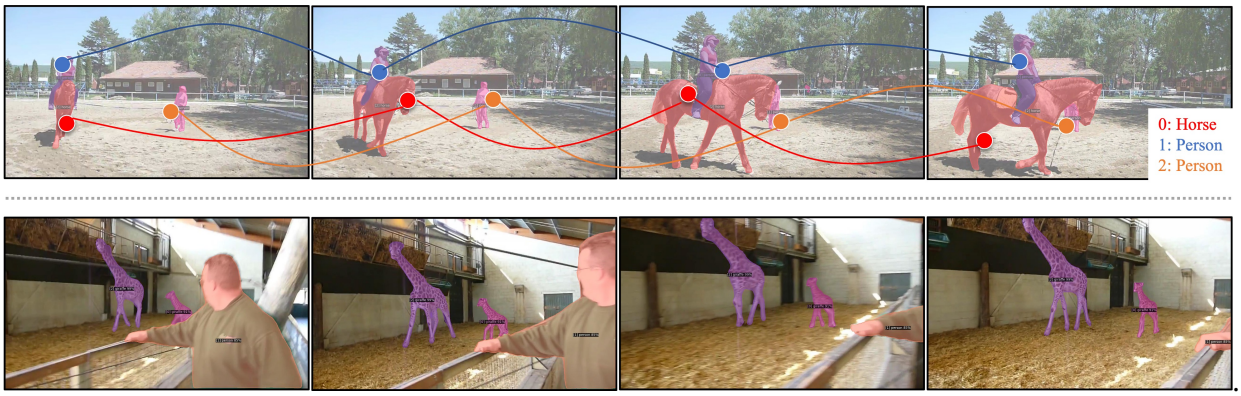


Figure 3.1: **Point-supervised video instance segmentation in this work (YoutubeVIS-2021)** Top: point-level annotations in the training set (pseudo masks generated from our method overlaid); Bottom: mask predictions in the validation set.

tation. They propose to learn with sub-sampled video frames [76], category annotations in videos [74], or without any video annotations [75]. However, these approaches either still require dense masks in the sub-sampled frames [76] or are barely competitive compared with supervised approaches [74] or can only handle the overlapped category between video and image dataset [75]. These limitations of existing approaches show that it is still unclear what is the optimal way to reduce annotation cost for video instance segmentation.

In this paper, we ask the question: *To what level can we reduce human annotations in videos and still train an accurate model for video instance segmentation?* We believe that point supervision presents a "sweet spot" for annotating objects in videos. Point annotations are significantly cheaper than other alternatives, such as bounding boxes, as one simple click only costs around 1 second [96, 97]. In the most extreme case we considered, every object instance in a frame only contains one labeled point, as shown in Figure 3.1. Despite the many benefits of point supervision, using it to supervise dense predictions is challenging and raises many ill-conditioned issues, such as the sparsity of ground truth and the lack of informative negative samples.

In this work, we introduce a point-supervised VIS framework (PointVIS) to address these

challenges. PointVIS leverages the knowledge from image-based (COCO) pre-raining and guides the VIS task in an open set manner.

Our main contributions are:

- PointVIS is the first attempt to comprehensively investigate video instance segmentation with point-level supervision. Our work significantly reduces the amount of required annotations in VIS and opens up the possibility to address the task with minimal supervision.
- PointVIS overcomes the challenges in point-supervised video instance segmentation, using the proposed class-agnostic proposal generation and a point-based matcher.
- PointVIS is simple to implement, achieving competitive results compared with fully-supervised methods on three major VIS benchmarks.
- We further conduct comprehensive studies in different settings of points with important observations, providing a deeper understanding on what kind of point supervision matters in the VIS task.

The key challenge of using point supervision is the sparsity of ground truth, which further leads to the lack of informative negative samples that are along the boundaries of dense mask annotations. We address these challenges by proposing to (1) generate class-agnostic spatio-temporal instance proposals without video mask annotation and (2) match these proposals with our point annotations to obtain dense pseudo-label for training the VIS model. We leverage COCO pre-trained image instance segmentation model to generate per-frame instance proposals and use bipartite matching on query embeddings to convert them to spatio-temporal video instance proposals. We further design a loss function to match these proposals with our sparse point annotations as there could be multiple proposals that overlap with a single point annota-

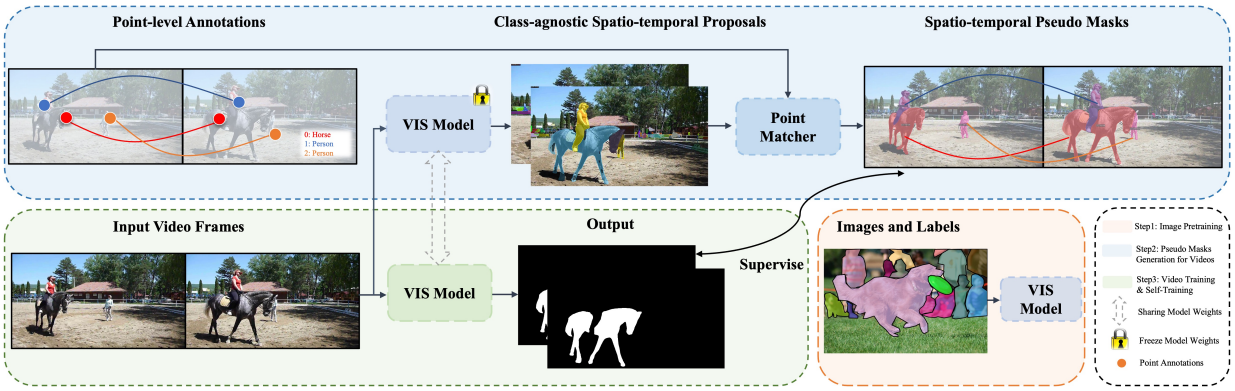


Figure 3.2: **Method Overview.** Our method consists of class-agnostic spatial-temporal proposal generation, a spatio-temporal point-based matcher to match object queires with point annotations for high-quality pseudo-label generation, and self-training to mitigate the domain gap between images and videos. See text for details.

tion. We show that both of these designs are crucial to generating high-quality pseudo mask annotations for learning with only point supervision. An overview of our approach is shown in Figure 3.2.

We further conduct comprehensive studies on how point annotations affect VIS and make the following important observations: (1) even one positive point annotated per video object already achieves good performance, retaining 87% of the performance of fully-supervised methods on Youtube-VIS 2019 [4]; (2) given positive points, increasing negative points improves performance, while adding positive points alone could provide little gain; (3) the positions of positive points have limited effect on performance while the positions of negative points matter more. These observations shed lights on what kind of point supervision matters for video instance segmentation, making it a step closer towards more realistic open-world applications.

3.1.2 Related Work

3.1.2.1 Image instance segmentation

Supervised Instance Segmentation. Instance segmentation requires bounding-box regression, classification, and pixel-level segmentation of all objects present in images. After the success of two-stage instance segmentation methods [98–102], one-stage instance segmentation methods [103–108] not only significantly improve the accuracy but also reduce the computation cost. Recent video instance segmentation approaches [76, 87, 95, 109–111] are built on those one-stage methods.

Weakly supervised instance segmentation. Due to the heavy segmentation annotation costs, weakly supervised instance segmentation is a potential way to reduce this cost. Zhou et al. [112] and Ahn [113] propose learning instance segmentation with image-level annotations. Another collection of instance segmentation approaches leverage box-level supervision [31, 114–116]. Recently, image instance segmentation with both box and point-level supervision shows competitive results [97]. Note that our point-supervised video instance segmentation differs from [97] as we do not use any additional bounding box annotations.

3.1.2.2 Video Instance Segmentation

Supervised video instance segmentation. Video Instance Segmentation is a joint task of detection, instance segmentation, and tracking, which was first proposed by Yang et al. [4]. Most previous approaches [81, 82, 86, 87, 111] follow the tracking-by-detection paradigm, which segments and classifies objects and then associates objects across frames. Another trend of video

instance segmentation [84, 117, 118] follows the clip-match paradigm, where the video is separated into multiple overlapped clips, and objects are detected and segmented in each clip and then associated across different clips. Tracking-by-regression approaches [109, 110] have also been proposed to generate detections and associate object bounding boxes of the same objects across contiguous frames. Recently, transformer-based approaches for VIS have attracted much attention [94, 95, 119, 120]. Our work is built on top of MinVIS [76] for its excellent performance in VIS using image-based training. We verify that this design also benefits our task and provides better inductive bias in our weakly-supervised learning setting.

Weakly/Semi-supervised video segmentation. As annotations of VIS are expensive, there are emerging methods that aim to learn VIS with reduced annotations [74–76]. Liu et al. [74] propose to learn VIS by using image-level annotations with correspondences [18, 22, 113]. Fu *et al.* [75] propose to leverage instance segmentation annotations of COCO dataset and learn VIS without video annotations. However, these methods either still require dense masks in the sub-sampled frames [76] or are barely competitive compared with supervised approaches [74] or can only handle the overlapped category between video and image dataset [75]. In contrast, our PointVIS can cover and handle all categories with largely reduced annotation cost. In addition, we for the first time show that video instance segmentation with one point per object can achieve decent performance compared with the fully-supervised counterparts.

3.1.2.3 Point-supervised methods.

Recently point-level supervision has attracted growing attention in computer vision, including object localization [121], object detection [122, 123], image instance segmentation [97, 124], and image panoptic segmentation [125, 126]. Point-level interactions are also popular in the field

of interactive image segmentation [127–129] and interactive video segmentation [130–133], with a focus on label propagation or reducing interaction time. The closest work to ours is PointPanoptic [126] where they use a single point to train image panoptic segmentation. Note that this fundamentally differs from our work, as we aim for reducing video annotations by utilizing pretrained image representations while PointPanoptic [126] aims for reducing image annotations by training from scratch. To our best knowledge, there is no prior work that utilizes point-level supervision for VIS.

3.1.3 Method

Our PointVIS design is motivated by three major challenges in point-supervised VIS:

Sparsity in ground truth. Unlike masks and boxes, points do not provide detailed localization and shape information about objects, especially their boundaries, sizes and extreme points. Points are also sparse, which causes difficulties during model training. Our task is further complicated by the lack of negative supervision when only positive points are annotated, which is important to learn a correct decision boundary. Since many VIS methods involve COCO pretraining, we similarly leverage this pipeline to enrich our supervision using object shape priors.

Matching sparse annotations. An important step in recent Transformer-based instance segmentation models is to match their mask proposals with ground truth masks using some costs, usually defined as the intersection-over-union (IoU) between two masks. This matching step is the key to enable end-to-end training without non-maximum suppression (NMS). However, such matching process becomes problematic in our case, since points do not provide informative measurement on how accurate a mask is. Therefore, special designs need to be taken into consideration when defining the cost of matching step in PointVIS.

Novel categories. In real world applications, there is no guarantee that the categories of downstream VIS tasks overlap with those in images. Therefore, a point-supervised framework has to handle arbitrary new categories and learn efficiently with sparse supervision.

Our solution. We therefore propose several designs in PointVIS to address the above challenges. We first pre-train instance segmentation models on COCO and use the pretrained models to generate spatiao-temporal mask proposals in training videoes. Despite the issue mentioned above, this class-agnostic proposal generation mechanism works well in open set scenarios, providing good coverage on categories not seen in COCO. We then propose a point-based matcher that incorporates annotation-free negative cues from other instances in the same video frame. Finally, we address the generalization issue for new categories in videos by conducting self-training to mitigate the domain gap and refine our results. We iterate the training with pseudo masks from prior round. These solutions together allow us to learn VIS with points effectively.

3.1.3.1 Problem Setup

Due to the high annotation cost in videos, in the standard supervised setup, a prevalent strategy to train a video instance segmentation model is to first train on image instance segmentation datasets, and then finetune on video instance segmentation datasets. Formally, there is a full-labeled image instance segmentation dataset $\mathcal{D}_I$ with category space size $\mathbf{O}_I$ and a full-labeled video instance segmentation dataset $\mathcal{D}_V$ with category space size $\mathbf{O}_V$, both of which are annotated with object masks.

In our proposed point-supervised setup, we adopt a video dataset with point annotations (denoted as $\mathcal{D}_{V_p}$) instead of with masks for each video object as in $\mathcal{D}_V$. Specifically, given a

video $\mathbf{V} \in \mathbb{R}^{H \times W \times T}$ of T RGB frames with width W and height H, the point annotations for the j^{th} video object in the video $\mathbf{V}$ is denoted as $\mathbf{G}_j = \{\{\mathbf{P}_j^t, \mathbf{L}_j^t\}_{t=1}^T, \mathbf{b}_j\}$, where $\mathbf{P}_j^t \in \mathbb{R}^{N_j^t \times 2}$ are the x-y coordinates of the annotated points at the t^{th} frame, $\mathbf{L}_j^t \in \mathbb{R}^{N_j^t}$ are the corresponding binary labels for the annotated points indicating foreground or background, N_j^t is the number of annotated points, $\mathbf{b}_j \in \mathbb{R}^{\mathbf{O}_V}$ is the one-hot category label and t is the time index. In this setup, a video instance segmentation model $\mathbf{F}$ is first trained on image dataset $\mathcal{D}_I$ with full masks, resulting an image model $\mathbf{F}(;\theta_I)$. Then $\mathbf{F}(;\theta_I)$ is finetuned on video dataset $\mathcal{D}_{V_p}$ with point-level annotations, resulting the final video instance segmentation model $\mathbf{F}(;\theta_{V_p})$.

Supervising a model solely based on annotated points with a loss function like cross-entropy can lead to the model collapsing, especially if only positive points are annotated. Therefore, densifying point annotations and including negative supervision signals become crucial for learning a structured dense prediction task via points.



Figure 3.3: Visualization of point annotations and pseudo masks obtained by our method on Youtube-VIS 2019 [4] (row1), Youtube-VIS 2021 [4] (row2), and OVIS [5] (row3) training set.

3.1.3.2 Learning VIS with Sparse Points

Class-agnostic proposal generation. To obtain meaningful dense supervision with abundant negative signals, we shift one step back to the image model instead of dwelling on a degenerated video model finetuned with points. Recall that we have image instance segmentation datasets with mask annotations ready at hands. A pretrained image model on such datasets should already know rough object shape, even if never trained on videos and the categories do not fully overlap. Given this simple yet non-trivial discovery, we propose to generate dense class-agnostic spatial-temporal proposals for each video by utilizing a pretrained image model that encodes rich shape prior.

It is challenging to obtain spatio-temporal proposals when video supervision is unavailable. We address this challenge by leveraging COCO pre-trained image instance segmentation model [108] to generate per-frame instance proposals and use bipartite matching on query embeddings to convert them to spatio-temporal video instance proposals. While previous work has used a similar technique for VIS with full video supervision [76], we here show that this enables cross-domain video instance proposals generation trained with image dataset only.

Concretely, given model $\mathbf{F}(;\theta_I)$ trained on image dataset $\mathcal{D}_I$ and a video sequence $\mathbf{V}$ from $\mathcal{D}_{V_p}$, we obtain the initial proposals $\hat{\mathbf{R}}$ for $\mathbf{V}$ by conducting inference as below:

$$\hat{\mathbf{R}} = \mathbf{F}(\mathbf{V};\theta_I) = \{\hat{\mathbf{M}}_r, \hat{c}_r\}_{r=1}^{R} \tag{3.1}$$

, where $\hat{\mathbf{M}} \in \mathbb{R}^{H\times W\times T}$ is a spatial-temporal proposal with continous logits after sigmoid but before binarization, $\hat{c}_r$ is the confidence score, R is the maximum number of proposals for a

video (e.g. 100).

Given the above dense proposals, there is still one open-set problem worth resolving. As there could be new categories in $\mathcal{D}_{V_p}$ and $\mathbf{F}(;\theta_I)$ has never been finetuned on $\mathcal{D}_{V_p}$, the confidence score $\hat{c}_r$ is not meaningful for every video. To represent the confidence of class-agnostic proposals without the reliance on categories, we propose to use maskness score following [134] to obtain the confidence of a mask as $c_r = \frac{1}{H \times W \times T} \sum_{x,y,z} \hat{\mathbf{M}}_r(x, y, z)$ where x,y are the x-axis and y-axis spatial coordinates, t indexes the time.

Therefore, our final class-agnostic dense spatio-temporal proposals $\mathbf{R}$ for a video $\mathbf{V}$ is denoted as $\mathbf{R} = \{\mathbf{M}_r, c_r\}_{r=1}^{R}$, where $\mathbf{M}_r \in \mathbb{R}^{H \times W \times T}$ is the binarized mask of $\hat{\mathbf{M}}_r$, c_r is the maskness score.

Point-based matcher. Given the dense class-agnostic proposals, the next challenge is to assign the best proposal to a video object and produce the final pseudo mask when only point annotations are available in videos. This can be challenging since there may be multiple proposals overlapping with a single point, making it difficult to determine which proposal provides the best boundary information.

To address this issue, we develop a matching cost function that combines cues from both point annotations and spatio-temporal proposals to effectively match proposals and video objects with points. With our proposed point-based matcher, we can formulate the pseudo-label filtering problem as a bipartite matching problem between the proposals and the objects. We provide details on the matching process and the design of the matching cost below.

Specifically, we search for a permutation $\hat{\sigma}$ between the set of dense proposals and the set of video ground truth with points given a video. Assuming R is larger than the number of

objects in the video, we consider $\mathbf{G}$ as a set of video ground truth with size R padded with $\varnothing$ (no object). To find a bipartite matching between $\mathbf{R}$ and $\mathbf{G}$, we search for a permutation $\sigma \in \Omega_R$ of R elements with the lowest cost:

$$\hat{\sigma} = \underset{\sigma \in \Omega_R}{\arg\min} \sum_{j=1}^{R} \mathcal{L}_{\text{match}}(\mathbf{G}_j, \mathbf{R}_{\sigma(j)}) \tag{3.2}$$

where $\mathcal{L}_{\text{match}}(\mathbf{G}_j, \mathbf{R}_{\sigma(j)})$ is a pair-wise matching cost between ground truth $\mathbf{G}_j$ and a proposal with index $\sigma(j)$. This optimal assignment is computed efficiently with the Hungarian algorithm following prior work [135]. Next, we present the specific matching cost $\mathcal{L}_{\text{match}}$ given point annotations.

To penalize proposals that do not overlap with the annotated points consistently, we first define an annotated cost $\mathcal{L}_{\text{ann}}(\mathbf{G}_j, \mathbf{R}_{\sigma(j)})$ calculated over T frames as below:

$$\mathcal{L}_{\text{ann}}(\mathbf{G}_j, \mathbf{R}_{\sigma(j)}) = \sum_{t=1}^{T} \sum_{k=1}^{N_j^t} \mathbb{1}[\mathbf{M}_{\sigma(j)}(\mathbf{P}_j^t(k), t) \neq \mathbf{L}_j^t(k)] \tag{3.3}$$

where $\mathbb{1}[\cdot]$ is the indicator function, k is the point index, t is the time index, $\mathbf{P}_j^t(k) \in \mathbb{R}^2$ is the x-y coordinates for the k^{th} point of the j^{th} video object at the t^{th} frame.

To combat a server lack of negative points during matching, especially when only positive points are annotated, we further develop a cross instance negative cost $\mathcal{L}_{\text{cineg}}(\mathbf{G}_j, \mathbf{R}_{\sigma(j)})$ to filter out multiple overlapping proposals. The key idea is that the positive points for one video object can serve as accurate negative points for the other video objects in the same video frame. By aggregating the positive point annotations from other video objects in the same frame, we obtain additional accurate negative point annotations for each video object. Therefore, this annotation-

Method	Dataset	Sup.	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
TeViT [136]	YouTube-VIS-2019	$\mathcal{M}$	56.8	80.6	63.1	52.0	63.3
IDOL [79]	YouTube-VIS-2019	$\mathcal{M}$	64.3	87.5	71.0	55.6	69.1
MinVIS [76]	YouTube-VIS-2019	$\mathcal{M}$	61.6	83.3	68.6	54.8	66.6
PointVIS (P1)	YouTube-VIS-2019	$\mathcal{P}_1$	53.9 (87.5%)	75.7 (90.9%)	61.8 (90.1%)	47.5 (86.7%)	61.4 (92.2%)
PointVIS (P1N1)	YouTube-VIS-2019	$\mathcal{P}_2$	59.6 (96.7%)	83.3 (100%)	67.1 (97.8%)	52.7 (96.2%)	63.8 (95.8%)
SeqFormer [80]	YouTube-VIS-2021	$\mathcal{M}$	51.8	74.6	58.2	42.8	58.1
IDOL [79]	YouTube-VIS-2021	$\mathcal{M}$	56.1	80.8	63.5	45.0	60.1
MinVIS [76]	YouTube-VIS-2021	$\mathcal{M}$	55.3	76.6	62.0	45.9	60.8
PointVIS (P1)	YouTube-VIS-2021	$\mathcal{P}_1$	46.3 (83.7%)	70.5 (92.0%)	51.1 (82.4%)	37.7 (82.1%)	52.9 (87.0%)
PointVIS (P1N1)	YouTube-VIS-2021	$\mathcal{P}_2$	48.5 (87.7%)	73.0 (95.3%)	54.4 (87.7%)	41.7 (90.8%)	54.1 (89.0%)
MaskTrack [137]	Occluded VIS	$\mathcal{M}$	28.9	56.3	26.8	13.5	34.0
IDOL [79]	Occluded VIS	$\mathcal{M}$	42.6	65.7	45.2	17.9	49.6
MinVIS [76]	Occluded VIS	$\mathcal{M}$	39.4	61.5	41.3	18.1	43.3
PointVIS (P1)	Occluded VIS	$\mathcal{P}_1$	28.6 (72.6%)	49.6 (80.7%)	27.5 (66.6%)	15.0 (82.9%)	32.8 (75.8%)
PointVIS (P1N1)	Occluded VIS	$\mathcal{P}_2$	28.6 (72.6%)	51.2 (83.3%)	27.2 (65.9%)	14.7 (81.2%)	32.2 (74.4%)

Table 3.1: **Full mask ($\mathcal{M}$) vs. our point supervision ($\mathcal{P}$) on validation set of YouTube-VIS 2019 [4], YouTube-VIS 2021 [4], and OVIS [5].** All results below are based on Swin-L backbone. Our PointVIS results are with self-training.

free $\mathcal{L}_{\text{cineg}}$ can penalize inaccurate proposals that overlap with the positively annotated points in other video instances.

In addition to the annotated cost and cross instance negative cost, we also define a maskness cost $\mathcal{L}_{\text{maskness}}$ as the negative of the maskness score to favor confident proposals. As a result, our proposed matching cost $\mathcal{L}_{\text{match}}$ is a weighted combination of the annotated cost, cross instance negative cost, and maskness cost as below:

$$\mathcal{L}_{\text{match}} = \lambda_1 \mathcal{L}_{\text{ann}} + \lambda_2 \mathcal{L}_{\text{cineg}} + \lambda_3 \mathcal{L}_{\text{maskness}} \tag{3.4}$$

where λ_1, λ_2 and λ_3 are the weight balancing parameters.

To handle the birth and death of objects, we compute the matching cost only over video frames where objects show up. After matching, we remove the pseudo-labels for frames where objects die. With our carefully designed matching cost, we can obtain high quality dense pseudo-mask for objects with point annotations via the optimal permutation.

Self-training for generalization. With the above high-quality pseudo masks, we can train our video instance segmentation model on videos with standard loss for mask prediction (cross-entropy and dice loss) and cross-entropy loss for classification following the existing work [76, 95].

To generalize from images to videos for new categories, we conduct self-training by regenerating pseudo-labels again from our finetuned video model. The reason is that our pseudo-labels are initially generated from an image model that has never been trained on videos, and there is obviously a domain gap. During self-training, we use confidence score instead of maskness score for pseudo-label matching as the model has been finetuned on videos.

3.1.4 Experiments

We evaluate our method on three VIS datasets: YouTube-VIS 2019 [4], YouTube-VIS 2021 [4], and Occluded VIS [5]. We describe our experimental setup (Sec.3.2.4.1), compare PointVIS with SOTA fully-supervised methods (Sec.3.2.4.2), and provide an ablation study (Sec. 3.2.4.3). For more details, please refer to the supplementary material.

3.1.4.1 Experimental Setup

Datasets. YouTube-VIS 2019 [4] is a popular dataset for VIS with 2,883 labeled videos, 131K instance masks, and 40 classes. **YouTube-VIS 2021** [4] is an improved version with 8,171 unique video instances and 232k instance masks. **OVIS** [5] is a recently proposed challenging VIS dataset with heavy occlusion and long sequences, containing 296k instance masks and 5.8 instance per video from 25 classes. We synthesize point annotations by randomly sampling points

given the ground truth mask in each frame.

Architecture and optimization. We build our PointVIS on top of MinVIS [76] by strictly following its model architecture, training hyper parameters and losses. The only two modifications during video training are: 1) we use pseudo masks obtained from our method given point-level annotations while MinVIS uses annotated masks; 2) we use larger iterations as pseudo-labels require longer time for convergence. All models are pre-trained with COCO instance segmentation [93] then finetuned on videos with Swin backbone [138] unless otherwise stated. For our point-based matcher, we set $\lambda_1 = 5.0$, $\lambda_2 = 5.0$, and $\lambda_3 = 2.0$ for all three datasets. We conduct one iteration of self-training.

Baselines. We propose new baselines for comparison as no prior work is directly applicable to our new point-supervised setting for VIS. 1) **VISP**: naive training MinVIS with points, where only locations with annotated points are supervised during video training. 2) **VISP+CINeg**: adding annotation-free negative point loss to VISP by enforcing cross-instance negative cues on top of it, as described in Sec. 3.1.3.2. This strategy is denoted as "CINeg". 3) **VISC**: to ablate the benefits of points and decouple the impact of pretrained image model, we obtain pseudo-labels by selecting the top k most confident proposals from our proposal set $\hat{\mathbf{R}}$ sorted by the confidence score $\hat{c}_r$.

We also include the existing unsupervised/semi-/weakly-supervised VIS work to compare with our point-supervised setting as follows. 1) **VIS-Unsup** [75]: pretrained COCO image model built on top of SOLO [139] without finetuning on videos. 2) **VIS-Semi** [75]: finetuning on videos on top of VIS-Unsup but without video annotations. Note that VIS-Unsup and VIS-Semi can only handle the overlapped categories, therefore we report AP/AR (seen) in line with [75]. 3) **VIS-Weak** [74]: the first weakly-supervised VIS model that uses category annotations only in each

Model	$\mathcal{L}_{\text{Ann}}$	$\mathcal{L}_{\text{CINeg}}$	$\mathcal{L}_{\text{maskness}}$	Self-Train	mAP
MinVIS [76] (Upperbound)	-	-	-	-	55.3
VISP (P1)	-	-	-	-	0.4
VISP+CINeg (P1)	-	-	-	-	9.7
PointVIS (P1)	✓	-	-	-	40.4
PointVIS (P1)	✓	✓	-	-	45.7
PointVIS (P1)	✓	✓	✓	-	46.0
PointVIS (P1)	✓	✓	✓	✓	47.3

Table 3.2: **Effects of each component on YouTube-VIS 2019 [4] val-dev.**

Model ID	Sampling method for Pos	Sampling method for Neg	mAP
PointVIS (P1)	Random	-	46.0
PointVIS (P1)	Distance Transform	-	47.1
PointVIS (P1N1)	Random	Random (In-box)	48.6
PointVIS (P1N1)	Random	Random (Out-box-but-in-200%-box)	48.0

Table 3.3: **Analysis of point selection bias on YouTube-VIS 2019 [4] val-dev.**

video frame.

Metrics. We use AP and AR for evaluation and train on the training split and evaluate on the validation split using public evaluation servers of the three datasets. Baselines VISC, VIS-Unsup [75] , and VIS-Semi [75] can only handle categories overlapped with the image model, so we report AP/AR (seen) computed by averaging only over the overlapped categories for these baselines following [75].

3.1.4.2 Comparison with Fully-Supervised SOTA

We compare our PointVIS with recent fully-supervised methods including IDOL [79], MinVIS [76], TeViT [136], SeqFormer [80], and MaskTrack [137], as shown in Table 3.7. Note that

MinVIS [76] serves as the fully-supervised counterpart of PointVIS, therefore we compute the retention rate as the performance of PointVIS divided by the performance of MinVIS. We use $\mathcal{M}$ to indicate full supervision from dense masks, and $\mathcal{P}_n$ to indicate sparse supervision from n points. P1 means only one positive point is labeled. P1N1 means one positive and one negative opint are labeled, and similarly for larger number of points.

YouTube-VIS 2019. With one single point per object (P1), PointVIS achieves 53.9 mAP. With two points per object (P1N1), PointVIS achieves 59.6 mAP, which is 96.7% of the supervised counterpart and even surpassed recent fully-supervised method TeViT [136] by nearly 3 AP. These competitive results demonstrate the potential of our point-supervised video instance segmentation framework.

YouTube-VIS 2021. On this more challenging dataset, we can still achieve 87.7% performance of the supervised counterpart, reaching 48.5 AP, which is only 3 AP away from recent fully-supervised approach SeqFormer.

OVIS. PointVIS achieves 28.6 AP on this challenging dataset, 72.6% of the supervised counterpart, and matches previous fully-supervised methods [137]. The retention rate is relatively lower compared to the other two datasets, likely due to the quality of spatio-temporal proposals, which degrades due to extremely long sequences in OVIS.

Model ID	CINeg	DPPointMatcher	P1	P10	P1N1	P5N5
VISP	-	-	0.4	0.5	27.6	33.0
VISP+CINeg	✓	-	9.7	10.0	45.6	48.9
PointVIS (Ours)	✓	✓	46.0	45.9	48.6	49.4

Table 3.4: **Effects of additional points on YouTube-VIS 2019 [4] val-dev.** "DPPointMatcher" means Dense Pseudo via Point Matcher. "CINeg" means enforcing additional negative point loss.

Model	Backbone	Sup.	mAP	AP_{50}	AP_{75}	AR_1	AR_{10}
VIS-Unsup [75]	R50	$\mathcal{V}$	23.9	43.3	21.5	26.7	37.3
VIS-Semi [75]	R50	$\mathcal{V}$	38.3	61.1	39.8	36.9	44.5
VISC	R50	$\mathcal{V}$	42.0	62.5	47.3	48.7	56.7
PointVIS (P1-Ours)	R50	$\mathcal{P}_1$	47.0	67.4	53.4	44.4	50.9
PointVIS (P1-Ours)	Swin-L	$\mathcal{P}_1$	58.6	80.3	66.2	54.1	64.2

(a) Evaluation on seen categories

Model	Backbone	Sup.	mAP	AP_{50}	AP_{75}	AR_1	AR_{10}
VIS-Weak [74]	R50	$\mathcal{C}$	10.5	27.2	6.2	12.3	13.6
PointVIS (P1)	R50	$\mathcal{P}_1$	38.5	58.9	41.2	36.4	46.4
PointVIS (P1-Ours)	Swin-L	$\mathcal{P}_1$	52.5	74.5	59.2	47.2	61.5

(b) Evaluation on all categories

Table 3.5: **Comparison with baselines on YouTube-VIS 2019 [4] validation set.**

3.1.4.3 Ablation Study

In this section, we conduct ablation studies to 1) verify the effectiveness of each individual component of our proposed framework; 2) analyze training with subsampled frames; 3) analyze the effect of number of points; 4) analyze point selection bias and 5) compare with baselines on YouTube-VIS 2019 [4]. We split the training set of YouTube-VIS 2019 [4] into train-dev for training and val-dev for testing with Swin-B backbone unless otherwise stated. We do not use self-training by default for simplicity unless otherwise stated. Each ablation is conducted under the same experimental setting for a fair comparison.

3.1.4.4 Effects of model components.

Table 3.8 presents the ablation results for each component of our model. The VISP baseline, which trains the VIS model with only point supervision, yields a near-zero AP. Incorporating

cross instance negatives (VISP+CINeg) improves it to 9.7 mAP. Our method achieves a decent mAP of 40.4 by incorporating pseudo masks generated through our point-based matcher with only $\mathcal{L}_{\text{Ann}}$ in the matching process. Further addition of $\mathcal{L}_{\text{CINeg}}$ and $\mathcal{L}_{\text{maskness}}$ results in a significant boost of 5.7 mAP, highlighting the importance of incorporating cross instance negatives. Finally, with self-training reducing the domain gap, our PointVIS achieves 47.3 mAP, which is 85.6% of the fully-supervised upperbound.

3.1.4.5 Analysis of subsampling frames.

PointVIS can be extended to the setting of using a subset of frames following MinVIS [76] (Table 3.6). With point labels in only 1% frames, PointVIS (55.0 AP) is only 6.6 AP behind the one uses full-masks in 100% frames, and achieves a retention rate of 93.2% of its counterpart with full-masks in 1% frames. This indicates the practical utility of our work.

PointVIS could be further easily extended to a per-frame setting where frames are annotated with points in parallel w/o tracking annotations. The main change here is that we cannot compute spatio-temporal matching cost. Instead, we change matching to per-frame by computing the matching cost for each annotated frame independently. We refer the resulting per-frame model as PointVIS (PF). PointVIS (PF) achieves competitive results while PointVIS has its own advantages, especially with less frames (Table 3.6).

3.1.4.6 Analysis of point selection bias.

To analyze how point selection bias affects performance, we synthesized point annotations using different sampling methods (Table 3.3). Under the P1 setting, we compared random sampling with human distance transform and found that different methods achieved similar results,

but distance transform performed around 1AP better. This suggests that our method is generally robust to annotated point locations, but human annotation can potentially yield higher gains. Under the P1N1 setting, we compared sampling negatives inside versus outside the bounding box and found that the latter was only 0.6 AP behind, indicating that negatives do not need to be constrained within boxes.

3.1.4.7 Analysis with additional points.

Table 3.4 summarizes our results on adding more points. Increasing the number of positive points alone (P1 vs P10) did not improve performance because the model lacks additional cues about the background region. Adding both negative and positive points improved the performance of VISP and VISP+CINeg, which rely heavily on point annotations for knowing foreground and background. Our proposed method's performance improved significantly with just one more negative point and remained stable across all point settings, indicating its robustness.

3.1.4.8 Comparison with additional baselines.

Table 3.5 shows the comparison with additional baselines. Compared with VIS-Unsup [75] without using video annotations, our PointVIS outperforms VIS-Unsup by more than 10 AP with little annotation overhead. Compared with the VISC baseline that does not use video annotations but generating pseudo-labels by confidence, PointVIS outperforms it by 5 AP showing the benefit of point annotations.

Model	Sup.	Matching	1%	5%	10%	100%
MinVIS [19]	$\mathcal{M}$	-	59.0	59.3	61.0	61.6
PointVIS (PF)	$\mathcal{P}_2$	Per-frame	52.3 (88.6%)	54.5 (91.9%)	54.7 (89.7%)	57.3 (93.0%)
PointVIS	$\mathcal{P}_2$	Spatio-temporal	55.0 (93.2%)	55.5 (93.6%)	56.1 (92.0%)	57.4 (93.2%)

Table 3.6: **PointVIS (P1N1) with subsampled video frames on YouTube-VIS 2019 validation set (w/o self-training).**

3.1.5 Conclusion

In this work, we address the challenging point-supervised video instance segmentation problem and introduce a point-supervised VIS framework. The key ingredients are utilizing object shape prior from a pretrianed COCO image instance segmentation model, and our proposed spatio-temporal point-based matcher for generating high-quality dense pseudo-labels for videos. Our method allows us to reduce annotations to only one point for each object in a video frame, yet retaining high quality mask predictions close to full supervision. We conduct comprehensive experiments on three datasets and achieve competitive performance compared with the fully-supervised methods.

Limitations. While PointVIS shows promising results in the direction of weakly-supervised video instance segmentation, it has some limitations. Typical failure cases are shown in Figure 3.4. We observe missing instances and temporal inconsistency. One hypothesis is that our performance is bounded by the quality of proposals, and we believe stronger video instance architecture can mitigate this gap. Another potential direction is to utilize video correspondences for label propagation and other denoising techniques for higher-quality proposals. We leave these as future work.



Figure 3.4: **Failure cases on OVIS [5].** We observe temporal inconsistency (tiger in the top left) or missing instances (person in white) in our pseudo masks.

Broader Impacts. We open up the possibility of learning video instance segmentation with point-level supervision. We hope that our framework can be used to largely reduce annotation cost for a wide range of video recognition tasks in computer vision, including video object detection, video object segmentation, and video panoptic segmentation.

3.2 UVIS: Unsupervised Video Instance Segmentation

3.2.1 Introduction

Video Instance Segmentation (VIS) [4] is the task of classifying, segmenting, and tracking individual objects within a video, with a wide range of industry applications such as in robotics, sports, autonomous driving, surveillance, AR/VR, 3D navigation [140–148], etc. It is a challenging problem due to variations in object appearances, occlusions, and cluttered scenes over time. Reliable models for VIS require dense annotated data which is costly. To circumvent the need for costly dense annotations in videos, existing methods have utilized various strategies such as pretraining on densely-labeled image datasets like COCO and finetuning on fully-labeled [95] or unlabeled [75] videos, or reducing annotations through subsampled frames [76], boxes [149], per-frame category labels [74]. However, these methods still rely on annotations [74, 76, 149] or can only handle categories that overlap with the densely-labeled image dataset [75]. In contrast, the human perception leverages image and video-level priors to effortlessly recognize, segment, and track objects [150]. This leads us to explore whether it is possible to learn an unsupervised video instance segmentation model without any dense pretraining or video annotations, covering all categories in a dataset.

Unsupervised video instance segmentation presents several challenges when only the category label set is provided for the video dataset. The first challenge is accurately predicting object boundaries without dense labeling in videos. The second challenge is conducting object classification when only the category label set is available. To address these challenges, we draw inspiration from recent advancements in large-scale unsupervised vision models, specifically the

dense shape prior in self-supervised vision model DINO [151] and the open-set recognition capability in image-caption supervised vision and language model CLIP [152]. By combining these strengths, our unsupervised VIS model can effectively segment and recognize objects within a given vocabulary set without the need for dense pretraining. To the best of our knowledge, we are the first work to explore CLIP and DINO in the field of VIS. This naturally solves the limitation of existing works that can only handle categories that overlap with densely labeled external image datasets [75].

To this end, we introduce an unsupervised video instance segmentation framework (UVIS), which is the first VIS framework, that can learn to segment all categories in videos without any dense annotation based pretraining or video annotations, as shown in Figure 5.1. Our unsupervised framework for video instance segmentation comprises of three essential steps. First, we generate class-agnostic instance masks for each video frame using a pre-trained self-supervised model [10] and equip the masks with semantic labels by using CLIP [11]. Second, we train a transformer-based video instance segmentation model by using the per-frame pseudo-masks obtained from the first step [76]. Third, during inference, we generate dense and consistent mask tubes by linking per-frame predictions using bipartite matching of query features.

To generate high-quality VIS predictions in the unsupervised setup, we further propose a novel dual-memory design on top of the above proposed UVIS framework for semantically accurate and temporally-consistent predictions. Specifically, to obtain semantically accurate pseudo-labels, we construct a class-specific prototype memory bank during pseudo-label generation. The prototypes serve as representative references, enhancing generalization and handling noisy false positives. In addition, to address the inherent limitations of the online inference pipeline in VIS that utilized only short-term information for tracking, we propose a simple but effective tracking

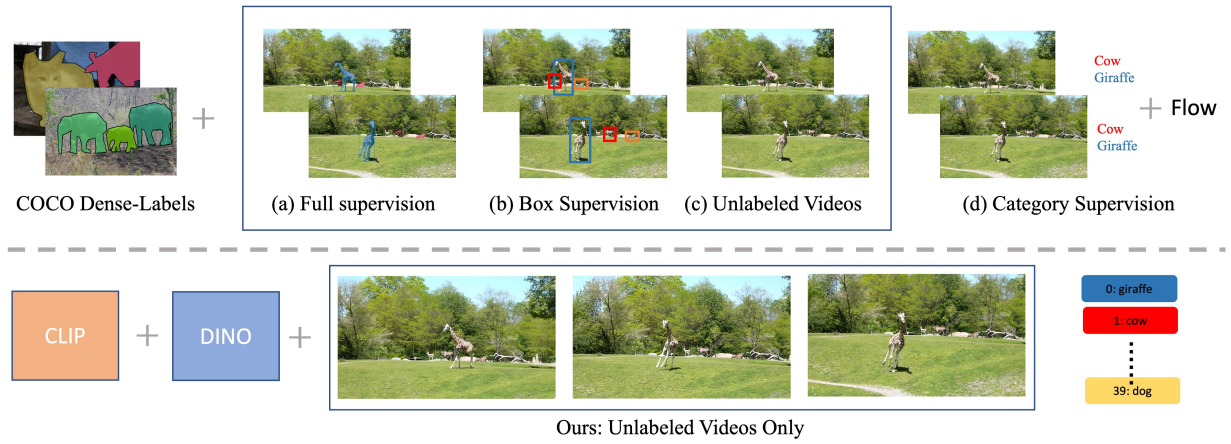


Figure 3.5: **Setting Overview.** Previous approaches have tried to use COCO dense annotations in addition to VIS dataset full supervision (a), box supervision (b) and no supervision (c). Additionally, previous works have also used flow information along with frame-level category labels (d). Our approach UVIS works in the unsupervised setting and does not require any dense labels or per frame labels and instead utilizes foundation models.

memory that models long-term temporal information. To summarize, our main contributions are as follows:

- We introduce the first VIS framework that eliminates the need for any video annotations or dense label based pretraining, thereby significantly reducing annotation costs. Our framework covers all categories in the dataset, offering a comprehensive solution.
- We propose a novel dual-memory design on top of our unsupervised VIS framework. This design includes a prototype memory filtering component, which enhances the quality of pseudo-labels, and a tracking memory bank, which captures long-term temporal information for accurate tracking.
- We conduct comprehensive experiments in three standard video instance segmentation datasets including YoutubeVIS-2019 [4], YoutubeVIS-2021 [4], and Occluded VIS [5], demonstrating the potential of our unsupervised VIS framework.

3.2.2 Related Work

Video object segmentation. Video object segmentation (VOS) [132, 153–156] is a dense binary classification problem of separating salient foreground objects from the background in videos. The most popular task in VOS is the so-called *Semi-supervised* VOS, where the goal is to segment objects in target frames given ground truth masks in the first frame. To prevent the annotation costs of exhaustively labeling each frame, several weakly and unsupervised VOS method have been proposed. [157] uses video level tags while [158] uses point supervision as weak labels to train a weakly supervised VOS system. [159–162] propose unsupervised VOS to completely eliminate the need for supervision and is a much harder problem than fully and weakly supervised VOS. In this work, we tackle a much harder problem of unsupervised Video Instance segmentation that not only does background separation but additionally performs instance segmentation that requires classification and tracking without any human supervision.

Supervised video instance segmentation. Video Instance Segmentation (VIS), initially proposed by Yang et al. [4], is an extension of image instance segmentation to videos, where the goal is to classify, segment and track objects across video frames. Early approaches [81,82,86,87,111] segment and classify objects in each frame independently, and then associated the objects across frames using heuristics such as mask or box IoU. Recently, transformer-based approaches for VIS have gained significant attention [94,95,119,120]. These approaches train VIS models in a video-based manner where they feed a clip as input and generate spatio-temporal mask predictions in one shot. A more recent development is the introduction of MinVIS [76]. This pioneering work demonstrates that a transformer-based VIS model trained solely on images can achieve competitive performance without video-based training or specialized video-based architecture design.

They observe that instance tracking naturally emerges in query-based image instance segmentation models with proper architectural constraints. We build our work on top of MinVIS [76] due to its excellent performance in VIS using image-based training. Note that such a pipeline differs fundamentally from existing approaches such as IDOL [79], which rely on post-processing steps like non-maximum suppression (NMS) during inference for tracking. However, MinVIS does not consider long-term temporal information during tracking, we address this inherent limitations by incorporating crucial temporal information in image-based VIS.

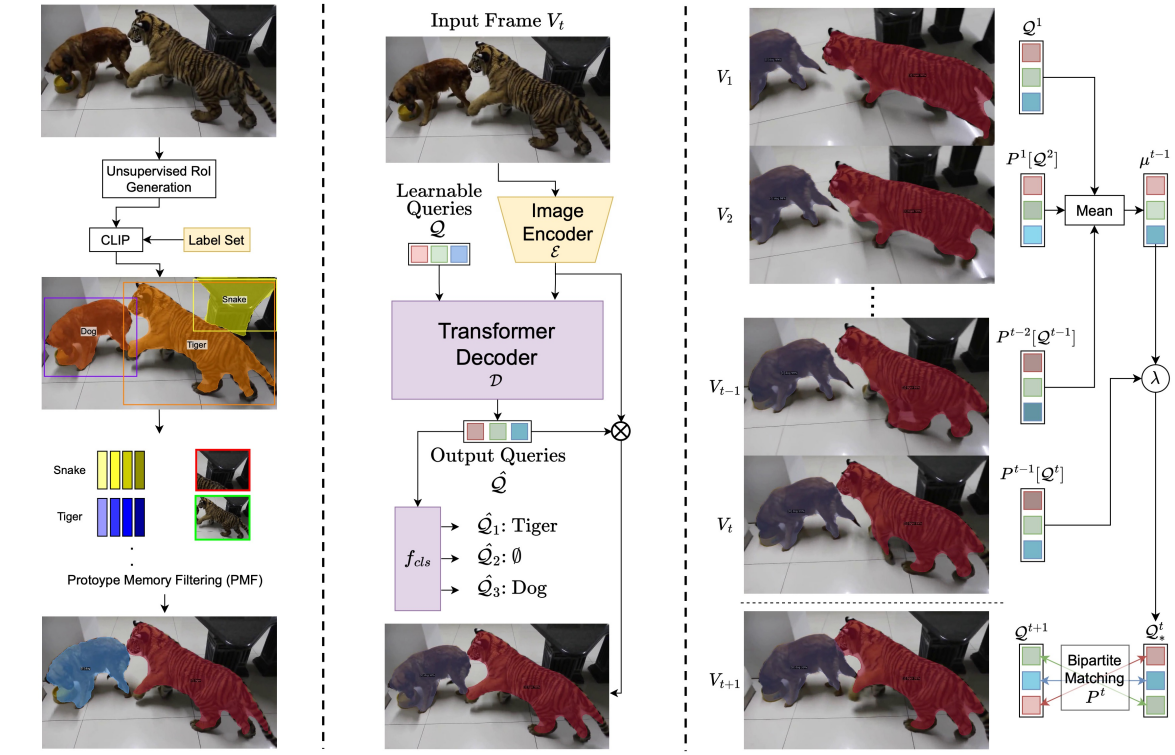
Weakly/Semi-supervised video segmentation. Reducing the annotation requirements in VIS has become a focus of recent research efforts [74–76]. Liu *et al.* [74] utilize per-frame category annotations and correspondences [18, 22, 85, 113] in videos, but exhibiting limited competitiveness compared to supervised approaches. Fu *et al.* [75] utilize instance segmentation annotations from the COCO dataset to learn VIS without video annotations, but are only applicable to overlapping categories between video and image datasets. Huang *el al.* [76] utilized annotations in sub-sampled frames but still rely on dense annotations. Huang *el al.* [77] utilized point supervision in videos. In contrast, our UVIS method handles all categories for a given vocabulary without any per-frame category/box/mask label or COCO pretraining. To the best of our knowledge, this is the first unsupervised VIS framework that achieves impressive results without any human annotations.

VL models based segmentation. Recently, foundational models trained on large amounts of unimodal or multi-modal data using weak or self-supervision have gained significant attention [163]. CLIP [152], a vision-language model using image-text pairs as supervision, has been particularly popular. CLIP has been extended to perform per-pixel detection and segmentation tasks in images [164–167]. However, the effectiveness of CLIP for videos and instance segmentation tasks

has not been thoroughly studied. In this work, we explore the use of CLIP for unsupervised VIS, which has not been adequately explored. DINO [151], a uni-modal foundational model trained on unlabeled images using self-supervised learning, demonstrates impressive segmentation capabilities. However, it cannot handle complex tasks like instance segmentation due to the lack of labeled information. Our approach combines the segmentation capabilities of self-supervised models with the zero-shot capabilities of CLIP to perform instance segmentation in videos. While NamedMask [168] is a related approach that performs semantic segmentation on images, our approach specifically focuses on VIS, offering a more comprehensive solution.

3.2.3 Method

Our objective is to learn a video instance segmentation model without groundtruth mask, box, or point annotations. The problem is challenging since we need to maintain temporal consistency while the objects may undergo appearance changes, occlusions, or partial visibility, making it difficult to track and segment them accurately over time. We build upon the recent advances in large-scale models pre-trained with Internet-scale data without any dense labels, also often called 'foundation' models. Many of these models are image and text-based and do not extend trivially to videos. Hence, in this section, we propose the framework to utilize them for the video segmentation task. Our framework consists of three steps as shown in Figure 3.6: (1) we start by generating pseudo-masks (Section 5.4.1) per video frame and build a prototype memory bank for different classes in the training data. Our proposed prototype memory encodes per-class semantic information and is used to filter the false positives improving the quality of the pseudo-labels, as shown in Figure 3.6 (a); (2) Secondly, we train a transformer-based video instance segmentation



(a) Pseudo-label generation and filtering. (b) Model-training with learnable queries. (c) Learned query-based tracking and tracking memory.

Figure 3.6: **We present our approach UVIS.** On the left we show our pseudo-label generation pipeline which involves generating masks and instance labels using CutLER [10] and CLIP [11] followed by Prototype Memory Filtering (PMF). In the center we show our model training which uses and image encoder and a transformer decoder to learn queries to predict per-frame predictions. On the right we show our proposed tracking memory approach which utilizes per frame queries and a memory based update rule to perform matching between frames to track instances and generate temporally consistent predictions.

model (Section 3.2.3.2) by using the per-frame pseudo-masks generated from the first step as shown in Figure 3.6 (b); (3) Finally, during inference, we perform bipartite matching between instances of consecutive frames and propose a tracking memory (Section. 3.2.3.3) to build dense and consistent mask tubes across the video as shown in Figure 3.6 (c).

Formally, we are given a collection of N videos $\mathcal{V} = \{V_n\}_{n=1}^{N}$ (with no pixel, box, or instance-level annotations), and a set of categories that we want to segment in these videos as $\mathcal{C} = \{l_c\}_{c=1}^{C}$, where $C = |\mathcal{C}|$ is the number of categories, and l is the text label. Note that we assume that no per-video label information is provided, , the set of videos are not tagged with

labels. Furthermore, a video may have zero or more of object instances corresponding to each label. Each video $V \in \mathcal{V}$ can also have a variable number of frames and we denote the t^{th} frame for this video as V_t.

3.2.3.1 Generating pseudo-labels for instance masks

Class agnostic mask generation. In the first step of our approach, we leverage self-supervised image models to generate pseudo-labels for video frames. Self-supervised learning (SSL) models such as [151, 169–173], are typically trained using unlabeled ImageNet [174] train set, and have an innate discriminative and localization abilities. Several methods [175–178] have been proposed to extract object masks from images using features from the SSL models. These methods typically work by performing a graph partitioning over features corresponding to various images patches and iteratively refining these partitions. We adopt CutLER [10]'s self-training strategy to generate mask and box predictions for each image using the pre-trained SSL model DINO [151]. See the supplementary material for the details of the approach. Given a video frame V_t, CutLER predicts a set of boxes $\{b_t^i\}$, masks $\{M_t^i\}$ and their corresponding objectness scores $\{o_t^i\}$ where i corresponds to the i^{th} object instance in the frame.

CLIP based Text-Instance Matching. In order to associate each mask M_t^i to the corresponding label of interest, we utilize CLIP [11], a vision-language model trained with aligned text and image data. CLIP consists of a vision module $f_{\text{vision}}^{\text{CLIP}}$ and a text module $f_{\text{text}}^{\text{CLIP}}$ to compute image and text embeddings respectively. Given an image I, the model assigns a class (from a list of classes) to it by computing the cosine similarity between the image embedding and the embeddings of a list of text prompts, and selecting the closest prompt. The list of text prompts is generated from

labels by simple strings such as "a photo of $<$ class $>$". In practice, a larger set of text prompts per class is used, we provide details in the supplemental. In our case, we generate the CLIP embeddings and the scores for each of the instance regions $\{M_t^i\}$ by using the corresponding box ($\{b_t^i\}$) to get the instance crop ($\{b_t^{i\oplus}\}$). We assign initial class labels to each of the instances using the CLIP model as following.

$$\text{class}(i) = \arg\max_{l \in \mathcal{C}} \left(f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus}) \cdot f_{\text{text}}^{\text{CLIP}}(\text{a photo of } \langle l \rangle) \right) \tag{3.5}$$

where $b_t^{i\oplus}$ is the cropped instance region for frame V_t and class(i) is the initial class assigned to the i^{th} instance by the CLIP model. We also denote the CLIP class score for this instance as $u_t^i = f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus}) \cdot f_{\text{text}}^{\text{CLIP}}(\text{a photo of } <$ class$(i) >)$.

Prototype Memory Filtering (PMF). These initial classes or pseudo-labels are often noisy and contain a lot of false positives. To address this, we create class specific prototypes using the initial class labels as following. For each class label $l \in \mathcal{C}$ we accumulate all the instance features given by $f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus})$. We apply K-Means clustering on the features and compute k^l centroids. We set k^l to be proportional to the number of instances in class l. We denote these clusters as the prototype clusters for class l. We can now compute an out-of-distribution score for each instance i such that class$(i) = l$. To do this we compute the cosine similarity between the prototype clusters of class l and CLIP features for an instance $f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus})$ which has the same predicted class. We discard all instances for which the maximum similarity with any prototype is less than a threshold τ. Together with objectness score and CLIP score, τ determines the final instances we retain for each prototype in our prototype memory. These protoypes' embeddings collectively reflect various pose, appearances and instances of the objects within the same category.

3.2.3.2 Training the segmentation model

Using the pseudo-mask labels from the last step, we next train an instance segmentation model which comprises of a convolutional image encoder $\mathcal{E}$ and a transformer decoder $\mathcal{D}$. Our setup is similar to the one used in MinVIS [76] which uses a supervised setting, as compared to our unsupervised case. We provide more details below and some additional details in the supplemental.

Given a frame V_t and corresponding pseudo-labels l_t^i and M_t^i, the model uses the fully convolutional image encoder to extract multi-scale features $\mathbf{F}_t = \mathcal{E}(V_t)$. Input to the decoder are $q \in \mathcal{Q}$ learnable query embeddings along with the the encoder features ($\mathbf{F}_t$) with $|\mathcal{Q}| = N_q$. The transformer decoder then outputs transformed queries such that $\hat{q} = \mathcal{D}(\mathbf{F}_t, q)$. Each query $\hat{q} \in \hat{\mathcal{Q}}$ is passed to a classification head f_{cls} to obtain classification scores $s = f_{\text{cls}}(\hat{q}), s \in \mathbb{R}^{1 \times C}$ where $C = |\mathcal{C}|$.

Along with classification score per query, we also obtain a segmentation masks $M \in \mathbb{R}^{N_q \times H \times W}$ for the query by convolving transformed query embedding $\hat{q}$ with last layer's features in $\mathbf{F}_t$ where H and W are the height and width of the image. In other words, $M = \sigma(\hat{q} * \mathbf{F}_t^{-1})$ where $\sigma(.)$ is the sigmoid function, $*$ is the convolution operation and -1 represents the last layer's features. During training, the classification head outputs (s) and segmentation head outputs (M) are used to perform bipartite matching between predictions and pseudo-labels that minimize the classification and segmentation losses. Once assigned, the losses are recomputed based on the matching to obtain the total loss $\mathcal{L}_{\text{vis}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{seg}}$ where $\mathcal{L}_{\text{cls}}$, the classification loss, is computed using cross entropy. $\mathcal{L}_{\text{seg}}$, the segmentation loss, is computed using binary cross entropy and dice loss [179].

Once the model is trained, the per-frame learned queries $\hat{q}$ are used to perform tracking and generate temporally consistent instance masks for each instance along with the predicted class labels.

3.2.3.3 Tracking using learned Queries and Memory

Query based Tracking. Our transformer-based model learns queries which help us identify and label each instance region. During training, we utilize per frame predictions to propogate the loss and do not use any temporal cues. But while performing inference, we require temporally and spatially consistent predictions. To extend the per-frame predictions during inference, we utilize the similarity between query embeddings between adjacent frames. Using cosine similarity based Hungarian matching between queries $\mathcal{Q}^t$ and $\mathcal{Q}^{t+1}$ of frames V_t and the next frame V_{t+1} we obtain the permutation operator (P^t) of queries ($\mathcal{Q}^t$) which assigns them to $\mathcal{Q}^{t+1}$. Utilizing a large number of queries, automatically tackles occlusion, and birth and death of tracklets by detecting null/empty masks, with enough queries remaining to track the foreground objects of interest. The final class prediction in this case for each tracklet is computed using the averaged logits across time.

Tracking with Memory Bank. While using cosine similarity to propagate instances across frames using Hungarian matching can give us tracklets and corresponding labels, we notice that for the unsupervised case, these are not very accurate. This can be attributed to the noise in pseudo-labels involved in training and the lack of encoding of variations in appearance of the same instance. To further improve the tracking we utilize a tracking memory module. This module performs averaging of the query vectors based on the matching between two frames to

use a weighted query feature from all previous frames. This adds temporal memory to each query feature which is able to encode the instance appearance over a time window instead of just focusing on the previous frame. Specifically, given two frames, V_t and V_{t+1} of video V_n, having queries $\mathcal{Q}^t$ and $\mathcal{Q}^{t+1}$, instead of matching Q^{t+1} with $\mathcal{Q}^t$ we define the averaged $\mathcal{Q}^t_*$ instead as follows:

$$\mathcal{Q}^t_* = \lambda * P^{t-1}[\mathcal{Q}^t] + (1-\lambda) * \mu^{t-1} \tag{3.6}$$

where $\mu^t \in \mathbb{R}^{N_q \times d}$ is the average memory. Here N_q is the number of per frame queries and d is the dimension of each query vector. We define μ^{t-1} as follows:

$$\mu^{t-1} = \frac{1}{t-1}\left(\mathcal{Q}^1 + P^1[\mathcal{Q}^2] + P^2[\mathcal{Q}^3] + ... + P^{t-2}[\mathcal{Q}^{t-1}]\right) \tag{3.7}$$

3.2.4 Experiments

We evaluate our method on three VIS benchmarks: YouTube-VIS 2019 [4] (YTVIS-2019), YouTube-VIS 2021 [4] (YTVIS-2021), and Occluded VIS [5] (OVIS). We describe our experimental setup in Section 3.2.4.1, compare UVIS with state-of-the-art fully-supervised approaches in Section 3.2.4.2, and provide an ablation study in Section 3.2.4.3. For more details, please refer to the supplement.

3.2.4.1 Experimental Setup

Datasets. YouTube-VIS 2019 dataset [4] (YTVIS-2019) is widely used for video instance segmentation task. It comprises 2,883 labeled videos, 131,000 instance masks, and covers 40 different classes. An improved version called **YouTube-VIS 2021** (YTVIS-2021) was also intro-

Table 3.7: **Mask ($\mathcal{M}$) / Box ($\mathcal{B}$) / category ($\mathcal{C}$) vs. our unsupervised setting on validation set of YouTube-VIS 2019 [4], YouTube-VIS 2021 [4], and OVIS [5].** * indicates training in videos without COCO pretrained model weights as initialization using authors' official code. "I-Sup." and "V-sup." indicate the supervision used in the image dataset and the video dataset, respectively. All results below are based on R50 backbone. Our UVIS achieves decent results in all three datasets without any videos annotations or dense supervision from images.

Method	Video-Dataset	COCO	I-Sup.	V-Sup.	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
IDOL [79]	YTVIS-2019	✓	$\mathcal{M}$	$\mathcal{M}$	49.5	74.0	52.9	47.7	58.7
MinVIS [76]	YTVIS-2019	✓	$\mathcal{M}$	$\mathcal{M}$	47.4	69.0	52.1	45.7	55.7
MaskFreeVIS [149]	YTVIS-2019	✓	$\mathcal{B}$	$\mathcal{B}$	42.5	66.8	45.7	41.2	51.2
MinVIS [76]	YTVIS-2019	✗	-	$\mathcal{M}$	30.3	51.3	30.1	34.7	38.1
WISE [180]	YTVIS-2019	✗	-	$\mathcal{C}$	6.3	17.5	3.5	7.1	7.8
IRN [113]	YTVIS-2019	✗	-	$\mathcal{C}$	7.3	18.0	3.0	9.0	10.7
WeakVIS [74]	YTVIS-2019	✗	-	$\mathcal{C}$	10.5	27.2	6.2	12.3	13.6
DeepSort [181]	YTVIS-2019	✗	-	-	12.5	27.1	10.8	15.3	18.1
UVIS	YTVIS-2019	✗	-	-	21.4	42.3	19.4	22.5	28.2
IDOL [79]	YTVIS-2021	✓	$\mathcal{M}$	$\mathcal{M}$	43.9	68.0	49.6	38.0	50.9
MinVIS [76]	YTVIS-2021	✓	$\mathcal{M}$	$\mathcal{M}$	44.2	66.0	48.1	39.2	51.7
MaskFreeVIS [149]	YTVIS-2021	✓	$\mathcal{B}$	$\mathcal{B}$	36.2	60.8	39.2	34.6	45.6
MinVIS [76]	YTVIS-2021	✗	-	$\mathcal{M}$	32.1	54.0	33.2	30.9	39.1
DeepSort [181]	YTVIS-2021	✗	-	-	10.3	23.0	9.4	11.9	15.5
UVIS	YTVIS-2021	✗	-	-	17.5	35.6	16.3	19.7	26.3
IDOL [79]	OVIS	✓	$\mathcal{M}$	$\mathcal{M}$	30.2	51.3	30.0	15.0	37.5
MinVIS [76]	OVIS	✓	$\mathcal{M}$	$\mathcal{M}$	25.0	45.5	24.0	13.9	29.7
MaskFreeVIS [149]	OVIS	✓	$\mathcal{M}$	$\mathcal{B}$	15.7	35.1	13.1	10.1	20.4
MinVIS [76]	OVIS	✗	-	$\mathcal{M}$	15.0	33.9	12.8	9.8	19.3
DeepSort [181]	OVIS	✗	-	-	1.6	4.0	1.4	1.9	3.9
UVIS	OVIS	✗	-	-	3.5	11.1	2.1	3.6	7.0

duced [4], featuring 8,171 unique video instances and 232,000 instance masks. **OVIS** is another challenging dataset, offering heavy occlusion, longer sequences and more number of objects. OVIS consists of 296,000 instance masks and contains an average of 5.8 instances per video across 25 classes.

Experimental Setup. We highlight our experimental setup here. For pseudo-label generation, we utilize CutLER [10] pretrained on ImageNet for class-agnostic masks generation using their Cascade-Mask-RCNN-based pretrained checkpoint. For labeling the proposed regions we use CLIP ViT-bigG-14 [152] from OpenCLIP [182]. We apply a threshold of 0.7 to both objectness

score from CutLER (o_t^i) and class score (u_t^i) from CLIP, and set $\tau = 0.7$.

For VIS architecture and optimization, we follow MinVIS [76]'s model architecture, training hyperparameters, and losses. Specifically, for the MinVIS architecture, we utilizes a ResNet-50 [183] (R50) image encoder and a transformer decoder and sets $N_q = 100$. However, we made three major modifications to it. Firstly, instead of relying on ground truth masks, we employed pseudo masks generated by our method (cf. Section 5.4.1). Secondly, instead of pretraining on COCO with dense labels, we use ImageNet classification for backbone initialization and train the transformer from scratch. Therefore, we increased the number of interactions to 320k as our setup requires more time to converge. Lastly,we incorporated our proposed tracking memory during inference and set λ=0.5.

Baselines. To our best knowledge, ours is one of the first works to introduce the task of unsupervised VIS and does not have any direct baselines to compare with. We propose a new baseline for comparison by utilizing DeepSort [181]. DeepSort [181] does not require any training and produces tracks given per-frame detections and deep features. We feed our per-frame pseudo-labels in the validation split and the associated CLIP CLS token features of each instance crop into DeepSort to generate satio-temporal masks for evaluation.

Metrics. For evaluation, we utilize the metrics of AP (Average Precision) and AR (Average Recall), and evaluate the performance on the validation split in line with the previous work [74, 76, 79].

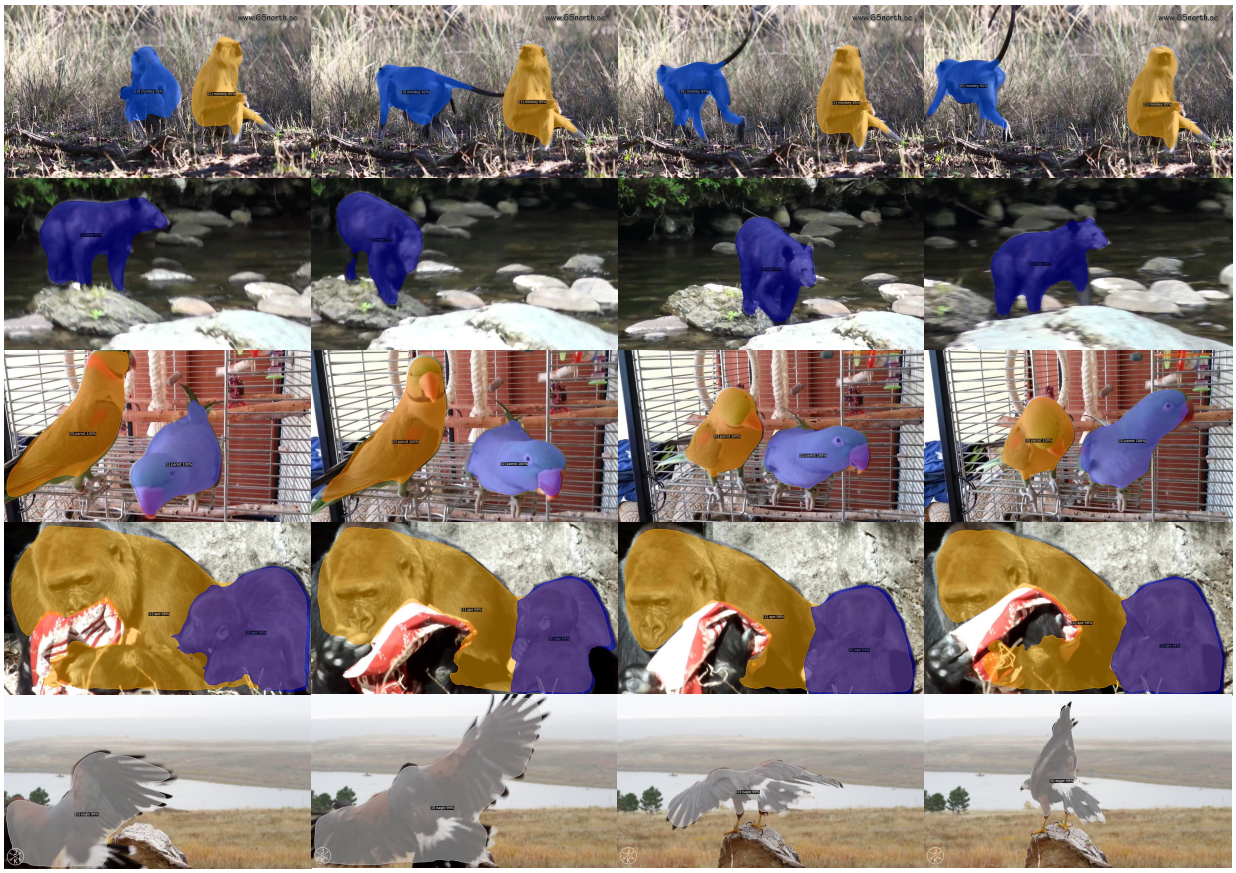


Figure 3.7: **Visualizations on YoutubeVIS-2019 [4] with our UVIS**. Each row shows temporal instance mask and class predictions. Our method is able to work for examples containing multiple instances of the same class (rows 1, 3, 4) and also when there are instances from different classes (row 5). UVIS shows promising results when instances of the same class might overlap (row 4).

3.2.4.2 Quantitative Comparison

We compare our UVIS with recent full-supervised methods including IDOL [79], MinVIS [76] as shown in Table 3.7. We also compare our method with recent box-supervised method MaskFreeVIS [149] and category-label supervised method WeakVIS [74]. Note that MinVIS [76] (w/o COCO pretraining) serves as the fully-supervised counterpart of UVIS.

YouTube-VIS 2019. As shown in Table 3.7, we achieved an impressive AP of 21.4 without relying on any annotations or COCO pretraining. This result outperforms the previous weakly-supervised method [74], which utilized per-frame category labels on videos and external flow

Table 3.8: **Ablation of different components of our pipeline on YTVIS-2019 [4] val set.**

Model ID	CLIP	Video Train	Label Denoise			Tracking Memory	AP	Δ
			Mask Score	CLIP Score	PMF			
MinVIS [76] (Upperbound)	-	-	-	-	-	-	30.3	
A1	✓	-	-	-	-	-	12.5	-
A2	✓	✓	-	-	-	-	16.6	4.1
A3	✓	✓	✓	-	-	-	18.4	5.9
A4	✓	✓	✓	✓	-	-	19.8	7.3
A5	✓	✓	✓	✓	✓	-	20.7	8.2
A6	✓	✓	✓	✓	✓	✓	**21.4**	8.9

networks, by a significant margin of 10.9 AP. Our self-constructed baseline of DeepSort [181] also performs better than WeakVIS [74] by 2 AP showing it as an effective approach for comparison. We also show qualitative results of our approach in Figure 3.7.

YouTube-VIS 2021. Our UVIS achieves 17.5 AP on this more challenging dataset. It also beats the DeepSort baseline by 7.2 AP showing the effectiveness of the proposed prototype memory filtering (PMF), training and memory based tracking. These compelling findings highlight the potential of our unsupervised video instance segmentation framework and its ability to deliver competitive results.

Occluded-VIS 2021. In the most challenging setting with the Occluded-VIS 2021 dataset, we achieve a modest result of 3.5 AP despite heavy occlusions and extremely long sequences. This is again a 1.9 AP improvement over the DeepSort baseline.

3.2.4.3 Ablation Study

We perform ablation on 1) the effects of each model component; 2) the prototype memory filtering design choice; and 3) the tracking memory component generalizability. All results are based on R50 backbone and conducted under the same configuration for a fair comparison.

Table 3.9: **Ablation of Prototype Memory Filtering (PMF) on YouTube-VIS 2019 [4] val**. Prototype Memory Filtering improves AP by 0.9.

τ	0.0	0.5	0.7	0.9
AP	19.8	20.1	**20.7**	19.9

Table 3.10: **Ablation of tracking memory on YouTube-VIS 2019 [4] val**. Our proposed tracking memory can be generalized to different datasets and different supervision settings.

Model	Sup.	Dataset	AP	AP (+Tracking Memory)
C0	-	YTVIS-2019	20.7	21.4 (+0.4)
C1	-	Occluded VIS	3.1	3.5 (+0.4)
C2	$\mathcal{I}_{\mathcal{M}} + \mathcal{V}_{\mathcal{M}}$	YTVIS-2019	47.3	50.7 (+3.4)
C3	$\mathcal{I}_{\mathcal{M}} + \mathcal{V}_{\mathcal{M}}$	Occluded VIS	26.7	27.2 (+0.5)



Figure 3.8: **Visualizations of failure cases on YoutubeVIS-2019 [4]**. On the left we show CLIP labeling failures where the CLIP model incorrectly classifies to the wrong class. In the center we show prediction inconsistencies where multiple instances are predicted as one. On the right we show temporal inconsistencies in predicted masks.

Effects of model components. We conducted an ablation study to assess the impact of each component of our model, as presented in Table 3.8. The Baseline-DeepSort achieves a validation split performance of 12.5 AP using CLIP features for tracking, without any video training. When training a VIS model with pseudo-labels without any filtering, the performance increases to 16.6 AP. By incorporating mask score and CLIP score for filtering, we observe improvements to 18.4 and 19.8 AP, respectively. Our Prototype Memory Filtering (PMF) component further enhances the performance to 20.7 AP, highlighting the importance of employing prototype memory banks for filtering out noisy labels. Finally, with the addition of our Tracking Memory component, the model achieves a 0.7 AP boost, resulting in a final performance of 21.4 AP without any supervision. This performance is only 8.9 AP lower than the upper bound achieved with full mask supervision in videos.

Prototype memory filtering ablation. We analyze the prototype memory filtering, shown in

Table 3.9, by adjusting the threshold for keeping proposals. We observe that a lower threshold is relatively safer and yields improvements (+0.3 AP) compared to not using any prototype memory filtering. As we increase the threshold (τ), the performance further improves (+0.9 AP) due to the removal of noisy labels facilitated by our prototype memory. However, we noticed that an excessively large threshold of 0.9 does not perform as well. This could be attributed to the fact that a higher threshold leads to the significant removal of true positives.

Tracking memory ablation. We ablate our tracking memory module in Table 3.10. In the unsupervised setup, incorporating the tracking memory resulted in a consistent 0.4 AP boost on both the YTVIS-2019 and Occluded VIS datasets. We observe similar consistent improvement in the supervised setup too, where we use our tracking module over the official fully-supervised MinVIS checkpoint and produce a boost of 3.4 AP on YouTube-VIS 2019 and 0.5 AP on Occluded VIS. This result highlights the importance of temporal information compared to MinVIS, which only utilizes information from consecutive frames for tracking. These experimental results confirm the generalization ability and effectiveness of our tracking memory component, both in unsupervised and supervised settings.

Failure cases. In Figure 3.8 we highlight some failure examples. We show examples where the CLIP model assigns incorrect class to the region (left). We also show multi-instance failures where the trained model assigns an instance mask covering multiple instances of the same category (center). This usually arises when the two objects occlude each other. Finally, we show temporal inconsistency failures where the model predicts masks that are not temporally consistent and end up not masking the object perfectly.

3.2.5 Conclusion

In this work, we address the challenging point-supervised video instance segmentation problem and introduce a point-supervised VIS framework. The key ingredients are utilizing object shape prior from a pretrianed COCO image instance segmentation model, and our proposed spatio-temporal point-based matcher for generating high-quality dense pseudo-labels for videos. Our method allows us to reduce annotations to only one point for each object in a video frame, yet retaining high quality mask predictions close to full supervision. We conduct comprehensive experiments on three datasets and achieve competitive performance compared with the fully-supervised methods.

Limitations. While PointVIS shows promising results in the direction of weakly-supervised video instance segmentation, it has some limitations. Typical failure cases are shown in Figure 3.4. We observe missing instances and temporal inconsistency. One hypothesis is that our performance is bounded by the quality of proposals, and we believe stronger video instance architecture can mitigate this gap. Another potential direction is to utilize video correspondences for label propagation and other denoising techniques for higher-quality proposals. We leave these as future work.

Broader Impacts. We open up the possibility of learning video instance segmentation with point-level supervision. We hope that our framework can be used to largely reduce annotation cost for a wide range of video recognition tasks in computer vision, including video object detection, video object segmentation, and video panoptic segmentation.

Chapter 4: Learning Dense Point Alignment for Video Action Recognition

4.1 Introduction

At the core of video understanding lies the fundamental synergy between motion and appearance cues. Motion patterns reveal the dynamic flow of actions through time. At the same time, appearance information captures the rich context, the interplay of objects, environments, and their relationships within each frame. In action recognition tasks, particularly in few-shot settings, explicitly modeling this complementary relationship becomes crucial, as both aspects provide distinct yet essential signals.

While appearance understanding in models has made great strides, the challenge of capturing crucial motion patterns remains complex. Traditional optical flow techniques [184–186] have been a primary approach, but they are fundamentally limited to analyzing adjacent frames and deteriorate under occlusions, resulting in incomplete motion representations. In parallel to optical flow methods, trajectory-based approaches emerged as an alternative paradigm. Early trajectory works [187–189] made progress by capturing longer-term patterns, and recent work [6] has further advanced this direction through point tracking. Unlike optical flow, point tracking [12, 190] explicitly maintains temporal correspondence across long sequences and handles occlusions naturally, making it particularly effective for capturing complex motion patterns in real-world actions. In this work, we aim to advance few-shot action recognition by building upon these advantages

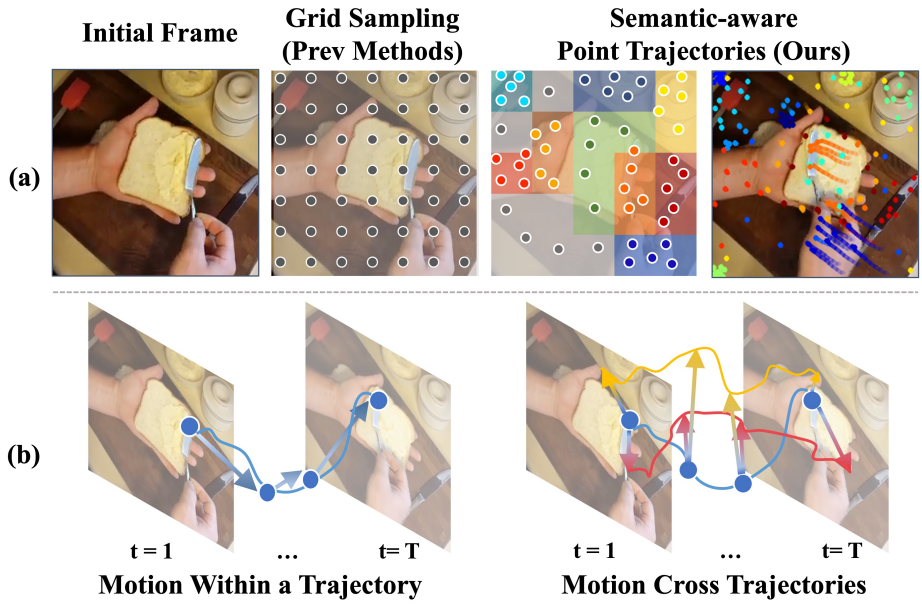


Figure 4.1: **Our motivation.** (a) Our semantic-aware points adapts better to object scale and semantic relevance while existing methods w/ grid sampling missed small objects w/ important motion (e.g., knife). (b) We explicitly model relational motions within a trajectory and across trajectories.

of point tracking approaches.

To advance point tracking for action recognition, we must address two fundamental challenges: (1) sampling informative query points to track through time, and (2) effectively modeling the complex motion patterns captured by these trajectories. Dense point sampling provides comprehensive coverage but is computationally expensive, while sparse sampling risks missing crucial motion information, especially from smaller objects essential for action understanding. Beyond these tracking challenges, current transformer-based approaches [191–193] attempt to learn motion patterns implicitly for action recognition, but it remains unclear whether these models truly capture and leverage motion information or rely on other contextual cues.

To address these limitations, we identify two key questions: (1) how can we develop an effective sampling strategy that balances coverage and efficiency? and (2) how can we explicitly

model and utilize the motion patterns captured in point trajectories? In this paper, we present a novel approach that leverages both semantic-aware sampling and explicit motion modeling to improve point tracking-based video understanding.

For the first challenge of effective point sampling, we propose a semantic-aware sampling strategy that adapts to object scale and importance. This approach is particularly crucial for actions involving small but critical objects, as illustrated in Figure 1(a), where the knife spreading the butter could be easily missed by uniform sampling due to its small size. By leveraging semantic information extracted from DINO [194] patch tokens, our method ensures comprehensive coverage of action-relevant objects regardless of scale. The sampling density, guided by semantic understanding, allocates denser points to smaller, action-critical objects (*e.g.,* knife) and sparser sampling to larger regions (*e.g.,* background desk). This approach ensures we capture the motion of all semantically meaningful objects while maintaining computational efficiency.

With our sampling strategy in place, we address the second challenge through our Relational Motion Modeling module, which processes point trajectories from modern trackers [12, 190] in two complementary ways. Revisiting the knife spreading butter example in Figure 4.1, our first component captures intra-trajectory dynamics using Histogram of Oriented Displacements (HoD) [195] to model the knife's own movement patterns and directions. Our second component extracts inter-trajectory relationships by tracking how different objects (*e.g.,* the knife and bread) interact with each other, revealing the distinctive motion patterns that define actions. This Relational Motion Modeling module effectively captures both individual object movements and their meaningful interactions (Figure 1(b)).

Overall, we introduce Trokens, a novel framework that leverages semantic-aware sampling and explicit motion modeling to effectively bridge motion and appearance information. Our

contributions are as follows:

- We propose a novel semantic-aware point sampling approach leveraging semantic priors to adaptively distribute tracking points based on object scale and relevance.
- We develop a novel Relational Motion Modeling module that explicitly captures both intra- and inter-trajectory dynamics to understand complex motion patterns.
- We conduct comprehensive experiments with Trokens on six few-shot action recognition benchmarks including Something-Something (Full & Small) [196], Kinetics [197], UCF101 [198], HMDB51 [199], and FineGym [200], and achieve state-of-the-art performance.

4.2 Related Work

Few-Shot Action Recognition. Human Action Recognition requires one to model the complex temporal dynamics of a scene while also filtering out the redundant information shared between frames [201–204]. While these challenges are typically addressed using large training datasets, in the setting of few-shot action recognition, methods must instead use well-constructed mechanisms to achieve effective performance with limited data. Many methods are based on metric-learning [205–216], introducing various mechanisms to determine if two videos are similar or different, thus enabling action classification. Meanwhile, other methods focus on improving feature representations for spatio-temporal modeling [6, 13, 207, 217–222]. Some recent works also leverage multi-modal language-image pretraining [11] and/or additional text data at training or inference time to further enhance performance [223–231]. While these works show strong results, they represent a bifurcation of the field into two domains: multi-modal and vision-only

few-shot action recognition. Our method, Trokens, is a vision-only method, and we focus on primarily comparing with like baselines. A recent work [232] proposes a state space based architecture for long sequence few shot action recognition. While promising, improving architecture is orthogonal to our contributions.

Point Tracking for Feature Learning. Point tracking has a long history in computer vision research, and now recent advances in the field have enabled the efficient generation of dense and high-quality point tracks [12, 233–239]. These tracks lend themselves well to a fundamental element of video learning: the disentanglement of motion and appearance information. Several prior works have successfully applied point tracks to guide the extraction of deep and classical features [187–189]. The recent work TATs [6] further demonstrates the power of point tracking for transformer token pooling in few-shot action recognition. However, TATs falls short on our two key challenges: its uniform grid-based sampling fails to adapt to object scales, and it treats trajectories merely as feature anchors, neglecting the rich motion patterns they contain. These limitations motivate our Trokens approach that addresses both sampling and motion modeling challenges.

Motion Features. In recent years, many architectures and approaches have been proposed to learn joint or disentangled appearance and motion features from video [240–244]. However, such approaches are reliant on training data and struggle in low-data few-shot regimes. Meanwhile, other methods have been proposed to model motion features directly from optical flow [184, 185], or point trajectories [188, 245, 246]. In this work, we aim to improve few-shot action recognition performance by efficiently leveraging motion features derived from our trajectories. We draw inspiration from classical vision methods like Histogram of Oriented Gradients (HoG) [247], and

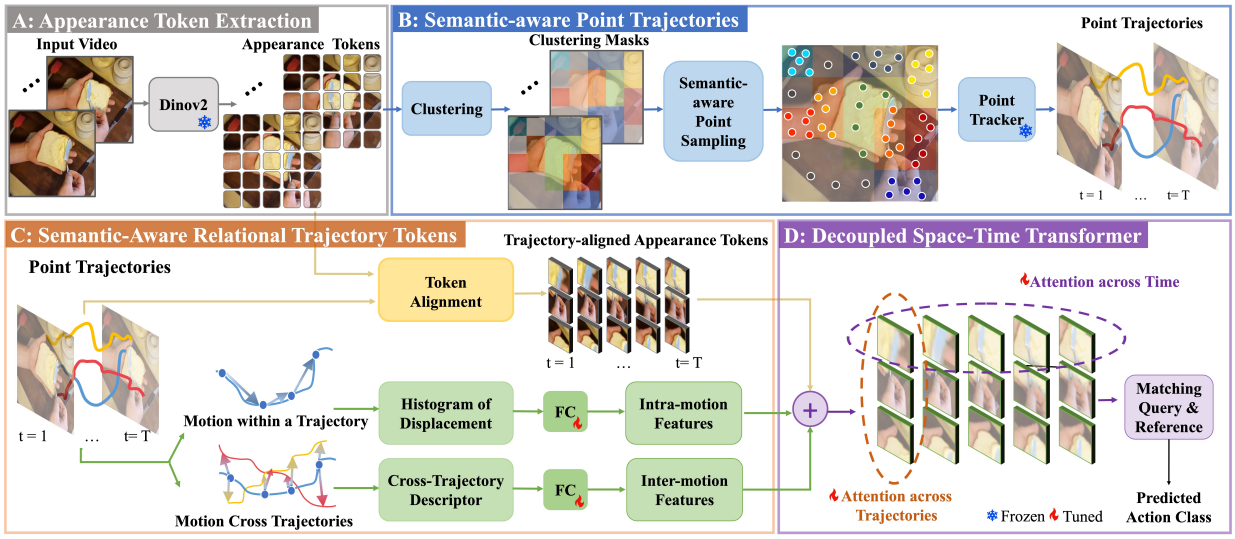


Figure 4.2: **Method Overview.** (A) Given an input video, we extract appearance tokens using DINOv2. (B) We then cluster these tokens and sample semantic-aware points in the initial frame, which are tracked using Co-tracker [12] to obtain point trajectories. (C) We compute intra- and inter-motion features, reorder appearance tokens via token alignment [6], and fuse them with motion features via element-wise addition to form semantic-aware relational trajectory tokens. (D) Finally, we input these tokens into a Decoupled Space-Time Transformer for few-shot action classification.

Histogram of Oriented Displacements (HoD) [195]. Specifically, we present a new implementation of HoD tailored to toward general object intra-track motion features, which we describe in Section 4.4.3. While the original HoD is focused on only human skeleton keypoints, our version is designed for general object motion characterization. Additionally, we preserve temporal order through per-timestep computation rather than whole-trajectory pyramidal aggregation like [195].

4.3 Few-shot Action Recognition Setup

Few-shot action recognition is the problem of recognizing novel action classes with few labeled instances per class. Unlike fully supervised learning, training and test classes are mutually exclusive in the few-shot setting. Formally, given a training set $D_{\text{train}} = \{(v_i, y_i) \mid y_i \in C_{\text{train}}\}$ and test set $D_{\text{test}} = \{(v_i, y_i) \mid y_i \in C_{\text{test}}\}$, $C_{\text{train}} \cap C_{\text{test}} = \phi$. Training is done with episode-based

meta-learning where each episode has a support set S with N classes and K examples per class (termed N-way K-shot, e.g., 5-way 1-shot), and a query set Q with samples to classify into these N classes.

4.4 Our Approach

4.4.1 Overview

Our work aims to leverage semantic and motion priors for few-shot video action classification. We propose the following key components: (1) Semantic-aware point trajectories sampling, where we leverage DINO features to sample semantically meaningful points for motion tracking (Sec 4.4.2); and (2) a Relational Motion Modeling module that explicitly models motion changes within individual trajectories and across trajectories to capture detailed movement patterns (Sec 4.4.3). The appearance features and the proposed motion features are sampled using the semantic-aware point trajectories and subsequently fused to form the trajectory-aligned tokens following [6]. Next, we adopt a Decoupled Space-Time Transformer [6], to process the trajectory-aligned tokens. All components of our method are trained in an end-to-end fashion using a standard few-shot loss [13].

4.4.2 Semantic-aware Point Trajectories

Dense point tracking [12, 248] offers promising video understanding capabilities, but its effectiveness depends on the initial selection of tracking points. The standard practice for point-based few-shot action classification is to sample points in a uniform grid [6]. Employing uniform grid sampling, despite its simplicity, often under-samples small objects crucial for understanding

actions while capturing redundant information from background regions.

To address this limitation, we leverage a semantic prior to guide point selection. DINO's self-supervised learning framework produces patch tokens with rich semantic information, where tokens from the same object naturally cluster in feature space [177, 249–252]. Leveraging this property, we construct semantic-aware clustering masks over patch tokens and sample points accordingly. Our strategy enables semantic-aware point sampling that adapts to object scale and semantic relevance.

Formally, we extract DINO appearance features $\mathcal{F}^{\text{RGB}} \in \mathcal{R}^{H \times W \times T \times C}$ from the input video, where T is the number of frames and H, W are spatial dimensions. We cluster the appearance features [253] into L groups which are subsequently used to sample semantic-aware points. For M trajectories, we sample $q = \frac{M}{L}$ points per cluster, from the first frame, as the semantic-aware points. We denote these points for all clusters as $P_s = \{(x_s^i, y_s^i)\}_{i=1}^{M}$, where (x_i, y_i) is the spatial coordinate of a point. These points serve as initialization for trajectory extraction.

To track the sampled points, we utilize pretrained dense point tracking model Co-tracker [12] (denoted as $\mathcal{T}$). The extracted point trajectories, known as semantic-aware point trajectories, are given by $\mathcal{P} = \mathcal{T}(P_s) = \{\mathcal{P}^m\}_{m=1}^{M}$. Each trajectory $\mathcal{P}^m = [(x_t^m, y_t^m)]_{t=1}^{T} \in \mathcal{R}^{T \times 2}$, captures the motion of a point over T frames. As shown in Fig. 4.1, our approach provides better coverage of small but significant objects while reducing redundant background trajectories compared to uniform sampling.

4.4.3 Relational Motion Modeling

Given semantic-aware point trajectories, a natural question is how to best capture their rich motion dynamics. To this end, we propose to explicitly model motion in two key aspects: dynamics within individual trajectories (intra-motion) and relationships across different trajectories (inter-motion). Such fine-grained representations capture both local motion patterns and cross trajectory interactions, providing discriminative features crucial for action recognition.

Intra-motion Module. Inspired by the Histogram of Oriented Gradients (HoG) [247], we revisit *Histogram of Displacement (HoD)* [254] to encode both magnitude and orientation changes over time on top of point trajectories. Given a trajectory $\mathcal{P}^m$, we compute the displacement at time t as $\Delta x_t = (x_t - x_{t-\delta})$ and $\Delta y_t = (y_t - y_{t-\delta})$, where δ is a hyperparameter controlling the temporal interval (we omit m for simplicity). The displacement magnitude is $\Delta d_t = \sqrt{\Delta x_t^2 + \Delta y_t^2}$, and the direction is $\theta_t = \arctan 2(\Delta y_t, \Delta x_t)$, with zero padding for $t < \delta$. For each timestep, we bin the orientation θ_t into a histogram with B bins spanning 360 degrees (*e.g.,* $B = 32$). Each displacement Δd_t contributes to the two nearest orientation bins proportionally, weighted by its magnitude. This produces a histogram of displacement descriptor for each trajectory:

$$\mathbf{H}_{\text{HoD}} = f_{\text{HoD}}(\mathcal{P}^m) \in \mathbb{R}^{T \times B}$$

We then apply a fully connected layer to project this descriptor into a C-dimensional space for all M trajectories to obtain our intra-motion features $\mathcal{F}_{\text{intra}}^{\text{motion}}$ as below:

$$\mathcal{F}_{\text{intra}}^{\text{motion}} = \text{FC}(f_{\text{HoD}}(\mathcal{P})) \in \mathbb{R}^{M \times T \times C}.$$

Our approach differs from prior work [254] in encoding HoD in several aspects. First, we preserve temporal order through per-timestep computation rather than whole-trajectory aggregation. Second, we enhance expressiveness via learnable projections. Moreover, our formulation generalize beyond human keypoints to arbitrary trajectories for broader applicability to motion analysis tasks.

Inter-motion Module. While intra-motion features capture dynamics within individual trajectories, complex actions involve coordinated movements (e.g., relative hand positions distinguish "opening a door" from "chopping vegetables"). We complement our intra-motion features with inter-motion modeling that captures evolving spatial relationships among trajectories. Specifically, we compute pairwise relative displacements between trajectories. For each trajectory $\mathcal{P}^m = [(x_t^m, y_t^m)]_{t=1}^T$ at time t, we define its cross-trajectory descriptor as:

$$\mathbf{d}_t^m = \left[(x_t^m - x_t^{m'}, y_t^m - y_t^{m'})\right]_{m'=1}^M \in \mathbb{R}^{2M}$$

This captures relative positions between trajectory m and all other trajectories in a fixed order. The complete cross-trajectory descriptor $\mathbf{d} \in \mathbb{R}^{M \times T \times 2M}$ represents spatial relationships across all trajectories and timesteps. Finally, we obtain inter-motion features by projecting the cross-trajectory descriptors to the feature space:

$$\mathcal{F}_{\text{inter}}^{\text{motion}} = \text{FC}(\mathbf{d}) \in \mathbb{R}^{M \times T \times C},$$

It is worth noting that while transformers could potentially learn motion patterns through self-attention, their position embeddings primarily encode static locations without directly capturing

temporal displacements or cross-trajectory relationships. Our explicit modeling of motion dynamics provides prior knowledge that helps the model focus on discriminative motion features rather than relying on self-attention to implicitly discover these patterns.

4.4.4 Motion-aware Space-Time Transformer

Given point trajectories $\mathcal{P}$, intra, inter motion features $\mathcal{F}^{\text{motion}}_{\text{intra}}$, $\mathcal{F}^{\text{motion}}_{\text{inter}}$, and appearance tokens from DINO $\mathcal{F}^{\text{RGB}}$, we utilize a transformer for spatiotemporal modeling of both motion and appearance. We first construct trajectory-aligned appearance tokens from point trajectories and appearance features following [6]. Given video appearance tokens $\mathcal{F}^{\text{RGB}} \in \mathbb{R}^{H\times W\times T\times C}$ and point trajectories $\mathcal{P} \in \mathbb{R}^{M\times T\times 2}$, we extract trajectory-aligned appearance tokens:

$$\mathcal{F}^{\text{RGB}}_{\text{traj}} = \text{Align}(\mathcal{F}^{\text{RGB}}, \mathcal{P}) \in \mathbb{R}^{M\times T\times C}, \tag{4.1}$$

where $\text{Align}(\cdot)$ samples appearance features given point coordinates. This reordering aligns visual information with motion paths, helping self-attention learn motion explicitly. We then fuse our intra-inter motion features and trajectory-aligned appearance tokens via element-wise addition, enabling the transformer to capture both intra- and inter-trajectory dependencies:

$$\mathcal{F}^{\text{fuse}} = \mathcal{F}^{\text{RGB}}_{\text{traj}} + \mathcal{F}^{\text{motion}}_{\text{intra}} + \mathcal{F}^{\text{motion}}_{\text{inter}}. \tag{4.2}$$

Given the semantic-aware relational trajectory tokens $\mathcal{F}^{\text{fuse}}$, we employ a decoupled attention strategy that processes the temporal and spatial dimensions separately following [6]. Self-attention is applied within each trajectory to model temporal dependencies, and across trajectories to cap-

ture spatial relationships in parallel, the results of which are then added together to form the final output embeddings $\mathcal{F}_{\text{final}}$.

Finally, we conduct cross-attention between a learnable token and the final output embeddings $\mathcal{F}_{\text{final}} \in \mathbb{R}^{M \times T \times C}$, and produced the output class token $\mathbf{c}_{\text{cls}} \in \mathcal{R}^{C}$.

4.4.5 Few-shot Loss

We extract the output embeddings $\mathcal{F}_{\text{final}}$ and class token $\mathbf{c}_{\text{cls}}$ for all samples in the support and query sets. To obtain the classification output, we apply a fully connected layer on the output class token, mapping it to the number of classes C_{train} in the training set with $p_{\text{cls}} = \text{FC}(\mathbf{c}_{\text{cls}}) \in \mathbb{R}^{C_{\text{train}}}$.

Following prior works [6,13,207,208], our loss consists of two parts. One is a cross-entropy loss applied to the classification output on the query set p^{Q}_{cls}, capturing global C_{train} class information. The other is a contrastive loss [13] applied to final embeddings between the query set $\mathcal{F}^{\text{Q}}_{\text{final}}$ and the reference set $\mathcal{F}^{\text{S}}_{\text{final}}$ for N-way few-shot classification to encourage feature discrimination:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(p^{Q}_{\text{cls}}, y) + \mathcal{L}_{\text{Contrastive}}(\mathcal{F}^{Q}_{\text{final}}, \mathcal{F}^{S}_{\text{final}}). \tag{4.3}$$

We refer readers to [6] and [13] for further details.

4.5 Experiments

4.5.1 Datasets

We evaluate our approach's effectiveness across multiple action recognition benchmarks using established few-shot splits: Something-Something [196], Kinetics [197], UCF101 [198], HMDB51 [199], and FineGym [200]. For Something-Something, we use two standard configurations [205]: SSV2 Small (100 samples per class; 100 classes) and SSV2 Full (all classes). Our evaluations follow the split protocols from previous works: Kinetics splits from [220], UCF101 and HMDB51 splits from [212, 219], and FineGym splits from [6]. To ensure fair comparison, we maintain consistency with the evaluation protocols used in prior works [6, 13, 207, 208, 222].

4.5.2 Implementation details

We follow prior work [6, 13] for most architectural choices and training configurations. Using DINOv2-base [194], we get semantic clusters from which 256 semantic-aware points are sampled for tracking via CoTracker [12, 248]. The architecture employs a single transformer block and uses 32-bin Histogram of Directions (HoD) in the intra-motion module. During training, only the transformer and motion modules are optimized while other components remain frozen. Following standard protocols [6, 13, 207, 222], we evaluate using average few-shot accuracy across 10,000 episodes.

Table 4.1: Comparison of few-shot action accuracy (1-5 shots) on SSV2 Full and Kinetics datasets versus contemporary methods. Best results are bolded, second-best underlined, and "-" indicates unavailable data.

		SSV2 Full					Kinetics				
Method	**Reference**	1-shot	2-shot	3-shot	4-shot	5-shot	1-shot	2-shot	3-shot	4-shot	5-shot
OTAM [205]	CVPR'20	42.8	49.1	51.5	52.0	52.3	72.2	75.9	78.7	81.9	84.2
TRX [206]	CVPR'21	42.0	53.1	57.6	61.1	64.6	63.6	76.2	81.8	83.4	85.2
STRM [217]	CVPR'22	43.1	53.3	59.1	61.7	68.1	62.9	76.4	81.1	83.8	86.7
MTFAN [219]	CVPR'22	45.7	-	-	-	60.4	74.6	-	-	-	87.4
HYRSM [207]	CVPR'22	54.3	62.2	65.1	67.9	69.0	73.7	80.0	83.5	84.6	86.1
HCL [209]	ECCV'22	47.3	54.5	59.0	62.4	64.9	73.7	79.1	82.4	84.0	85.8
Nguyen et al [255]	ECCV'22	43.8	-	-	-	61.1	74.3	-	-	-	87.4
Huang et al [256]	ECCV'22	49.3	-	-	-	66.7	73.3	-	-	-	86.4
MoLo [13]	CVPR'23	56.6	62.3	67.0	68.5	70.6	74.0	80.4	83.7	84.7	85.6
SloshNet [218]	AAA1'23	46.5	-	-	-	68.3	70.4	-	-	-	87.0
GgHM [222]	ICCV'23	54.5	-	-	-	69.2	74.9	-	-	-	87.4
RFPL [257]	ICCV'23	47.0	54.6	58.3	60.3	61.0	74.6	80.0	82.1	84.1	86.8
CCLN [258]	PAMI'24	46.0	-	-	-	61.3	75.8	82.1	85.0	86.1	87.5
HYRSM++ [208]	PR'24	55.0	63.5	66.0	68.8	69.8	74.0	80.8	83.9	85.3	86.4
TATS [6]	ECCV'24	<u>57.7</u>	<u>67.1</u>	<u>70.0</u>	<u>70.6</u>	<u>74.6</u>	<u>81.9</u>	<u>86.5</u>	**89.9**	<u>90.6</u>	<u>91.1</u>
PointVIS	-	**61.5**	**69.9**	**73.8**	**75.9**	**76.7**	**82.9**	**87.7**	**89.9**	**90.8**	**91.2**

4.5.3 Quantitative Results

We evaluate Trokens against previous state-of-the-art approaches under the standard 5-way K-shot setting. Tables 4.1, 4.2, and 4.5 present our results on SSV2 Full and Kinetics (K=1-5), SSV2 Small/UCF-101/HMDB-51 (K=1,3,5), and FineGym respectively. On SSV2 Full, Trokens consistently outperforms TATs [6] with gains of 3.8%, 2.8%, 3.2%, 5.3%, and 2.1% across 1-5 shots respectively. For Kinetics, we achieve improvements of 1.0% and 1.2% in 1-shot and 2-shot settings, with comparable performance in higher shots. The modest gains reflect Kinetics' inherent appearance bias, where actions are primarily distinguishable through static cues, reducing the effectiveness of our motion-focused contributions. SSV2 Small demonstrates significant improvements with gains of 3.5%, 5.3%, and 4.5% across 1,3,5-shot settings. UCF-101 shows consistent improvements of 2.0%, 0.5%, and 2.4%, while HMDB-51 exhibits substantial gains of 9.8%, 8.2%, and 5.3% across respective shots. FineGym follows this trend with improvements

of 2.6%, 2.3%, and 2.0% for 1,3,5-shot settings. Furthermore, when varying N-way settings (Table 4.3), Trokens maintains its superior performance with consistent gains of 3-4% on SSV2 Full and 1-2% on Kinetics in the 1-shot setting. These substantial improvements across diverse datasets, shot settings, and N-way configurations demonstrate our approach's robust and superior nature.

Class-wise performance anlaysis. Figure 4.3 presents a class-wise performance comparison between our method and previous approaches [6, 13] on both SSV2 Small and SSV2 Full splits. Our method demonstrates consistent improvements across classes through the combined benefits of semantic-aware sampling and motion modules. The semantic sampling strategy ensures better tracking of smaller objects, while our motion modules capture complex temporal dynamics effectively. On SSV2 Small, classes like 'Unfolding something' and 'Twisting something' show notable improvements, particularly benefiting from our motion-aware architecture. Similarly, SSV2 Full exhibits enhanced performance in classes such as 'Pulling something from left to right' and 'Dropping something next to something'. However, our analysis also reveals limitations in handling rapid motions causing blur (e.g., 'Rolling something on flat surface') and significant camera movements (e.g., 'Picking something up'), where point tracking becomes challenging.

4.5.4 Ablation Analysis

We conduct an ablation study to evaluate key design decisions. Through systematic experimentation, we analyze the impact of each component on model performance and provide empirical justification for our final configuration. We also show additional ablations in our supplementary.

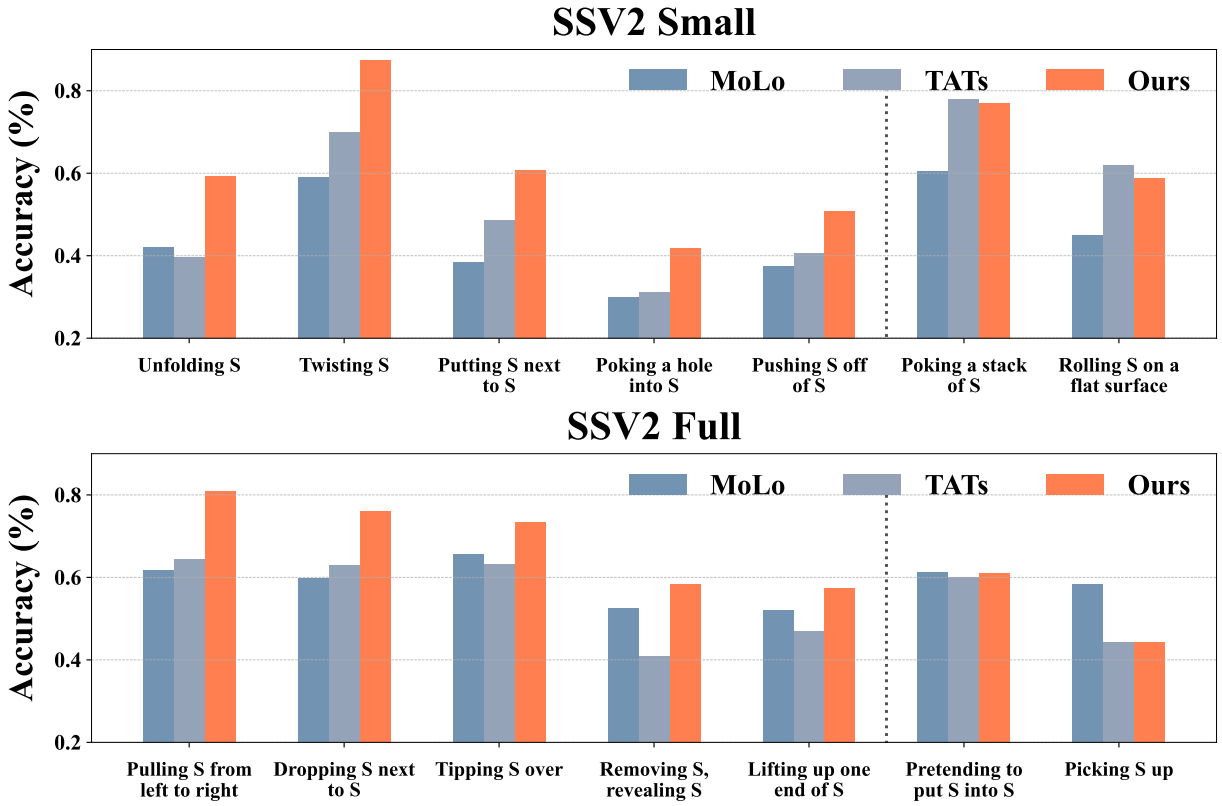


Figure 4.3: Class-wise accuracy comparison between MoLo [13], TATs [6] and our method. Left: classes where our approach shows performance gains. Right: classes without improvement.

Table 4.2: Comparison of few-shot action accuracy (1,3 and 5 shots) on SSV2 Small, UCF-101 and HMDB-51 datasets versus contemporary methods. Best results are bolded, second-best underlined, and "-" indicates unavailable data.

		SSV2 Small			UCF-101			HMDB-51		
Method	**Reference**	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
OTAM [205]	CVPR'20	36.4	45.9	48.0	79.9	87.0	88.9	54.5	65.7	68.0
TRX [206]	CVPR'21	36.0	51.9	56.7	78.2	92.4	96.1	53.1	66.8	75.6
STRM [217]	CVPR'22	37.1	49.2	55.3	80.5	92.7	<u>96.9</u>	52.3	67.4	77.3
MTFAN [219]	CVPR'22	-	-	-	84.8	-	95.1	-	-	-
HYRSM [207]	CVPR'22	40.6	52.3	56.1	83.9	93.0	94.7	60.3	71.7	76.0
HCL [209]	ECCV'22	38.7	49.1	55.4	82.5	91.0	93.9	59.1	71.2	76.3
Nguyen et al [255]	ECCV'22	-	-	-	-	-	-	59.6	-	76.9
Huang et al [256]	ECCV'22	38.9	-	61.6	71.4	-	91.0	60.1	-	77.0
MoLo [13]	CVPR'23	42.7	52.9	56.4	86.0	93.5	95.5	60.8	72.0	77.4
GgHM [222]	ICCV'23	-	-	-	85.2	-	96.3	61.2	-	76.9
RFPL [257]	ICCV'23	-	-	-	82.5	94.1	96.3	-	-	-
CCLN [258]	PAMI'24	-	-	-	86.9	94.2	96.1	<u>65.1</u>	<u>76.2</u>	<u>78.8</u>
HYRSM++ [208]	PR'24	42.8	52.4	58.0	85.8	93.5	95.9	61.5	72.7	76.4
TATs [6]	ECCV'24	<u>47.9</u>	<u>60.0</u>	<u>64.4</u>	<u>92.0</u>	<u>96.8</u>	95.5	60.0	71.8	77.0
PointVIS	-	**53.4**	**65.3**	**68.9**	**94.0**	**97.3**	**97.9**	**69.8**	**80.0**	**82.3**

Table 4.3: Comparative N-way 1-shot classification accuracy (N=5-10) on Kinetics and SSV2 Full datasets versus contemporary methods. Best and second-best results are bolded and underlined, respectively.

	SSV2 Full						Kinetics					
Method	5-way	6-way	7-way	8-way	9-way	10-way	5-way	6-way	7-way	8-way	9-way	10-way
OTAM [205]	42.8	38.6	35.1	32.3	30.0	28.2	72.2	68.7	66.0	63.0	61.9	59.0
TRX [206]	42.0	41.5	36.1	33.6	32.0	30.3	63.6	59.4	56.7	54.6	53.2	51.1
HyRSM [207]	54.3	50.1	45.8	44.3	42.1	40.0	73.7	69.5	66.6	65.5	63.4	61.0
MoLo [13]	56.6	51.6	48.1	44.8	42.5	40.3	74.0	69.7	67.4	65.8	63.5	61.3
TATs [6]	<u>57.7</u>	<u>55.7</u>	<u>52.5</u>	<u>50.0</u>	<u>47.0</u>	<u>45.8</u>	<u>81.9</u>	<u>79.0</u>	<u>76.1</u>	<u>75.2</u>	<u>72.2</u>	<u>72.0</u>
PointVIS	**61.5**	**59.1**	**56.5**	**54.6**	**51.4**	**49.1**	**82.9**	**80.2**	**78.5**	**76.8**	**75.5**	**73.3**

Table 4.4: Impact of each component on Trokens, demonstrating the relative contribution of individual elements with the final row representing our final setting.

Semantic-aware sampling	**Intra motion**	**Inter motion**	**SSV2 Small**		**SSV2 Full**	
			1-shot	5-shot	1-shot	5-shot
✗	✗	✗	47.9	64.4	57.7	74.6
✓	✗	✗	49.9	65.7	58.6	74.2
✓	✓	✗	51.8	67.7	61.3	75.7
✓	✗	✓	52.2	67.3	60.6	74.8
✓	✓	✓	**53.4**	**68.9**	**61.5**	**76.7**

Impact of each component. Table 4.4 presents an ablation study of our key components. Starting from the baseline configuration [6], we first evaluate our semantic-aware point sampling strategy, which yields improvements of 2% and 1.3% on SSV2 Small (1-shot and 5-shot), and 0.9% on SSV2 Full (1-shot). Given these gains, we retain this sampling strategy for subsequent experiments. We then analyze our motion modules independently. The intra-motion module alone achieves gains of $\sim$2% on both shots of SSV2 Small and up to 3.3% on SSV2 Full. Similarly, the inter-motion module independently shows improvements of 1.3–1.6% on SSV2 Small and up to 2% on SSV2 Full. When combined, these modules yield consistent improvements of 1–2% across all settings, suggesting they capture complementary motion information that enhances

Table 4.5: Comparison of few-shot action accuracy (1,3 and 5 shots) on the FineGym dataset.

	FineGym		
Method	1-shot	3-shot	5-shot
MoLo [13]	73.3	80.2	84.8
TATs [6]	81.8	86.0	87.9
PointVIS	**84.4**	**88.3**	**89.9**

Table 4.6: Comparative analysis of intra-motion module variants independent of inter-motion module.

	SSV2 Full		
Intra-motion module	1-shot	3-shot	5-shot
Displacement only	59.9	71.6	74.2
HoD (ours)	**61.3**	**73.6**	**75.7**

Table 4.7: Trainable parameter analysis. * represents our implementation of [6] with increased parameters for fair comparison.

		SSV2 Small		SSV2 Full	
Method	**Params**	1-shot	5-shot	1-shot	5-shot
TATs [6]	11.8 M	47.9	64.4	57.7	74.6
TATs* [6]	18.1 M	48.0	63.0	59.6	74.3
PointVIS	17.4 M	**53.4**	**68.9**	**61.5**	**76.7**

performance.

Analysis of Intra-motion module. Table 4.6 compares our choice of Histogram of Oriented Displacement (HoD) features against displacement-only features for intra-motion representation. To isolate the impact of feature choice, we conduct all experiments with the inter-motion module disabled. In this controlled setting, HoD features consistently outperform displacement-only features on SSV2 Full, yielding gains of 2.4%, 2.0%, and 1.5% for 1-shot, 3-shot, and 5-shot settings, respectively. These results demonstrate that incorporating directional information through HoD features captures richer trajectory patterns compared to using displacement information alone.

Varying number of points. To evaluate the effectiveness of our semantic sampling strategy, we analyze performance across different point counts in the 1-shot setting on both SSV2 Small and SSV2 Full splits. Fig 4.4 compares our method against TATs [6] using equivalent point counts. Notably, our approach with just 32 points outperforms TATs using 256 points, demonstrating that our semantic-aware sampling yields more informative points for the task. This highlights the quality of our sampling strategy in capturing motion information with significantly fewer points.

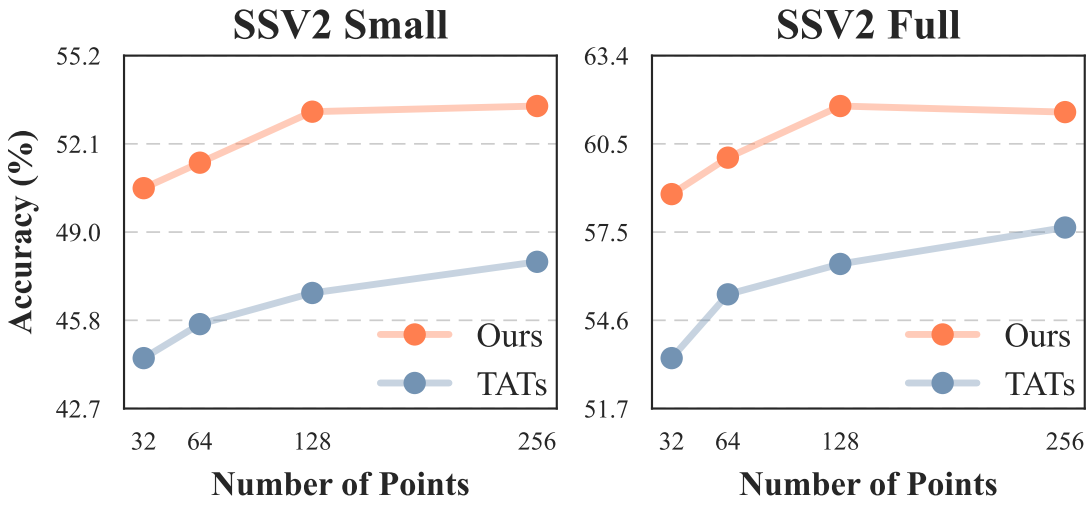


Figure 4.4: Few-shot accuracy by variyng the number of points on SSV2 Small and SSV2 Full for 1-shot setting.

Effect of trainable parameters. To address potential concerns about performance gains stemming from increased model capacity, we implemented a modified version of [6] with expanded parameters to match our model size. We maintaining the rest of their original configuration. Table 4.7 shows that this parameter-matched re-implementation of [6] does not yield improved performance. This demonstrates that our performance gains stem from the effectiveness of our motion modules in capturing complementary information, rather than from increased model capacity.

4.5.5 Qualitative Analysis

Figure 4.5 demonstrates our semantic-aware point sampling through trajectory visualization. Our method concentrates tracking points on action-relevant objects for meaningful motion capture. The top-right quadrant showing ”Taking something out of something” tracks both a lemon and bottle, capturing the essential lifting motion. Trajectories across different examples of the same action show striking similarities while remaining distinct from other actions. This



Figure 4.5: Visualization of action trajectory similarities across four classes, where semantic-based sampling enables object-focused trajectories. Each quadrant demonstrates intra-class motion consistency while maintaining inter-class discriminative features.

intra-class similarity and inter-class variation shows how our semantic sampling enhances action recognition by focusing on meaningful motion patterns.

4.6 Discussion

While our approach demonstrates strong performance overall, it faces certain limitations. On datasets like Kinetics our motion-focused approach provides limited performance gains as its actions are distinguishable from appearance alone. Additionally, our method faces challenges inherent to point tracking: it becomes vulnerable to rapid motions that cause motion blur. Significant camera movement can disrupt point trajectory consistency, impacting performance in classes involving substantial viewpoint changes. These limitations primarily stem from the fundamental challenges in point tracking rather than our architectural choices, suggesting potential future directions in developing more robust tracking mechanisms. While we demonstrate the effective-

ness of our approach in few-shot action recognition, we hope Trokens inspires future research in both full classification settings and broader video understanding tasks, where our motion-aware architecture could prove beneficial.

In this paper, we introduced Trokens, a novel approach that advances few-shot action recognition through semantic-aware motion trajectory tokens. Our approach addresses two fundamental challenges in point tracking-based video understanding through semantic-aware sampling and a relational motion modeling framework that captures both intra-trajectory dynamics and inter-trajectory relationships. The effectiveness of our method is demonstrated through state-of-the-art performance across six challenging few-shot action recognition benchmarks. These results validate our core insight that combining semantic guidance with explicit motion modeling through trajectory tokens provides a robust foundation for understanding human actions in limited-data scenarios while opening promising directions for future research in dynamic scene understanding and broader video recognition tasks.

Chapter 5: Learning Visual Plan Alignment for Robot Control

5.1 Introduction

Paper List: [7, 8, 14, 259–265]

One critical objective of robotic learning is building a universal agent capable of performing a vast number of tasks across a diverse set of environments. Achieving this goal is challenging as the definition of a particular state or action may vary based on the task description. For instance, the state and action space of a robot tasked with navigating through a cluttered warehouse is differently defined than a robot whose purpose is to assemble intricate machinery. These variations demand a policy that not only provides a universal representation of the state space but also precisely identifies the actions necessary for any given task.

One promising direction is jointly using video and text descriptions to define a generalized state and action space. In these scenarios, a video generator is employed as a planner, which produces a sequential trajectory of frames as states given a text description of the immediate goal and an initial visual representation of the environment. Once the trajectory is generated, a policy conditioned on this trajectory (sequence of frames) is learned to infer the action taken between adjacent frames. The intuition is that using videos to represent the state space enables greater generalization across various tasks and environments. Recently, there has been considerable progress made in this field, with notable works [8, 259, 266] demonstrating success in tasks such

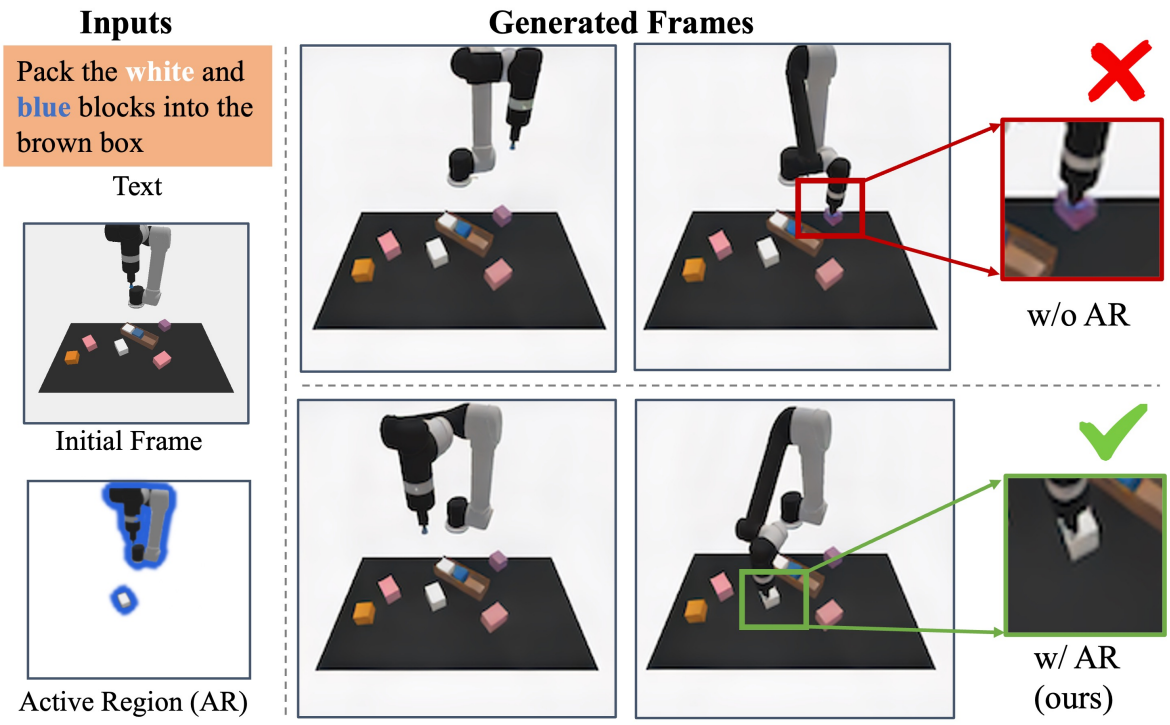


Figure 5.1: Given the task text and initial frame, we aim to generate a video as a planner. With active region conditioning (bottom left), our method ARDuP generates frames where the robot arm successfully picks up the white block (bottom), unlike the incorrect targeting of a purple block when w/o active region input (top), showing ARDuP's effectiveness in producing task-aligned video sequences.

as robot navigation and manipulation [147]. However, these methods often struggle to solve the task because they generate videos treating all pixels uniformly, often focusing on the wrong areas and neglecting to model pixels that are important for the policy. This can result in errors in generated frames, such as those seen on the top right of Figure 5.1. The robot picks up the wrong block (purple block) and can still reduce the loss in later frames by simply changing the color of the block to match the text description (e.g., changing purple to white).

To address this, our key insight is that only certain regions are important to solve the task; and these *active regions*, which represent areas where objects are being interacted with, contain critical context that helps to counteract the errors seen in prior work. By emphasizing active

regions during generation, we align the policy's attention to objects referred to by the given task description. As shown in Figure 5.1, our framework identifies the arm and the white box as active regions, which is highlighted in the image on the bottom left. This knowledge is then used to guide the video generation. As a result of this guidance, our approach successfully learns a policy to pick up the white block, while the baseline approach selects a purple block.

Formally, we propose Active Region Video Diffusion for Universal Policies (TREND), as shown in Figure 5.2. TREND consists of the following key components: 1) decomposed video generation via (a) a latent active region diffusion model to generate the active region proposals, and (b) a latent video diffusion model which conditions on the given active region context to generate future trajectory latent codes; and 2) an inverse dynamics model to decode the synthesized trajectory latent codes into an action sequence.

Since the ground truth active region is not available in the real-world scenario, we generate pseudo active regions from videos for training. Specifically, we employ Co-Tracker [263], a pre-trained dense point tracking model, to identify moving trajectories. These moving points are fed to SAM [264] to produce pseudo masks of active regions. This eliminates the need for manual annotations. In addition, we implement our video planner and inverse dynamics module in the latent space, which further reduces the computational cost.

We evaluate our method on public benchmarks, including the simulator CLIPort [7], and the real-world dataset Bridgev2 [14]. Specifically, we outperform the baseline by 21.3%, 17.2%, and 15.7% for the Place Bowl, Pack Object, and Pack Pair tasks, respectively. We also conduct a detailed ablation study to illustrate the benefits of our proposed modules. The main contributions of this work are as follows:

- We propose a novel formulation for video-based policy learning that focuses on the generation of active regions, enhancing the conditional policy's attention to interactive areas critical for task execution.
- We introduce TREND, a latent framework that integrates active region conditioning with latent diffusion models for video planning, and adopt latent representations for direct action decoding via inverse dynamics modeling. By utilizing motion cues in videos for automatic active region discovery, our method eliminates the need for manual annotations of active regions.
- We validate TREND's efficacy via extensive experiments on CLIPort and real-world dataset BridgeData v2, achieving notable improvements in success rates and generating convincingly realistic video plans.

5.2 Related Work

5.2.1 Video Diffusion Models for Decision Making

Recent advancements in diffusion models have led to their application across various decision-making problems [267, 268]. For instance, Diffuser [269] developed an unconditional diffusion model to generate trajectories comprising states and actions based on joints. As a pioneering approach, UniPi [8] casts the decision-making problem as a text-conditioned video generation problem and uses a video diffusion model as a universal policy. Extending UniPi, [260] leverages dense correspondence for action inference. And [259] incorporates VLMs with video diffusion models to facilitate the generation of long-horizon video plans. Existing works usually use the

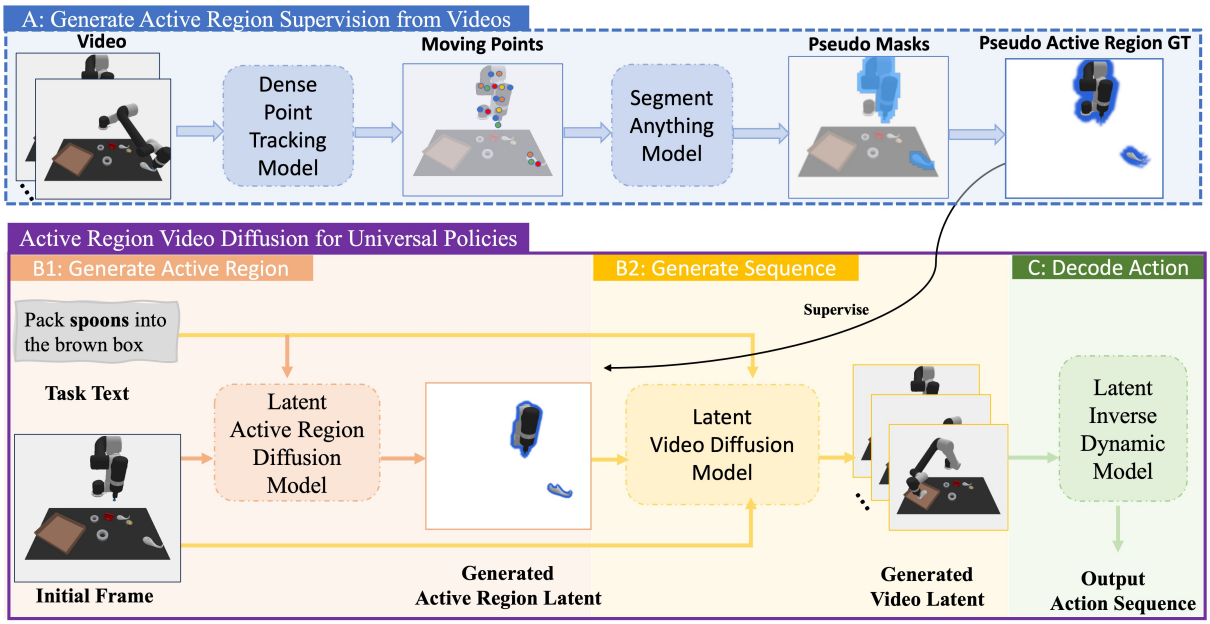


Figure 5.2: **Overview of our Active Region Video Diffusion for Universal Policies (ARDuP).** Starting with a video, we use Co-Tracker to identify moving points at the initial frame, which SAM then uses to generate pseudo masks of active regions. These pseudo masks delineate the pseudo active region, serving as supervision for training our Latent Active Region Diffusion Model. The generated latent active region informs the Latent Video Diffusion Model, resulting in a synthesized video latent sequence. Finally, the Latent Inverse Dynamic Model decodes the generated latent sequence into a corresponding action sequence.

video diffusion model to generate every pixel indifferently of the videos. In comparison, we propose to emphasize the generation of task-relevant objects, which leads to policy more aligned with the task requirements.

5.2.2 Active Region for Vision and Robotics

Active objects are objects being manipulated by the user and are important to the action being performed [262]. The idea that some regions are more important than others has been investigated in First Person Vision [270] and egocentric activity recognition [271,272]. In contrast, active regions for video-based policy are under-explored. We emphasize active regions in video generation and align the policy's attention to active regions based on the task text. Since the

ground truth active region is not available in real world, we generate pseudo-active regions from videos via dense point tracking and SAM to eliminate manual labeling [18,22,77,78,85,263,264].

5.3 Problem Formulation

5.3.1 Preliminaries

Markov Decision Processes (MDPs) [273] play a crucial role in finding a solution for sequential decision-making problems. However, scaling algorithms that solve MDPs to diverse environments is challenging. This is because there is no standardized way to define states of the MDP as well as the difficulty defining an appropriate reward structure and the reliance of MDPs on environment-specific dynamics models.

The Unified Predictive Decision Process (UPDP) is introduced in [8] and aims to solve the issues that arise with an MDP. They do this by (1) using video as their state space, (2) utilizing recent advances [11] in text-to-image understanding to use text to define the goal instead of an arbitrary reward and (3) developing a task-agnostic planning algorithm to find the action instead of relying on a predefined dynamics model. These three features enable UPDPs to scale across a wide variety of tasks in a way that MDPs cannot.

Formally, a UPDP is a tuple $\mathcal{G} = (\mathcal{X}, \mathcal{C}, H, \rho)$, where $\mathcal{X}$ is the observation space and each $x_0, x_1, \ldots, x_H \in \mathcal{X}$ is an RGB frame, $\mathcal{C}$ is the set of task descriptions, H is the task length and $\rho(\cdot|x_0, c)$ is a conditional video generator that synthesizes an H-step video $\{x_h\}_{h=1}^{H} \in \Delta(\mathcal{X}^H)$, where $x_1, x_2, \ldots, x_H$ are predicted future frames conditioned on the first ground truth frame x_0 and the task description c.

Given a UPDP $\mathcal{G}$, an action prediction algorithm is defined as $\mu(\cdot|\{x_h\}_{h=0}^{H}, c) \to \Delta(\mathcal{A}^H)$,

where $\mathcal{A}^H$ represents an H-step action. This algorithm outputs an action sequence that aligns with the provided trajectory $\{x_h\}_{h=1}^H$ in the UPDP $\mathcal{G}$ for the task c. This algorithm is trained offline assuming access to a dataset of existing experiences $\mathcal{D} = \{(x_i, a_i)_{i=0}^{H-1}, x_H, c\}$. Given $\mathcal{D}$, $\rho(\cdot|x_0, c)$ and $\pi(\cdot|\{x_h\}_{h=0}^H, c)$ can be estimated.

5.3.2 Latent Unified Predictive Decision Process conditioned on Active Region (LUPDP-AR)

In the UPDP framework, the success of a task heavily relies on the action prediction algorithm, $\mu(\cdot|\{x_h\}_{h=0}^H, c)$. The action prediction, in turn, is conditioned on the images generated in the video generation stage. However, not all pixels generated in the video have an equal impact on the action. Active regions, which are typically objects that are the most likely to be interacted with, are more likely to have an impact on the action. By prioritizing focus on generation of the active regions we can better align the predicted actions with the task description c. We introduce an enhanced version of UPDP, which we call LUPDP-AR (Latent Unified Predictive Decision Process conditioned on Active Region). LUPDP-AR introduces active region conditioning to the video generator to foster a more interaction-aware policy.

Formally, an LUPDP-AR is defined as a tuple $\hat{\mathcal{G}} = (\hat{\mathcal{X}}, \mathcal{C}, \hat{\mathcal{O}}, H, \phi)$, where $\hat{\mathcal{X}}$ and $\hat{\mathcal{O}}$ represent the latent spaces for RGB frames and active region frames, respectively. A frame encoder $\mathcal{E}(\cdot)$ is adopted to map both RGB frames and active regions into these latent spaces [265]. ϕ is a latent conditional generator that synthesizes an H-step latent trajectory $\{\hat{x}_h\}_{h=1}^H \in \Delta(\hat{\mathcal{X}}^H)$.

To ensure the accuracy of the active region in the generated trajectory we propose to condition the generator on the active region of the initial frame. Unlike the original UPDP, LUPDP-AR

conditions on the latent representation of the active region $\hat{o} \in \hat{\mathcal{O}}$ as well as the latent of the initial frame $\hat{x}_0 \in \hat{\mathcal{X}}$ and the task description c, instead of conditioning on just the original frame x_0 and the task description c. We define this new video generator as $\phi(\cdot|\hat{x}_0, c, \hat{o})$. Conditioning on active regions focuses the video generation process leading to more accurate actions for task completion.

To capture the active region in the initial frame, we define an active region generator as $\psi(\hat{o}|\hat{x}_0, c) : \hat{\mathcal{X}} \times \mathcal{C} \to \hat{\mathcal{O}}$, which generates the latent of the active region $\hat{o}$ based on the latent of the first frame $\hat{x}_0$ and the task description c. Intuitively, our methodology decomposes the challenging trajectory generation by generating the active region of the initial frame first, followed by the generation of the full sequence under the guidance of the active region.

Given an LUPDP-AR $\hat{\mathcal{G}}$, we define a latent conditioned action prediction algorithm $\pi(\cdot|\{\hat{x}_h\}_{h=0}^{H}, c) \to \Delta(\mathcal{A}^H)$ where $\mathcal{A}^H$ is an H-step action sequence. π only requires the latent of the generated images as input. This eliminates the need to generate RGB frames or decode a latent image, which is required in existing techniques such as [8, 259, 266]. We can still use a decoder, $\mathcal{R}$, to visually interpret the results, but it is not necessary for the action generation.

5.4 Method: Active Region Video Diffusion for Universal Policies (ARDuP)

In this section, we will describe the details of our approach, Active Region Video Diffusion for Universal Policies (ARDuP), a tailored instantiation of the conditional diffusion models within our proposed LUPDP-AR framework. First, OurName finds the active region (Sec 5.4.1). Next, OurName utilizes a video generator conditioned on the active region as well as the language instruction and the initial frame to predict the future frames for the episode (Sec 5.4.2).

Finally, given the generated video frames, OurName employs a latent inverse dynamic model to determine the actions the robot should take to complete the task (Sec 5.4.3).

5.4.1 Active Region Generator

5.4.1.1 Active Region Supervision from Videos

As discussed in 5.3.2, the key idea of our framework is to decompose the video generator $\rho(\cdot|x_0, c)$ into an active region generator $\psi(\hat{o}|\hat{x}_0, c)$ and a new video generator $\phi(\cdot|\hat{x}_0, c, \hat{o})$ conditioned on the active regions. To train our active region generator without manual labeling, we propose to construct an active region dataset $\mathcal{D}_{\mathcal{O}} = \{\hat{x}_0, c, \hat{o}\}$ from video demonstrations by a large pre-trained dense point tracking model to pinpoint areas of significant activity throughout the episode. By doing so, we can identify pseudo masks for active regions in each initial frame from $\mathcal{D}$. The process of obtaining the active region from videos is shown in Figure 5.2.

Formally, given a video $\mathbf{V} = \{x_h\}_{h=0}^{H}$, we first obtain dense point trajectories $\mathcal{P}$ through a dense point tracking model $\mathcal{F}_p$. While any dense point tracking model can be used we adopt Co-Tracker [263] pretrained on real-world videos due to its efficiency [263]. Co-Tracker divides the initial frame into an $M \times M$ grid and tracks each point across $H + 1$ frames. We denote the obtained dense point trajectories set as $\mathcal{P} = \mathcal{F}_p(\mathbf{V})$, where $|\mathcal{P}| = M \times M$ is the size of all dense point trajectories. Each point trajectory consists of a single point's location from timestep 0 to H.

Given all point trajectories $\mathcal{P}$, we find the moving point trajectories by analyzing the change in pixel location. For each point trajectory $\mathbf{p} \in \mathcal{P}$, we compute the absolute movement $\Delta\mathbf{p}_h$

between two adjacent frames at timestep h and $h-1$ as:

$$\Delta \mathbf{p}_h = ||\mathbf{p}_h - \mathbf{p}_{h-1}||_2, \quad \text{for } h = 1, \ldots, H \tag{5.1}$$

where $\mathbf{p}_h$ and $\mathbf{p}_{h-1}$ are the point coordinates at time h and $h-1$, respectively. The average movement $\Delta\bar{\mathbf{p}}$ of a point trajectory over H timesteps is given by:

$$\Delta\bar{\mathbf{p}} = \frac{1}{H}\sum_{h=1}^{H} \Delta \mathbf{p}_h \tag{5.2}$$

To identify moving point trajectories that exhibit significant displacement, we apply a selection criterion based on τ an average movement threshold, denoted by τ as follows:

$$\mathcal{P}_m = \{\mathbf{p} \in \mathcal{P} | \Delta\bar{\mathbf{p}} > \tau\} \tag{5.3}$$

where τ is a pre-selected threshold. $\mathcal{P}_m$ denotes the set of selected points whose average movement exceeds τ. We then obtain coordinates of these points at the initial frame from $\mathcal{P}_m$ and denote this set as $\mathcal{P}_0$.

To generate the pseudo masks $\mathbf{M}$ for the active region at the initial frame, we feed the initial frame x_0 into the segmentation anything model (SAM) [264] and prompt SAM with the coordinates at the initial frame $\mathcal{P}_0$.

Given the derived pseudo mask $\mathbf{M}$ of the active regions, we define the pseudo active region

frame o at the initial frame x_0 as follows:

$$o = x_0 \circ \mathbf{M} + x_b \circ (1 - \mathbf{M}) \tag{5.4}$$

, where $\circ$ is element-wise multiplication and x_b is a background frame with entirely white pixels. This process allows us to bypass the need for manual labeling. Finally, we apply the pretrained image encoder $\mathcal{E}$ from Stable Diffusion [265] to generate the latent representation of the active region $\hat{o}$.

5.4.1.2 Latent Active Region Diffusion

Once we have constructed our dataset $\mathcal{D}_{\mathcal{O}}$ we define a model to identify the active region at test time. OurName adopts a conditioned latent diffusion model $\psi(\hat{o}|\hat{x}_o, c; \theta_\psi)$ to accomplish this. The diffusion model takes as input both a textual description of a task and the latent representation of the initial frame and outputs the active region. It is necessary to use both inputs because only together can the model accurately pinpoint areas of potential interaction. Without the frame, the model would not know where items are positioned and without the text description, the model would not understand the context of the task.

Diffusion models offer flexibility, allowing the selection of various active regions, which is crucial for tasks with multiple valid solutions. For example, in a task described as "put blue blocks into the box", any unplaced blue block can be identified as a valid active region. This flexibility is particularly beneficial in dynamic environments where multiple paths to task completion may exist.

During inference, we generate the active region using the trained active region generator

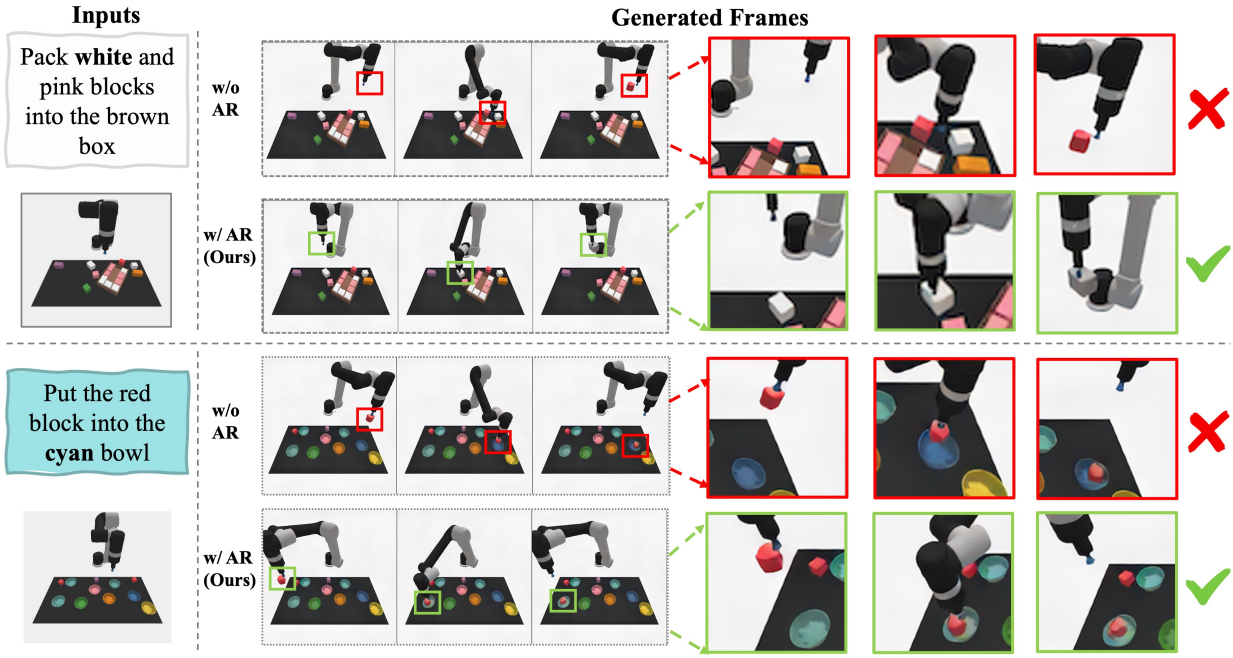


Figure 5.3: **Qualitative comparison of the generated video plans on CLIPort [7] unseen test tasks.** Our method successfully generates a video that packs the white block, while the counterpart w/o active region generates frames that pick the wrong object (the red block) or fails to correctly generate the arm position for grasping objects. We demonstrate higher visual quality, especially around objects of interest, thanks to the guidance provided by our active regions.

$\psi(\hat{o}|\hat{x}_o, c; \theta_\psi)$. The generated active region, along with the provided text and initial frame, is then input into the video planner $\phi(\cdot|\hat{x}_0, c, \hat{o}; \theta_\phi)$ for generating the latent sequence, as detailed below.

5.4.2 Universal Video Planner conditioned on Active Region

Motivated by the advances of text-to-video models [274], we develop a latent video diffusion model to serve as our trajectory planner. This module is designed to accurately generate future video latent representations based on an initial frame and a textual description of the task at hand as well as the active region.

Different from previous works, our proposed video generation model, $\phi(\cdot|\hat{x}_0, c, \hat{o}; \theta_\phi)$, integrates the active region alongside the initial frame and text description for conditioning. The

key insight is that by guiding our model with the active region, it can focus its generation efforts on areas crucial to the described task, enhancing the relevance and precision of the generated sequence.

To perform this integration, we concatenate each frame's latent with the latent of the active region $\hat{o}$. We do this in addition to the concatenation of the active region latent with the latent of the initial frame $\hat{x}_0$. This dual concatenation strategy ensures that, the model's denoising process for each frame aligns not only with the initial observation but also with the active region latents.

During inference, the input active region of $\phi(\cdot|\hat{x}_0, c, \hat{o}; \theta_\phi)$ is generated from the trained active region generator $\psi(\hat{o}|\hat{x}_o, c; \theta_\psi)$ given input text and initial frame. During training of $\phi(\cdot|\hat{x}_0, c, \hat{o}; \theta_\phi)$, the input active region is obtained from the training videos as detailed above.

5.4.3 Task Specific Action Decoding from Latent Sequence

Finally, we train a compact, task-specific latent inverse dynamics model to convert synthesized latent sequences directly into action sequences.

5.4.3.1 Latent Inverse Dynamics Model

Given two adjacent generated frame latents $\hat{x}_h$ and $\hat{x}_{h+1}$, our latent inverse dynamics module predicts the action a_h. The training of the inverse dynamics is independent of the video planner and can be done on a separate, smaller, and potentially suboptimal dataset generated by a simulator. Both the video planner and the inverse dynamics model utilize the same encoder $\mathcal{E}$ to map the RGB frame into the latent space. This ensures a consistent latent embedding space for both the video generation and the action decoding phases.

During inference, we decode actions directly from latent frames. Our latent-based inverse dynamic design represents a significant advancement over the existing RGB-based inverse dynamic models [8, 259, 266], which rely on RGB frames for action decoding. A main challenge with training video diffusion models to output RGB frames directly is the computational cost. Our latent dynamics module, on the other hand, bypasses the need to generate RGB frames because it can compute the action directly from the latent. This strategy not only streamlines the decoding process but also makes OurName's architecture more efficient and compact. For the purpose of visual interpretability, we can still employ the decoder $\mathcal{R}$ to transform these synthesized latents back into RGB frames if necessary.

5.4.3.2 Action Execution

During the execution phase of OurName, we start with x_0 and c. Next we generate the active region latent $\hat{o}$, followed by the synthesis of H-step latent sequence. The generated latents are then fed into our trained latent inverse dynamics model to produce the corresponding H actions. To enhance computational efficiency, we employ an open-loop control strategy by sequentially performing actions from the initially predicted action sequence.

5.5 Experiments

In the following sections, we evaluate our method on two public robot manipulation datasets, CLIPort [7] and BridgeData v2 [14]. First, we describe the implementation details in Sec. 5.5.1. In Sec. 5.5.2 we evaluate the generalization ability of our method across multiple tasks. We provide these results on CLIPort. Sec. 5.5.3 demonstrates the transfer ability of our method to

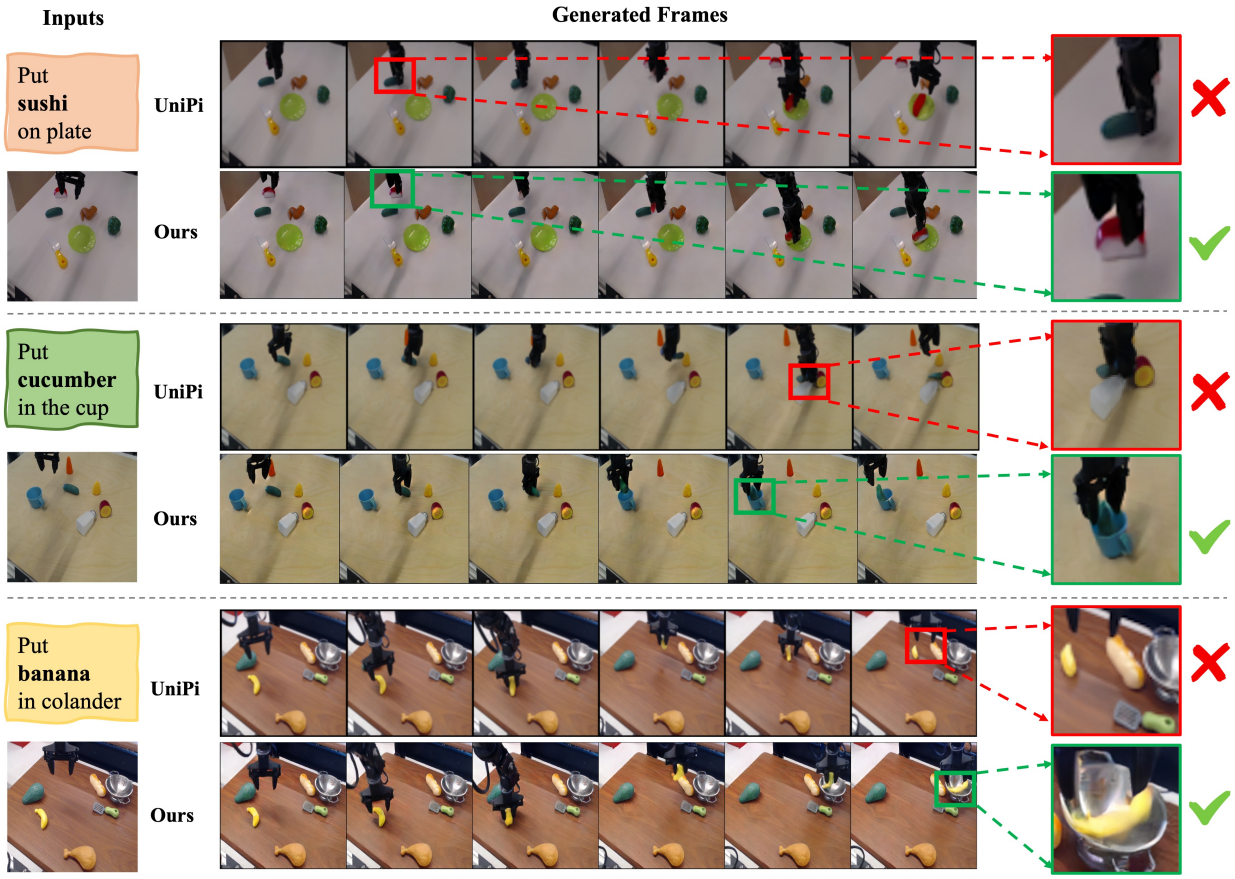


Figure 5.4: **Qualitative comparison of the generated video plans on BridgeData v2 [14].** Our method successfully picks up the correct object (sushi) or places the object in the appropriate location (in the cup or colander), while the UniPi* baseline selects the wrong object (cucumber) or places it incorrectly (on the desk), showing the advantage of our method.

real-world complex scenarios. We report results on BridgeData v2. Finally, ablation studies are provided in Sec. 5.5.4, and limitations are discussed in Sec. 5.5.5.

5.5.1 Implementation Details

For the active region generation, we use Co-Tracker [263], which requires a grid size to determine the total number of points to track. OurName uses a grid size of $M = 60$. We also set a threshold $\tau = 2$ for the minimum movement of a point trajectory to be considered active. We conduct data preprocessing of connected component labeling and analysis to retain pseudo masks

Table 5.1: Success rate in completing tasks across multiple environments in CLIPort [7]. When trained across various multi-task environments, our approach shows strong generalization capabilities to unseen environments, significantly outperforming the UniPi baseline [8]. UniPi* is reproduced based on the reference code provided by the authors.

Model	Place Bowl	Pack Object	Pack Pair
State + Transformer BC [277]	9.8	21.7	1.3
Image + Transformer BC [277]	5.3	5.7	7.8
Image + TT [278]	4.9	19.8	2.3
Diffuser [269]	14.8	15.9	10.5
UniPi* [8]	65.4	51.8	30.9
Ours	**86.7**(+21.3)	**69.0**(+17.2)	**46.6**(+15.7)

for manipulated objects while excluding those of the robot arm. The encoder $\mathcal{E}$ and decoder $\mathcal{R}$ are trained using a variational autoencoder as described in Stable Diffusion [265]. Images are resized to 512×512 when fed into the encoder, and the encoder produces a latent representation of shape $128 \times 128 \times 4$. The diffusion models and the inverse dynamic model are trained from scratch on 8 A100 GPUs.

To find the active region, we utilize the T5-XXL model [275] and a modified UNet architecture from [276]. We use the same architecture for the latent video diffusion model but utilize the temporal modules modification proposed in [260] for generating $H = 6$ frame latents. We adopt a learning rate of 3×10^{-4} and a batch size of 96 for training.

Our latent inverse dynamic model consists of convolutional layers with skip connections followed by a linear layer to project features into 7-dimensional action vectors for manipulation tasks. This latent action decoder is trained with a learning rate of 5×10^{-4} and a batch size of 3072.

5.5.2 Multi-Environment Transfer on CLIPort

Setup. CLIPort [7] is an extended benchmark of language-grounded tasks for manipulation in the simulator Ravens [279]. We train our models on 110k demonstrations spanning 11 distinct tasks and evaluate on 3 unseen tasks in CLIPort. Testing on 3 unseen tasks allows us to evaluate the model's ability to generalize to new tasks. We follow the multi-environment transfer setup in UniPi [8]. We report the final task completion success rate across new instances of the test environment and associated language prompts.

Baseline. UniPi [8] generates RGB video frames and employs an RGB-based inverse dynamic model. We provide results on our reproduction of UniPi based on the reference code provided by the authors, denoted as UniPi*. We aim to match the number of in-domain training demonstrations. Nevertheless, the performance discrepancy could still be the result of differences in training data, as the training and pre-training data for UniPi is not publicly available. Additionally, we also compare with the same set of imitation learning baselines listed in [8]. We refer readers to the original paper for a detailed description of the baselines.

Results. In Table 5.1, we show the performance of our method compared with established baselines on three unseen tasks in CLIPort. Notably, our method significantly improves the success rate over the UniPi baseline by 21.3%, 17.2%, and 15.7% for the Place Bowl, Pack Object, and Pack Pair tasks, resp. This performance is driven by better generation due to our active region-conditioned diffusion models and improved execution produced by the latent inverse dynamics module. We visualize generated video plans in Figure 5.3, highlighting the visual improvements of OurName over a counterpart model without active region conditioning. The counterpart is the same latent video diffusion model as ours except for conditioning on the active

region. Using an active region results in higher visual quality, especially around objects of interest. For example, our method successfully generates a video that packs the white block, while the counterpart generates frames that either pick the wrong object or fail to generate the arm position correctly.

5.5.3 Real-world Transfer on BridgeData v2

Setup. BridgeData v2 [14] is a large and diverse real-world dataset for robotic manipulation. Bridge's examples show a wide variety of skills using various unique objects in diverse environments. This dataset includes 60,096 trajectories with natural language instructions, gathered from 24 environments with a widely accessible, cost-effective robot. We use 95% of the data for training and the rest for evaluation.

Results. As illustrated in Figure 5.4, our method is able to generate realistic video plans that effectively handle chaotic environments, diverse language instructions, and varying camera poses. In the examples shown, our approach is continuously able to pick up the correct object and place it in the correct location. In contrast, the UniPi baseline often selects the wrong object and

Table 5.2: Effect of Active Region Quality on Performance. The higher-quality active region during training and testing phases, the more improvement in task success rates, highlighting the significance of the active region.

AR (Trn)	AR (Test)	Video Planner Setting	Place Bowl	Pack Object	Pack Pair
-	-	w/o Active Region	83.4	67.7	38.0
Unsupervised	Predicted	+Active Region	86.7	69.0	46.6
		Δ	+3.3	+1.3	+8.6
Supervised	Predicted	+Active Region	93.3	79.6	51.7
		Δ	+9.9	+11.9	+13.7
-	GT	+Active Region	100.0	92.1	62.8
		Δ	+16.6	+24.4	+24.8

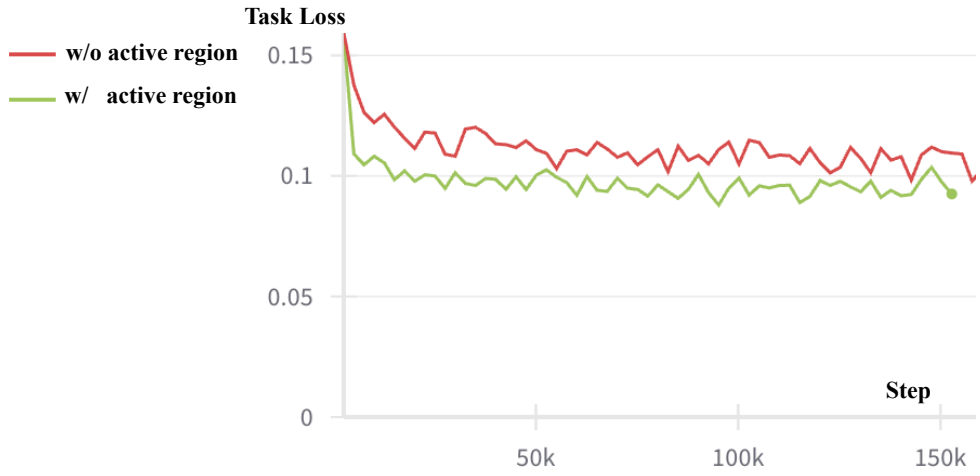


Figure 5.5: **Task Loss on CLIPort [7] test set.** Our method (green) shows significantly lower task loss compared to that without the active region (red), underscoring the active region's role in enhancing generation quality.

places it incorrectly. For example in the top example instead of picking up the sushi the UniPi baseline picks up a cucumber. In the bottom example, our method successfully finds the colander while the UniPi baseline places the banana back on the table. These results showcase the robust adaptability of our method across a wide range of complex real-world scenarios, highlighting its superior performance in understanding tasks in challenging settings.

Figure 5.6 showcases the high-quality pseudo masks generated by our method in real video scenarios. This is achieved without the need for manual annotations. This visualization underscores the effectiveness of our approach in accurately identifying moving objects directly from video data.

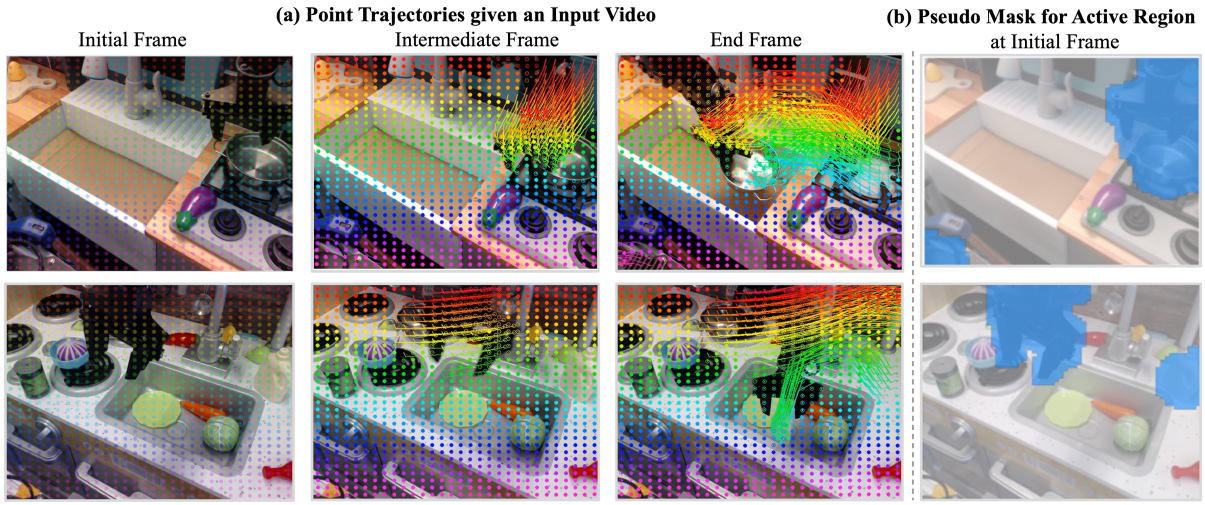


Figure 5.6: **Visualization of pseudo active region in BridgeData v2.** We visualize the dense point trajectories and the obtained pseudo masks of the active region in BridgeData v2 [14].

5.5.4 Ablation Study

5.5.4.1 Quantitative evaluation of generation quality for control tasks

To quantitatively evaluate the video generation quality for control-specific tasks, we propose a generation metric named task loss. Our task loss is calculated by feeding the generated video latents into our pre-trained latent inverse dynamic module for action decoding and then computing the L1 error between the decoded actions and the ground truth actions. Task loss provides a metric for how well the generated video is aligned with the expected action sequence. A visual comparison of the task loss curves for the CLIPort test set is shown in Figure 5.5. The green curve which represents our model when conditioned on active regions has a significantly lower task loss than the red line, which is the counterpart with no active region input. This result emphasizes the active region's role in enhancing generation quality and enabling more precise action decoding.

5.5.4.2 Effect of Active Region Quality on Performance

Table 5.2 shows the impact of active region quality on the success rate. Incorporating active regions in the unsupervised way described in the paper yielded a performance boost of +3.3%, +1.3%, and +8.6%. This boost is even more (+9.9%, +11.9%, and +13.7%, respectively) when the active region prediction model is trained with ground truth active regions. Furthermore, using ground truth active regions during inference led to even more substantial boosts (+16.6%, +24.4%, and +24.8%, respectively). These results show that the task performance continues to improve with higher quality active regions and underscore its critical role in enhancing task success rates through video-based policy learning.

5.5.5 Limitations and Future work

One limitation of our approach is that pseudo-active region quality depends on Co-Tracker, which may struggle with long-horizon tasks or complex backgrounds. Advanced dense point tracking models could improve pseudo label quality, and noisy label learning techniques might stabilize training. Alternatively, text-guided segmentation could be used, where the vision-language model identifies the target object based on the task, followed by segmentation using tools like Segment Anything. We leave these as future work.

5.6 Conclusion

In this work, we present ARDuP, a novel framework for video-based policy learning by prioritizing active region generation. The video planner in ARDuP is a decomposed video generator: firstly, active regions are generated via a latent conditional diffusion model based on text

input and an initial frame; secondly, video latents are synthesized with additional active region conditioning. The synthesized latent sequences are then decoded into action sequences through a latent inverse dynamic model. During training, we obtain active region supervision without human annotation by using Co-Tracker for moving point tracking and SAM for pseudo mask generation. ARDuP is validated on the CLIPort simulator and the real-world BridgeData v2 dataset, showing notable success rate improvement and the strong ability to generate realistic and action-centric video plans.

Chapter 6: Learning Robust Robot Behavior Alignment through Preferences

6.1 Introduction

A key challenge when using reinforcement learning (RL) to train autonomous agents for various tasks is defining a suitable reward function. The reward function needs to give dense feedback to guide the agent's learning while preventing unintended behaviors (series of actions) that might earn high rewards but do not match the user's intent. However, creating an effective reward function is often complex and time-consuming. It involves multiple rounds of testing with different adjustments to ensure the agent learns the intended behavior without developing unplanned habits.

To address these difficulties, Preference-based Reinforcement Learning (PbRL) has emerged as a promising approach. PbRL sidesteps the challenge of explicitly designing a reward function by instead using human preferences as the reward signal [280–286]. PbRL employs a human-in-the-loop methodology, where humans provide preference feedback by comparing pairs of trajectory segments and labeling which segment is closer to the goal. This approach addresses the challenge of designing explicit reward functions and can result in behaviors that are better aligned with human intent [280]. Recent advances in this domain have integrated vision-language models (VLMs) [287], which can autonomously generate preferences, potentially reducing the need for direct human involvement.

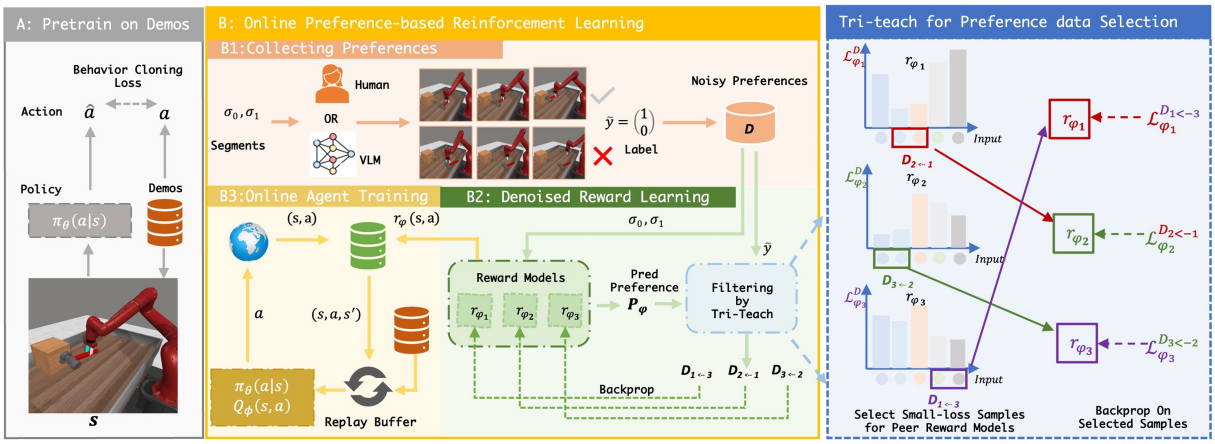


Figure 6.1: **Overview of our method TREND.** First, we pretrain the policy network using behavior cloning (BC) with few-shot expert demonstrations for effective exploration (A). In the online training phase, noisy preferences are collected from human annotators or a vision-language model (B1). We then apply our tri-teaching strategy for denoised reward learning, where three collaborative reward models identify clean preference samples for each other (B2). Finally, the learned reward model is used to guide the agent's training (B3), ensuring robust performance despite noisy labels.

However, in practice, both human and VLM-generated preference labels can be noisy and inconsistent. Human feedback is prone to bias and VLM labels often struggle with interpreting visual content, task-specific text, and temporal dynamics. For example, if two behaviors are qualitatively similar, it is likely that the preference labels provided by either human annotators or VLMs are not meaningful. As shown later in our experimental analysis (Sec. 6.5.3), VLM preference labels can have a noise rate up to 40%, making them insufficient for reward learning without denoising. To make matters worse, a recent work [288] showed that even a 10% label corruption rate can significantly degrade performance. This underscores the need for robust preference-based RL methods that can effectively handle noisy labels.

In this work, we propose (TREND), a novel framework to address the challenge of noisy preference labels, as shown in Figure 6.1. Our key insight is the use of peer models to identify clean samples from a batch of noisy labels, instead of using a single model that can be biased in its estimation of noise. We therefore introduce a tri-teaching strategy where three reward models

collaboratively select samples for training. Each model answers the question, "*Which preference pair is more likely to have clean labels?*", by computing losses over preference pairs and labeling those with small losses as likely clean. These selected pairs are then used to update the other models.

Our tri-teaching strategy is advantageous compared with existing works that simply average multiple reward model's predictions for sample selection [289] for the following reasons: (a) each peer model independently develops expertise in sample selection, enhancing robustness to noise, and (b) the definition of clean samples is dynamically learned from peer models rather than being fixed.

Additionally, to further combat extremely noisy preference labels, we integrate few-shot expert demonstrations into both pretraining and online PbRL adaptation. This provides a strong initialization for policy and ensures that at least some of the training data remains free of noise during online learning. Our experimental results demonstrate that under the extremely high noise rate of 40%, even a few demonstrations (1-3), we can achieve a success rate of $\sim$80%.

We evaluate our proposed method, TREND, on robotic manipulation tasks from Meta-world [290]. Our approach consistently outperforms baseline methods by a large margin, even under high noise (e.g., with noise rates up to 40%). We also conduct a detailed ablation study to demonstrate the benefits of our key components. The main contributions of this work are as follows:

- We present TREND, a novel framework that integrates expert demonstrations for robust learning under noisy preferences.
- We introduce a simple yet effective tri-teaching strategy for effective label denoising, lever-

aging peer models for cyclic clean feedback selection.

- TREND consistently outperforms existing PbRL baselines across a range of robotic manipulation tasks on Meta-World, even under varying levels of label noise. Notably, in scenarios with an extremely high noise rate of 40%, TREND achieves significant improvements in success rates: nearly 40% on the Button-Press task, 60% on the Drawer-Open task, and 70% on the Hammer task, compared to baselines using the same number of demonstrations. Even with noisy VLM-generated preference labels, TREND demonstrates an impressive success rate improvement of over 40% on the Drawer-Open task, showcasing its robustness in handling noisy data.

6.2 Related Work

Preference-based RL. While providing precise rewards for many tasks can be challenging for humans, expressing preferences for certain actions or states is often more intuitive and easier. Due to the simplicity and richness of relative preferences, preference-based reinforcement learning (PbRL) has gained significant attention in recent years [281, 282, 284–286, 288, 291]. Various approaches have been proposed to enhance the sample efficiency of PbRL. For example, PEBBLE [280] uses unsupervised policy pre-training to warm-start the policy and encourage exploration from the outset, while [292] uses imitation from expert demonstrations for policy initialization. Other works have focused on developing sampling strategies to select informative preference queries [282, 293, 294] and addressing exploration challenges by estimating reward model uncertainty [295].

Noisy label learning. Another critical issue with PbRL is its sensitivity to noise in preference

labels. Lee et al. [288] showed that even a modest 10% rate of corrupted preference labels can substantially impair algorithmic performance. This concern becomes more pronounced in broader application contexts where preferences are collected from non-experts, increasing the likelihood of noise in labels. The issue of noisy labels is well-studied in the context of supervised learning [296] with proposed solutions such as sample selection methods [65, 297], architectural modifications for noise adaptation [298], and regularization [299]. These methods, however, are not easily transferable to PbRL due to limited sample sizes [289]. To alleviate this problem in PbRL, [300] introduced an encoder-decoder framework to capture diverse human preferences and improve robustness, though it required a substantially larger amount of preference data. RIME [289] introduced a dynamic sample selection method using a denoising discriminator to filter out the noise and incorporate a warm start strategy to mitigate errors from noisy labels. In contrast, we select reliable samples by a tri-teaching method and, optionally, explicitly incorporate a few expert demonstrations to combat extremely high noise.

Vision Language Models (VLMs) for RL. Recent advancements in integrating VLMs into RL have shown promising outcomes [18,22,47,77,78,85,141,143,261,287,301–310], particularly in reward specification and preference-based methods. LLMs/VLMs have been employed to generate reward functions from text descriptions of goals, such as generating reward code directly [311] or serving as proxy reward functions aligned with given prompts [312]. Beyond reward generation, VLMs have been utilized for generating preference feedback, such as in RL-VLM-F [287], which uses VLMs to generate preference labels over pairs of the embodied agent's image observations. However, a key problem with VLM-generated preference labels is that they are often unreliable leading to high noise, so directly feeding all VLM feedbacks can lead to suboptimal (or worse) performance. In this work, we address this issue by introducing a tri-teaching method,

supplemented with a few expert demonstrations, to denoise VLM's noisy preference labels.

6.3 Preliminaries

Preference-based RL. In classical RL [313], an agent learns by interacting with the environment for T discrete time steps. At each step t the agent observes the environment state s_t and then uses a policy π to choose an action a_t. After executing the action, the environment returns a reward r and transitions to the next state s_{t+1} following the environment's dynamics. The objective of the learning agent is to learn a policy that maximizes the expected return, which is the discounted cumulative reward over time: $R_t = \sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k})$ where $\gamma \in [0, 1)$ is the discount factor, which balances the agent's preference for immediate versus long-term rewards.

Defining an explicit reward function can be difficult or infeasible in many real-world tasks. PbRL addresses this by relying on a teacher (often a human or an expert model) to provide feedback in the form of preferences between different behaviors (trajectory segments) of the agent [280, 289, 314, 315] (e.g., in Fig 6.1 B1). Instead of receiving scalar rewards from the environment, the agent learns from the teacher's preferences, using them to construct an internal reward model.

Formally, a trajectory σ is defined as a sequence of state-action pairs $\{(s_k, a_k), \ldots, (s_{k+H-1}, a_{k+H-1})\}$ over a fixed segment size H. Given two trajectory segments σ_0 and σ_1, the teacher provides feedback in the form of a preference label $\tilde{y} \in \{(1, 0), (0, 1), (0.5, 0.5)\}$. A "preference" is expressing which is preferred between pairs of clips of the agent's behavior, essentially distinguishing which segment has a higher reward. Here, $\tilde{y} = (1, 0)$ indicates $\sigma_0 \succ \sigma_1$. Correspondingly, $\tilde{y} = (0, 1)$ indicates $\sigma_1 \succ \sigma_0$, and $\tilde{y} = (0.5, 0.5)$ indicates that the teacher is indifferent between the two

trajectories. The teacher's feedback is stored in a dataset D as a tuple $(\sigma_0, \sigma_1, \tilde{y})$. The agent uses this data to learn a reward function r_ψ.

The probability that the teacher prefers σ_1 over σ_0 can be modeled using the Bradley-Terry model [316], based on the estimated reward function r_ψ as

$$P_\psi[\sigma_i \succ \sigma_j] = \frac{\exp\left(\sum_t \hat{r}_\psi(s_t^i, a_t^i)\right)}{\sum_{i\in\{j,i\}} \exp\left(\sum_t \hat{r}_\psi(s_t^i, a_t^i)\right)}. \tag{6.1}$$

The reward function r_ψ is learned by minimizing the cross-entropy between the predicted preference and the label provided by the teacher. The loss function is defined as:

$$\begin{aligned}\mathcal{L}_{\text{CE}}(\psi) = \mathbb{E}\left[\ell_{\text{Reward}}\right] = -\mathbb{E}\Big[&\tilde{y}(0) \ln P_\psi[\sigma_0 \succ \sigma_1] \\ &+ \tilde{y}(1) \ln P_\psi[\sigma_1 \succ \sigma_0]\Big]\end{aligned} \tag{6.2}$$

6.4 Method: Tri-teaching for Robust Preference-based Reinforcement Learning with Demonstrations (TREND)

In this section, we will describe our approach, (TREND), in detail. TREND simultaneously trains three reward networks. Each reward model updates its parameters by using the preference pairs selected from the peer network via our proposed tri-teaching (Sec. 6.4.1). To further mitigate the impact of noise, we can leverage expert demonstrations in the learning process (Sec. 6.4.2).

6.4.1 Tri-teaching for Preference Data Selection

Following the small-loss principle [65], we hypothesize that the loss of a preference pair is positively correlated with how noisy the label is. Since a single model can be biased in its estimation of noise, potentially leading to error accumulation, we introduce a tri-teaching strategy where three peer reward networks collaboratively select samples for training. Each network answers the question, "*Which preference pair is more likely to have clean labels?*" and teaches the other networks by providing these samples for training. We use three teachers as existing work [280, 281, 289] used an ensemble of three reward models to improve the stability in reward learning. Using an odd number of experts [317, 318] is also a standard practice to avoid ties and provide a majority consensus signal.

Formally, we define a selection loss $\mathcal{L}_{\psi}^{\mathcal{D}}$ to quantify how noisy the label is as estimated by the reward model r_{ψ}. Given a batch of training preference data $\mathcal{D} = \{(\sigma_0^i, \sigma_1^i, \tilde{y}_i)\}_{i=1}^N$, where (σ_0^i, σ_1^i) is a pair of trajectory segments, and $\tilde{y}_i$ represents the corresponding preference label, we compute the selection loss for model ψ_k as

$$\mathcal{L}_{\psi_k}^{\mathcal{D}} = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{reward}}((\sigma_0^i, \sigma_1^i), \tilde{y}_i; \psi_k), \quad k \in \{1, 2, 3\}, \tag{6.3}$$

where k is the index of reward models.

During training, each reward model ψ_k identifies the subset of samples that have the smallest selection loss within a batch of preferences. The size of the subset is determined by a selection rate γ, which controls the proportion of samples retained. These samples are considered clean

and are passed to a peer model for training. The selected data for peer training is defined as

$$\mathcal{D}_{k \leftarrow j} = \underset{\bar{\mathcal{D}}:|\bar{\mathcal{D}}| \geq \gamma |\mathcal{D}| \text{ and } \bar{\mathcal{D}} \subseteq \mathcal{D}}{\arg\min} \mathcal{L}_{\psi_j}^{\bar{\mathcal{D}}}. \tag{6.4}$$

Here $(k, j) \in \{(2, 1), (3, 2), (1, 3)\}$, and $\mathcal{D}_{k \leftarrow j}$ denotes the clean preferences identified by the reward model r_{ψ_j} which are then selected as training samples for r_{ψ_k}. In this way, the three models exchange opinions on which samples are clean, thereby enabling each to learn from its peers.

After the selection process, each reward model updates its parameters by using the preference pairs selected from the peer network as

$$\psi_k = \psi_k - \eta \nabla_{\psi_k} \mathcal{L}_{\psi_k}^{D_{k \leftarrow j}}. \tag{6.5}$$

This process is cyclic, where $\hat{r}_{\psi_1}$ selects for $\hat{r}_{\psi_2}$, $\hat{r}_{\psi_2}$ selects for $\hat{r}_{\psi_3}$, and $\hat{r}_{\psi_3}$ selects for $\hat{r}_{\psi_1}$. This tri-teaching strategy allows the models to collaboratively denoise the labels and improve learning stability despite noisy preferences.

We ensure diversity among the reward models by using different weight initialization and permuting the input sample orders. This prevents them from collapsing into similar models, maintaining diverse learning dynamics.

6.4.2 Incorporating Few-shot Expert Demonstrations

In the presence of noisy preference labels, purely learning from trial and error through online reinforcement learning can lead to sub-optimal performance. To further mitigate the im-

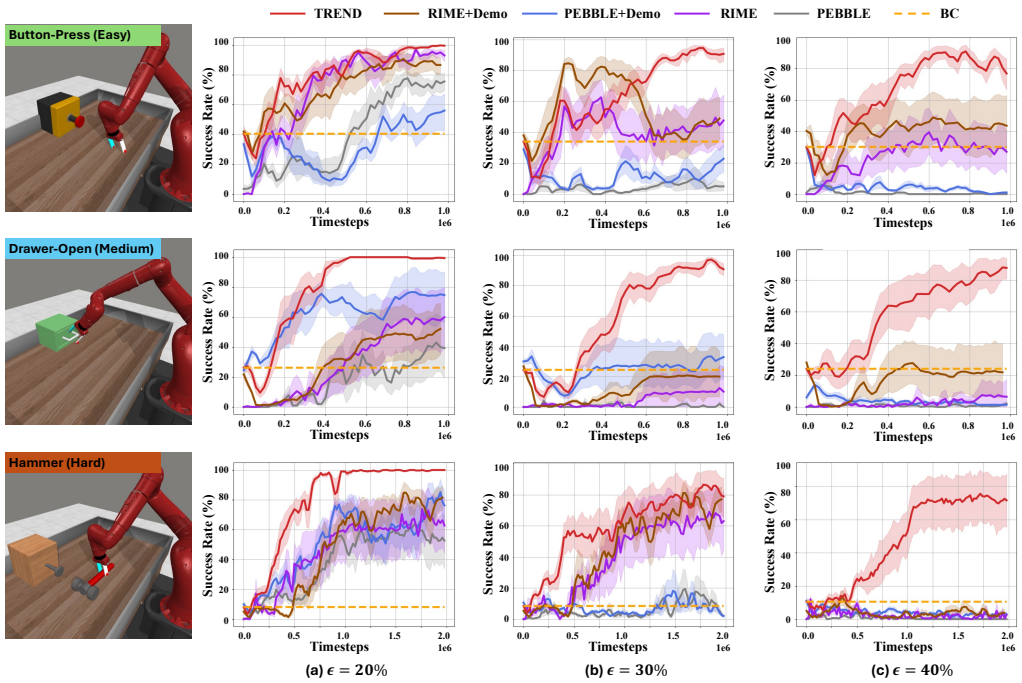


Figure 6.2: **Learning curves for robot manipulation tasks on Meta-world.** Each row represents results for a specific task and each column corresponds to a different error rate ϵ. Results are averaged over five seeds. Shaded Areas represent standard deviation across seeds.

pact of noise and to provide a clean supervisory signal, we incorporate expert demonstrations into the pertaining and online training process. Expert trajectories offer high-quality information, which is particularly valuable when learning under high-noise conditions. These demonstrations serve to guide the agent toward better behavior by ensuring that a portion of the training data is noise-free.

Our method builds upon the widely used PbRL algorithm PEBBLE [280]. In the early stages of training, querying the teacher for preference labels often yields uninformative results, as the agent's policy tends to be random and the states uninteresting. To mitigate this issue, PEBBLE [280] introduced unsupervised exploration through pretraining the agent with an intrinsic reward based on state entropy [319], a strategy that has been widely adopted in subsequent research [281, 289].

To encourage more diversified exploration, we replace the unsupervised exploration phase

of the PEBBLE algorithm with policy pretraining on demonstration data using behavior cloning (BC). Even though we are only using a few demonstrations, BC pretraining is significantly more effective than PEBBLE's intrinsic reward-based pretraining [280, 319]. BC establishes a stronger prior for the policy, enhances exploration, and improves the sampling of informative preference pairs. This is especially beneficial to hard tasks where extropy-based exploration struggles to find useful signal.

Next, after the policy initialization phase by BC, during the online training phase, for each batch of training data, we sample a proportion of $\alpha\%$ from the expert demonstration dataset $\mathcal{D}_{\text{expert}}$, where $\alpha\%$ controls the proportion of expert demonstrations. For the remaining data, we sample from the non-expert RL replay buffer $\mathcal{D}_{\text{replay}}$. Specifically, for data from the expert demonstrations, we update the policy network π_θ using the BC loss as

$$\mathcal{L}_{\pi_\theta}^{\text{BC}} = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}_{\text{expert}}}\left[\|\pi_\theta(s_t) - a_t\|^2\right]. \tag{6.6}$$

For non-expert data from the replay buffer, we apply the standard Soft Actor-Critic (SAC) [320] loss to update the policy network π_θ

$$\mathcal{L}_{\pi_\theta}^{\text{SAC}} = \mathbb{E}_{s_t\sim\mathcal{D}_{\text{replay}},a_t\sim\pi_\theta(\cdot|s_t)}\left[\alpha\log\pi_\theta(a_t \mid s_t) - Q_\phi(s_t,a_t)\right],$$

where Q_ϕ is the critic network. We then update the policy network π_θ using a weighted average of the two losses as

$$\mathcal{L}_{\pi_\theta}^{\text{TREND}} = \mathcal{L}_{\pi_\theta}^{\text{SAC}} + \lambda_{\text{BC}}\mathcal{L}_{\pi_\theta}^{\text{BC}}. \tag{6.7}$$

This way, the BC loss serves as a strong regularization term during online training, helping the

policy network retain its BC initialization and avoid over-reliance on noisy preference data, which is especially beneficial in high-noise conditions.

6.5 Experiments

6.5.1 Experimental Setup

Tasks. We evaluate TREND on three robotic manipulation tasks from Meta-world [290]: Button-Press, Drawer-Open, and Hammer. These tasks are visually rich, with diverse textures and shading, requiring precise, fine-grained control to successfully manipulate objects. The tasks are shown in the first column of Figure 6.2.

To analyze our method's effectiveness, we consider two types of noisy annotators: one that generates preferences using a scripted policy and another that utilizes a VLM-based preference generator.

Scripted Noisy Preference Annotator. We generate synthetic preference feedback with an oracle reward following the technique used in previous works [280, 289]. Specifically, we assume a scripted teacher that determines the preference between two trajectory segments based on the sum of the ground-truth reward for each segment. To introduce noise, we flip each preference label with a probability of $\epsilon = 20\%, 30\%, 40\%$ as done in RIME [289]. This allows us to easily control the level of noise in preference labels.

VLM Preference Annotator. We also evaluate TREND using preference labels generated by a VLM. This approach follows the work done in RL-VLM-F [287]. For all VLM experiments, we use Gemini-1.5-Flash [321]. We prompt the VLM using rendered images of the two trajectory segments and the task description and obtain the generated preference label following [287]. If

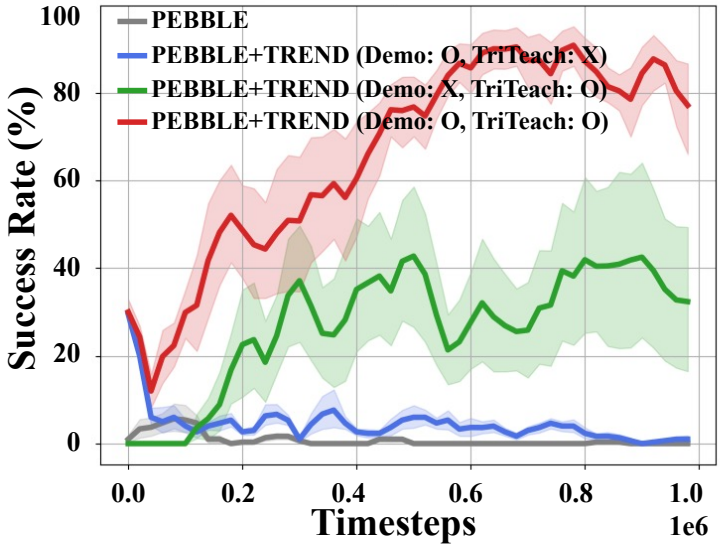


Figure 6.3: **Contribution of each component on Button-Press** ($\epsilon = 40\%$)**.** Our tri-teaching boosts performance significantly, and adding a single expert demonstration further enhances success, emphasizing the need for both components.

the VLM predicts that the provided segments are too similar, it will return a "no preference" answer and we skip this pair for reward training [287].

Implementation Details. We use PEBBLE [280] as our backbone algorithm. For the tri-teaching part of our method, we set $\gamma = 0.6$ (in Eq. 6.4). TREND uses 1 expert demonstration for Button Press, 2 for Drawer Open, and 3 for Hammer, with the BC loss coefficient $\lambda = 4.0$ (in Eq. 6.7). The proportion of sampling expert demos $\alpha\%$ is linearly decayed from 50% to 25% within the first 50k timesteps. We use the same hyperparameters for all experiments unless otherwise stated.

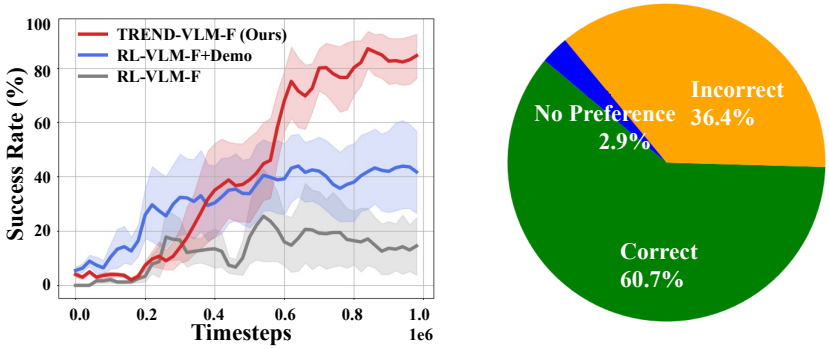
All other hyperparameters are consistent with PEBBLE. Both the baselines and our method use an ensemble of three reward models. The segment size H (i.e., length of σ) is 50. Preference queries are selected using disagreement sampling, following [289]. Disagreement sampling chooses pairs with high uncertainty based on the variance across an ensemble of predictors. All baseline methods are implemented using publicly available repositories. The initial state is randomly initialized for each episode, ensuring diverse configurations. We run each experiment with

five random seeds and report the average performance with standard deviations.

6.5.2 Main Results

Figure 6.2 presents the learning curves of TREND alongside baseline methods on Meta-world, evaluated across three tasks: Button-Press (easy), Drawer-Open (moderate), and Hammer (hard). We compare our method against PEBBLE [280], which lacks denoising techniques, and RIME [289], the current state-of-the-art (SOTA) method for denoising in preference-based reinforcement learning (PbRL). For a fair comparison, we incorporated expert demonstrations into both PEBBLE and RIME in the same way and the same amount as ours, referring to these as PEBBLE+demo and RIME+demo. We also include a behavior cloning (BC) baseline learning purely from the demonstrations without reinforcement learning.

As shown in Figure 6.2, our approach consistently outperforms all baselines across all noise levels, from 20% to 40%. Notably, we achieve over 80% success at 40% noise with just 1 demonstration for Button-Press, 2 for Drawer-Open, and 3 for Hammer. This success is driven by our tri-teaching strategy for label selection and the integration of expert demonstrations. On average the baselines either perform at half of our success rate or fail completely. TREND is also superior to PEBBLE+demo and RIME+demo, showing the benefits of TREND come from both the expert demonstrations and the cross-model selection, not the expert demonstrations alone. These results underscore TREND's robustness to noisy preferences and its effectiveness in high-noise environments.



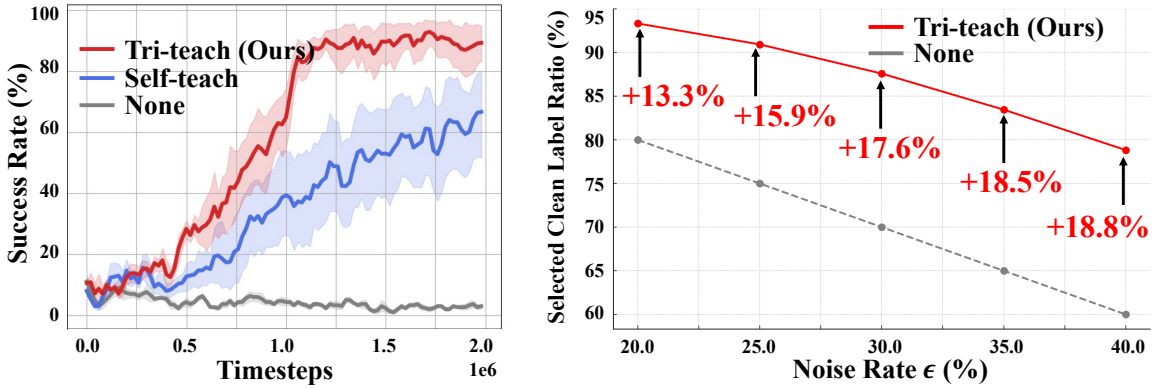
(a) Results w/ VLM Preference Labels **(b) VLM Label Quality**

Figure 6.4: **Results on Drawer-Open using VLM (Gemini-1.5-flash) to generate preference feedback.** Our TREND-VLM-F achives the best result (left) under the high noise rate of VLM labels (right).

6.5.3 Ablation Study

Effects of individual components. We study the effects of individual components as shown in Figure 6.3. When 40% of the labels are noisy, PEBBLE (grey) completely fails, as it lacks any denoising mechanism. However, when our tri-teaching strategy is applied, the success rate jumps to approximately 40% (green), highlighting its effectiveness in selecting clean labels from noisy data. Adding just one expert demonstration on top of the tri-teaching (orange) boosts the success rate to around 90%, demonstrating the critical role of even minimal clean supervision. Notably, adding an expert demo to PEBBLE alone (blue) does help avoid total failure, but is insufficient to achieve strong performance. These results underscore the necessity of both robust label selection through tri-teaching and additional clean supervision from expert demonstrations to effectively deal with high noise levels.

Performance with VLM Preference Annotator. Figure 6.4 compares our method with RL-VLM-F [287] on the Drawer-Open task, where a VLM is used for preference labeling instead of a scripted annotator. We observe that even with an advanced proprietary VLM such as Gemini-1.5-Flash, the generated preference labels still have a noise rate as high as 36.4%. Consequently,



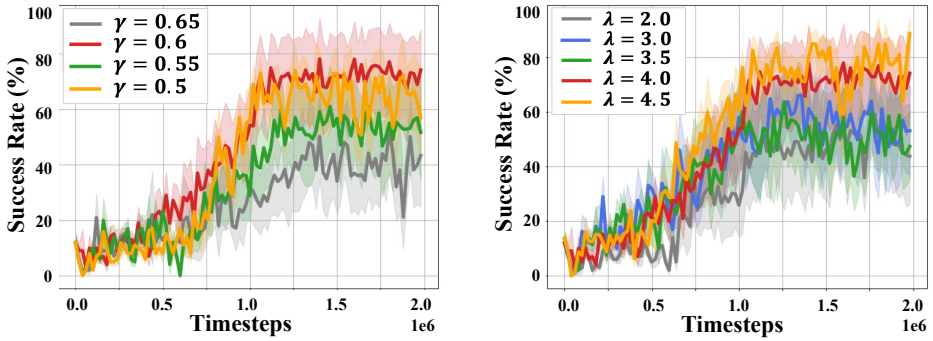
(a) Comparison of Denoising Strategies **(b) Analysis of Selected Clean Label Ratio**

Figure 6.5: **(Left):** Comparison between our method (w. Tri-teach) and a baseline denoising strategy (w. Self-teach) on Hammer ($\epsilon = 40\%$). **(Right):** Comparison of clean label ratio under different noise levels between our method (w. Tri-teach) and the baseline (w/o. Tri-teach) on Hammer.

as shown in the figure, RL-VLM-F (grey) achieves only a $\sim$20% success rate due to the high noise level, although adding one expert demonstration improves its success rate to about 40% (blue).[1] In contrast, our method, TREND-VLM-F, achieves the best performance, reaching a $\sim$90% average success rate. This demonstrates TREND's effectiveness in handling high noise VLM-generated preference labels.

We also tested on Hammer and Button Press, but the limitations of the current VLM preference annotator, Gemini-1.5-Flash, led to high noise rates of up to 50%, causing both the baseline and our method to struggle. These results underscore the challenges that current VLMs face, such as difficulties in adapting to the visual appearance of simulations, understanding visual and temporal content, and selecting optimal viewpoints. While future advances in VLM development may help mitigate these issues, our efforts for denoising preference labels address a complementary aspect of improving VLM preference annotation quality.

[1]Note that the results reported in the original RL-VLM-F paper were obtained using a deterministic environment based on its released codebase, whereas we follow standard practice by randomizing the initial robot arm position and goal location.



(a) Selection rate γ **(b) BC loss weight λ**

Figure 6.6: **Hyperparameter analysis on Hammer** (ϵ=40%).

Is cyclic preference data selection and training really necessary? In Figure 6.5 (left), we compare TREND with a baseline denoising strategy on the Hammer task with $\epsilon = 40\%$. Instead of cyclically selecting data with a small loss and passing it to another model for training, the baseline performs self-teaching, where each model uses its own small-loss samples for training. As shown in the figure, our method achieves the highest success rate, while the ablated self-teach approach performs worse than ours but still better than PEBBLE, which does not have any denoising mechanism. This demonstrates that selecting samples based on small loss can reduce noise to some extent, and that cyclic cross-model selection can further help denoising the preference labels.

Analysis of label quality with Tri-teach. In Figure 6.5 (right), we further analyze the clean preference label ratio achieved by our tri-teaching denoising method. Here, clean preference label ratio refers to the proportion of training data that has correct preference labels. On the Hammer task, using a scripted preference annotator, we documented the clean label ratio under varying noise levels (20%, 25%, 30%, 35%, and 40%) and compared it to the baseline PEBBLE without any denoising mechanisms. As shown in the figure, with our tri-teach mechanism, even at a noise ratio as high as 40%, the clean label ratio reaches $\sim$80%. This demonstrates that our

method significantly enhances preference label quality, particularly in high noise conditions.

Impact of hyperparameters. We analyze how the hyperparameters of TREND influence the performance of PbRL under a high noise rate of 40%. In Figure 6.6, we present the learning curves for TREND with varying hyperparameters: (a) selection rate $\gamma \in \{50\%, 55\%, 60\%, 65\%\}$ and (b) BC loss weight $\lambda \in \{2.0, 3.0, 3.5, 4.0, 4.5\}$. We observe that TREND is generally robust to changes in γ, but performance can degrade when γ is too high or too low. This is likely due to selecting more noisy labels or insufficient data. Ideally, selecting a γ that aligns with the estimated clean label rate can optimize performance. Regarding the BC loss weight λ, TREND shows robustness across different values, with potential performance boosts when appropriately tuned.

6.6 Conclusion

In this work, we study learning from noisy preference labels in PbRL formulation. We propose TREND, which leverages a tri-teaching strategy and incorporates few-shot expert demonstrations. Through cyclic peer training among three reward networks, TREND effectively filters out incorrect preference labels, improving label quality even under high noise levels. The integration of a few expert demonstrations further enhances its performance, providing a strong supervisory signal and regularization. Our experimental results on robotic manipulation tasks on Meta-world demonstrate that TREND consistently outperforms baseline methods, achieving high success rates even when the noise in the preference labels reaches up to 40%, highlighting the robustness of our methods under noisy preference labels.

Chapter 7: Future Work and Conclusion

7.1 Future Work

While this thesis establishes a cohesive framework for structured alignment, several promising avenues remain for future exploration.

One direction is to develop end-to-end unified video-based policy learning systems. This involves using a vision-language model as the backbone, with one branch trained to predict actions and another to generate future latent representations during training. At inference time, only the action prediction branch is retained. Such a design could streamline policy execution while benefiting from the temporal structure during learning.

Another direction is to advance world modeling for long-horizon tasks. This includes modeling temporally extended dynamics, reasoning about latent future states, and enabling lookahead planning beyond short reactive behaviors. Integrating learned dynamics models with structured visual representations could improve planning and generalization in complex environments.

Joint spatial-temporal alignment also remains an open problem. A unified architecture that models both fine-grained pixel-level correspondences and longer-range object or trajectory alignment may improve performance on tasks such as video object tracking, action segmentation, or video question answering.

In addition, multi-modal active alignment could further improve policy learning by com-

bining visual signals with language, audio, or haptic feedback. Incorporating richer signals for discovering active regions or intent may enable better grounding and robustness in real-world applications.

Finally, deploying these methods in real-world systems presents additional challenges related to generalization, safety, and interactive learning. Extending current models to operate in-the-wild and adapt to dynamic feedback from users or the environment is a critical step toward building truly intelligent embodied agents.

7.2 Conclusion

This thesis presents a unified perspective on structured alignment as a central mechanism for advancing both visual understanding and robot decision-making. Starting from semantic correspondence with sparse supervision, we propose SCorrSAN to enable dense pixel-level matching by leveraging contextual priors and teacher-student learning. Building on this foundation, we address instance-level alignment in videos through PointVIS and UVIS, which significantly reduce annotation requirements while maintaining high segmentation performance. We further explore temporal relational structure in videos via Trokens, introducing semantic-aware relational trajectory tokens that enable few-shot action recognition by modeling structured object interactions over time. Finally, we extend alignment principles to robotics via ARDuP, which introduces active region-guided video diffusion for policy generation, and TREND, a tri-teaching framework that enhances preference-based reinforcement learning under noisy supervision.

Collectively, these contributions establish structured alignment as a powerful and unifying concept for modeling correspondences across visual and action domains. Beyond reducing re-

liance on supervision, our work improves generalization across domains, enhances interpretability, and enables data-efficient learning. The methods proposed in this thesis demonstrate strong empirical results across multiple benchmarks including SPair-71k, YouTube-VIS, CLIPort, and Meta-World. More broadly, they provide a principled framework for connecting perception and control, and offer concrete steps toward building intelligent systems capable of reasoning across time, modality, and abstraction level. By aligning structure across modalities and domains, these approaches open new possibilities for reliable, scalable solutions in autonomous robotics, video understanding, human-robot interaction, and intelligent surveillance. The advances in this thesis pave the way for alignment-driven research in both foundational vision tasks and real-world embodied AI applications.

Bibliography

[1] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019.

[2] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[3] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016.

[4] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019.

[5] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *arXiv preprint arXiv:2102.01558*, 2021.

[6] Pulkit Kumar, Namitha Padmanabhan, Luke Luo, Sai Saketh Rambhatla, and Abhinav Shrivastava. Trajectory-aligned space-time tokens for few-shot action recognition. In *European Conference on Computer Vision*, pages 474–493. Springer, 2024.

[7] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[8] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2023.

[9] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2021.

[10] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. *arXiv preprint arXiv:2301.11320*, 2023.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[12] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35. Springer, 2024.

[13] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Changxin Gao, Yingya Zhang, Deli Zhao, and Nong Sang. Molo: Motion-augmented long-short contrastive learning for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18011–18021, 2023.

[14] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[15] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017.

[16] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.

[17] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2018.

[18] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2019.

[19] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

[20] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2020.

[21] Xin Li, Deng-Ping Fan, Fan Yang, Ao Luo, Hong Cheng, and Zicheng Liu. Probabilistic model distillation for semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021.

[22] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022.

[23] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[24] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[25] Abhishek Kumar Chauhan and Prashant Krishan. Moving object tracking using gaussian mixture model and optical flow. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(4), 2013.

[26] Amit Goldstein and Raanan Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics (TOG)*, 31(5):1–10, 2012.

[27] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017.

[28] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2019.

[29] Sangryul Jeon, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn. Parn: Pyramidal affine regression networks for dense semantic correspondence. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2018.

[30] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.

[31] Shiyi Lan, Zhiding Yu, Christopher Choy, Subhashree Radhakrishnan, Guilin Liu, Yuke Zhu, Larry S Davis, and Anima Anandkumar. Discobox: Weakly supervised instance segmentation and semantic correspondence from box supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021.

[32] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[33] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.

[34] Kevin Dale, Micah K Johnson, Kalyan Sunkavalli, Wojciech Matusik, and Hanspeter Pfister. Image restoration using online photo collections. In *Proceedings of the International Conference on Computer Vision(ICCV)*, 2009.

[35] Sangryul Jeon, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. Joint learning of semantic alignment and object landmark detection. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2019.

[36] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2022.

[37] Shuda Li, Kai Han, Theo W Costain, Henry Howard-Jenkins, and Victor Prisacariu. Correspondence networks with adaptive neighbourhood consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.

[38] Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N Sinha. Patchmatch-based neighborhood consensus for semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021.

[39] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[40] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[41] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016.

[42] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2018.

[43] Seungryong Kim, Stephen Lin, SANG RYUL JEON, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2018.

[44] Seungryong Kim, Dongbo Min, Somi Jeong, Sunok Kim, Sangryul Jeon, and Kwanghoon Sohn. Semantic attribute matching networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

[45] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.

[46] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021.

[47] Shuaiyi Huang, Qiuyue Wang, and Xuming He. Confidence-aware adversarial learning for self-supervised semantic matching. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2020.

[48] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Probabilistic warp consistency for weakly-supervised semantic correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022.

[49] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2020.

[50] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.

[51] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2017.

[52] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S Davis. Rethinking pseudo labels for semi-supervised object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence(AAAI)*, 2022.

[53] Bo He, Xitong Yang, Le Kang, Zhiyu Cheng, Xin Zhou, and Abhinav Shrivastava. Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022.

[54] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[55] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In *Proceedings of the International Conference on Learning Representations(ICLR)*, 2015.

[56] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017.

[57] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

[58] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.

[59] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[60] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2019.

[61] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2018.

[62] Bo He, Xitong Yang, Zuxuan Wu, Hao Chen, Ser-Nam Lim, and Abhinav Shrivastava. Gta: Global temporal attention for video action understanding. *Proceedings of the British Machine Vision Conference(BMVC)*, 2020.

[63] Songyang Zhang, Xuming He, and Shipeng Yan. Latentgnn: Learning efficient non-local relations for visual recognition. In *Proceedings of the International Conference on Machine Learning(ICML)*, 2019.

[64] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016.

[65] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2018.

[66] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Annual Conference on Learning Theory(COLT)*, 1998.

[67] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *Proceedings of the International Conference on Computer Vision(ICCV)*, 2021.

[68] Dongyang Zhao, Ziyang Song, Zhenghao Ji, Gangming Zhao, Weifeng Ge, and Yizhou Yu. Multi-scale matching networks for semantic correspondence. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2021.

[69] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[70] 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(CVPR)*, 2016.

[71] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[72] Sangryul Jeon, Dongbo Min, Seungryong Kim, Jihwan Choe, and Kwanghoon Sohn. Guided semantic flow. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2020.

[73] Kai Han, Rafael S. Rezende, Bumsub Ham, Kwan-Yee K. Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *Proceedings of the International Conference on Computer Vision(ICCV)*, 2017.

[74] Qing Liu, Vignesh Ramanathan, Dhruv Mahajan, Alan Yuille, and Zhenheng Yang. Weakly supervised instance segmentation for videos with temporal mask consistency. In *CVPR*, 2021.

[75] Yang Fu, Sifei Liu, Umar Iqbal, Shalini De Mello, Humphrey Shi, and Jan Kautz. Learning to track instances without video annotations. In *CVPR*, 2021.

[76] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. *arXiv preprint arXiv:2208.02245*, 2022.

[77] Shuaiyi Huang, De-An Huang, Zhiding Yu, Shiyi Lan, Subhashree Radhakrishnan, Jose M Alvarez, Abhinav Shrivastava, and Anima Anandkumar. What is point supervision worth in video instance segmentation? *arXiv preprint arXiv:2404.01990*, 2024.

[78] Shuaiyi Huang, Saksham Suri, Kamal Gupta, Sai Saketh Rambhatla, Ser-nam Lim, and Abhinav Shrivastava. Uvis: Unsupervised video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2682–2692, 2024.

[79] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *ECCV*, 2022.

[80] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021.

[81] Yang Fu, Linjie Yang, Ding Liu, Thomas S Huang, and Humphrey Shi. Compfeat: Comprehensive feature aggregation for video instance segmentation. *arXiv preprint arXiv:2012.03400*, 2020.

[82] Kira Maag, Matthias Rottmann, Serin Varghese, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Improving video instance segmentation by light-weight temporal uncertainty estimates. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[83] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 1–18. Springer, 2020.

[84] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia. Video instance segmentation with a propose-reduce paradigm. *arXiv preprint arXiv:2103.13746*, 2021.

[85] Bo He, Xitong Yang, Hanyu Wang, Zuxuan Wu, Hao Chen, Shuaiyi Huang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Towards scalable neural representation for diverse videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6132–6142, 2023.

[86] Tianfei Zhou, Jianwu Li, Xueyi Li, and Ling Shao. Target-aware object discovery and association for unsupervised video multi-object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6985–6994, 2021.

[87] Minghan Li, Shuai Li, Lida Li, and Lei Zhang. Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11215–11224, 2021.

[88] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, pages 88–104. Springer, 2020.

[89] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild". In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5359–5368, 2019.

[90] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Computer Vision and Pattern Regognition (CVPR)*, 2019.

[91] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[92] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

[93] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[94] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. *NeurIPS*, 2021.

[95] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021.

[96] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[97] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. *arXiv*, 2021.

[98] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICVV*, 2017.

[99] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017.

[100] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. *arXiv preprint arXiv:1506.06204*, 2015.

[101] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European conference on computer vision*, pages 75–91. Springer, 2016.

[102] Hexiang Hu, Shiyi Lan, Yuning Jiang, Zhimin Cao, and Fei Sha. Fastmask: Segment multi-scale object candidates in one shot. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–999, 2017.

[103] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019.

[104] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020.

[105] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12193–12202, 2020.

[106] Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. Mask encoding for single shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10226–10235, 2020.

[107] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 282–298. Springer, 2020.

[108] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[109] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9816–9825, 2021.

[110] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12352–12361, 2021.

[111] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *ICCV*, 2021.

[112] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3800, 2018.

[113] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019.

[114] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *CVPR*, 2021.

[115] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. *Advances in Neural Information Processing Systems*, 32:6586–6597, 2019.

[116] Aditya Arun, CV Jawahar, and M Pawan Kumar. Weakly supervised instance segmentation by learning annotation consistent instances. In *European Conference on Computer Vision*, pages 254–270. Springer, 2020.

[117] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020.

[118] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020.

[119] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021.

[120] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Ying Shan, Bin Feng, and Wenyu Liu. Tracking instances as queries. *arXiv preprint arXiv:2106.11963*, 2021.

[121] Xuehui Yu, Pengfei Chen, Di Wu, Najmul Hassan, Guorong Li, Junchi Yan, Humphrey Shi, Qixiang Ye, and Zhenjun Han. Object localization under single coarse point supervision. *arXiv preprint arXiv:2203.09338*, 2022.

[122] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. Points as queries: Weakly semi-supervised object detection by points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8823–8832, 2021.

[123] Pei Wang, Zhaowei Cai, Hao Yang, Gurumurthy Swaminathan, Nuno Vasconcelos, Bernt Schiele, and Stefano Soatto. Omni-detr: Omni-supervised object detection with transformers. *arXiv preprint arXiv:2203.16089*, 2022.

[124] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Proposal-based instance segmentation with point supervision. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2126–2130. IEEE, 2020.

[125] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Yukang Chen, Lu Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation with point-based supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[126] Junsong Fan, Zhaoxiang Zhang, and Tieniu Tan. Pointly-supervised panoptic segmentation. In *European Conference on Computer Vision*, pages 319–336. Springer, 2022.

[127] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309, 2022.

[128] Qin Liu, Meng Zheng, Benjamin Planche, Srikrishna Karanam, Terrence Chen, Marc Niethammer, and Ziyan Wu. Pseudoclick: Interactive image segmentation with click imitation. *arXiv preprint arXiv:2207.05282*, 2022.

[129] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021.

[130] Brian L Price, Bryan S Morse, and Scott Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *2009 IEEE 12th International Conference on Computer Vision*, pages 779–786. IEEE, 2009.

[131] Arnaud Benard and Michael Gygli. Interactive video object segmentation in the wild. *arXiv preprint arXiv:1801.00269*, 2017.

[132] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 2018.

[133] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Guided interactive video object segmentation using reliability-based attention maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7322–7330, 2021.

[134] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *CVPR*, 2022.

[135] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[136] Shusheng Yang, Xinggang Wang, Yu Li, Yuxin Fang, Jiemin Fang, Liu, Xun Zhao, and Ying Shan. Temporally efficient vision transformer for video instance segmentation. In *CVPR*, 2022.

[137] Zhuang Li, Leilei Cao, and Hongbin Wang. Limited sampling reference frame for mask-track r-cnn. In *ICCVW*, 2021.

[138] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[139] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020.

[140] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. *arXiv preprint arXiv:2012.09812*, 2020.

[141] Yao Wei, Yanchao Sun, Ruijie Zheng, Sai Vemprala, Rogerio Bonatti, Shuhang Chen, Ratnesh Madaan, Zhongjie Ba, Ashish Kapoor, and Shuang Ma. Is imitation all you need? generalized decision-making with dual-phase training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16221–16231, October 2023.

[142] Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[143] Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé III, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-taco: Pretraining multitask representation via temporal action-driven contrastive loss. *arXiv preprint arXiv:2402.06187*, 2024.

[144] Ruijie Zheng, Xiyao Wang, Huazhe Xu, and Furong Huang. Is model ensemble necessary? model-based RL via a single model with lipschitz regularized value function. In *The Eleventh International Conference on Learning Representations*, 2023.

[145] Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. COPlanner: Plan to roll out conservatively but to explore optimistically for model-based RL. In *The Twelfth International Conference on Learning Representations*, 2024.

[146] Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, Shuzhen Li, Yanjie Ze, Hal Daumé III, Furong Huang, and Huazhe Xu. Drm: Mastering visual reinforcement learning through dormant

ratio minimization. In *The Twelfth International Conference on Learning Representations*, 2024.

[147] Ruijie Zheng, Ching-An Cheng, Hal Daumé III, Furong Huang, and Andrey Kolobov. Prise: Learning temporal action abstractions as a sequence compression problem, 2024.

[148] Fengda Zhu, Yi Zhu, Xiaodan Liang, and Xiaojun Chang. Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*, 2021.

[149] Lei Ke, Martin Danelljan, Henghui Ding, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask-free video instance segmentation. *arXiv preprint arXiv:2303.15904*, 2023.

[150] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[151] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[152] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[153] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[154] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[155] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019.

[156] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[157] Lili Wei, Congyan Lang, Liqian Liang, Songhe Feng, Tao Wang, and Shidi Chen. Weakly supervised video object segmentation via dual-attention cross-branch fusion. *ACM Trans. Intell. Syst. Technol.*, 13(3), mar 2022.

[158] Yufei Wang, Yongjiang Hu, Alan Wee-Chung Liew, and Junhu Wang. Weakly supervised video object segmentation. In *TENCON 2018 - 2018 IEEE Region 10 Conference*, pages 0315–0320, 2018.

[159] Mingmin Zhen, Shiwei Li, Lei Zhou, Jiaxiang Shang, Haoan Feng, Tian Fang, and Long Quan. Learning discriminative feature with crf for unsupervised video object segmentation. In *European Conference on Computer Vision*, 2020.

[160] Minhyeok Lee, Suhwan Cho, Seung-Hyun Lee, Chaewon Park, and Sangyoun Lee. Unsupervised video object segmentation via prototype memory network. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5913–5923, 2022.

[161] Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5140–5149, 2023.

[162] Ren Sucheng, Liu Wenxi, Liu Yongtuo, Chen Haoxin, Han Guoqiang, and He Shengfeng. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[163] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021.

[164] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, June 2022.

[165] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[166] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *ArXiv*, abs/2211.14813, 2022.

[167] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, 2021.

[168] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models. In *CVPRW*, 2023.

[169] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[170] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[171] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[172] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[173] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[174] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[175] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, 2022.

[176] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022.

[177] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. November 2021.

[178] Sai Saketh Rambhatla, Ishan Misra, Rama Chellappa, and Abhinav Shrivastava. Most: Multiple object localization with self-supervised transformers for object discovery, 2023.

[179] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[180] Issam H Laradji, David Vazquez, and Mark Schmidt. Where are the masks: Instance segmentation with image-level supervision. *arXiv preprint arXiv:1907.01430*, 2019.

[181] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[182] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below.

[183] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[184] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[185] Janez Perš, Vildana Sulić, Matej Kristan, Matej Perše, Klemen Polanec, and Stanislav Kovačič. Histograms of optical flow for efficient representation of body motion. *Pattern Recognition Letters*, 31(11):1369–1376, 2010.

[186] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[187] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

[188] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103:60–79, 2013.

[189] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[190] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *European Conference on Computer Vision*, pages 306–325. Springer, 2024.

[191] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[192] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[193] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021.

[194] Maxime Oquab, Timoth'ee Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

[195] Mohammad Abdelaziz Gowayyed, Marwan Torki, Mohammed Elsayed Hussein, and Motaz El-Saban. Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition. In *Twenty-third international joint conference on artificial intelligence*, 2013.

[196] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017.

[197] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[198] Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012.

[199] Hilde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.

[200] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[201] Orit Kliper-Gross, Tal Hassner, and Lior Wolf. One shot similarity metric learning for action recognition. In *Similarity-Based Pattern Recognition: First International Workshop, SIMBAD 2011, Venice, Italy, September 28-30, 2011. Proceedings 1*, pages 31–45. Springer, 2011.

[202] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[203] Xiang Wang, Zhiwu Qing, Ziyuan Huang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Changxin Gao, and Nong Sang. Proposal relation network for temporal action detection. *arXiv preprint arXiv:2106.11812*, 2021.

[204] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for semi-supervised temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1905–1914, 2021.

[205] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627, 2020.

[206] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 475–484, 2021.

[207] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Mingqian Tang, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. Hybrid relation guided set matching for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19948–19957, 2022.

[208] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. Hyrsm++: Hybrid relation guided temporal set matching for few-shot action recognition. *Pattern Recognition*, 147:110110, 2024.

[209] Sipeng Zheng, Shizhe Chen, and Qin Jin. Few-shot action recognition with hierarchical matching and contrastive learning. In *European Conference on Computer Vision*, pages 297–313. Springer, 2022.

[210] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint arXiv:1907.09021*, 2019.

[211] Songyang Zhang, Jiale Zhou, and Xuming He. Learning implicit temporal alignment for few-shot video classification. *arXiv preprint arXiv:2105.04823*, 2021.

[212] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 525–542. Springer, 2020.

[213] Xiao Wang, Weirong Ye, Zhongang Qi, Xun Zhao, Guangge Wang, Ying Shan, and Hanzi Wang. Semantic-guided relation propagation network for few-shot action recognition. In

Proceedings of the 29th ACM International Conference on Multimedia, pages 816–825, 2021.

[214] Xinzhe Ni, Yong Liu, Hao Wen, Yatai Ji, Jing Xiao, and Yujiu Yang. Multi-modal prototype-enhanced network for few-shot action recognition. *arXiv preprint arXiv:2212.04873*, 2022.

[215] Yuqian Fu, Li Zhang, Junke Wang, Yanwei Fu, and Yu-Gang Jiang. Depth guided adaptive meta-fusion network for few-shot video recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1142–1151, 2020.

[216] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[217] Anirudh Thatipelli, Sanath Narayan, Salman Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Bernard Ghanem. Spatio-temporal relation modeling for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19958–19967, 2022.

[218] Jiazheng Xing, Mengmeng Wang, Boyu Mu, and Yong Liu. Revisiting the spatial and temporal modeling for few-shot action recognition. In *AAAI Conference on Artificial Intelligence*, 2023.

[219] Jiamin Wu, Tianzhu Zhang, Zhe Zhang, Feng Wu, and Yongdong Zhang. Motion-modulated temporal fragment alignment network for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9160, 2022.

[220] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–766, 2018.

[221] Linchao Zhu and Yi Yang. Label independent memory for semi-supervised few-shot video classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):273–285, 2020.

[222] Jiazheng Xing, Mengmeng Wang, Yudi Ruan, Bofan Chen, Yaowei Guo, Boyu Mu, Guang Dai, Jingdong Wang, and Yong Liu. Boosting few-shot action recognition with graph-guided hybrid matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1740–1750, 2023.

[223] Xiang Wang, Shiwei Zhang, Jun Cen, Changxin Gao, Yingya Zhang, Deli Zhao, and Nong Sang. Clip-guided prototype modulating for few-shot action recognition. *International Journal of Computer Vision*, 132(6):1899–1912, 2024.

[224] Cong Wu, Xiao-Jun Wu, Linze Li, Tianyang Xu, Zhenhua Feng, and Josef Kittler. Efficient few-shot action recognition via multi-level post-reasoning. In *European Conference on Computer Vision*, pages 38–56. Springer, 2024.

[225] Congqi Cao, Yueran Zhang, Qinyi Lv, Lingtong Min, and Yanning Zhang. Exploring the adaptation strategy of clip for few-shot action recognition. In *Proceedings of the 1st International Workshop on Efficient Multimedia Computing under Limited*, pages 39–48, 2024.

[226] Fuqin Deng, Jiaming Zhong, Nannan Li, Lanhui Fu, Bingchun Jiang, Yi Ningbo, Feng Qi, He Xin, and Tin Lun Lam. Text-guided graph temporal modeling for few-shot video classification. *Engineering Applications of Artificial Intelligence*, 137:109076, 2024.

[227] Yikang Wang, Fei Guo, Li Zhu, and Yuan Guo. Sfmm: Semantic-to-frame matching with multi-classifier for few-shot action recognition. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[228] Bozheng Li, Mushui Liu, Gaoang Wang, and Yunlong Yu. Frame order matters: A temporal sequence-aware model for few-shot action recognition. *arXiv preprint arXiv:2408.12475*, 2024.

[229] Fei Guo, YiKang Wang, Han Qi, Wenping Jin, Li Zhu, and Jing Sun. Multi-view distillation based on multi-modal fusion for few-shot action recognition (clip-mdmf). *Knowledge-Based Systems*, 304:112539, 2024.

[230] Yutao Tang, Benjamín Béjar, and René Vidal. Semantic-aware video representation for few-shot action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6458–6468, 2024.

[231] Congqi Cao, Yueran Zhang, Yating Yu, Qinyi Lv, Lingtong Min, and Yanning Zhang. Task-adapter: Task-specific adaptation of image models for few-shot action recognition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9038–9047, 2024.

[232] Wenbo Huang, Jinghui Zhang, Guang Li, Lei Zhang, Shuoyuan Wang, Fang Dong, Jiahui Jin, Takahiro Ogawa, and Miki Haseyama. Manta: Enhancing mamba for few-shot action recognition of long sub-sequence. *arXiv preprint arXiv:2412.07481*, 2024.

[233] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.

[234] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022.

[235] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *arXiv preprint arXiv:2306.08637*, 2023.

[236] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023.

[237] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. *arXiv preprint arXiv:2312.00786*, 2023.

[238] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. *arXiv preprint arXiv:2306.05422*, 2023.

[239] Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. Dino-tracker: Taming dino for self-supervised point tracking in a single video, 2024.

[240] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1430–1439, 2018.

[241] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6566–6575, 2018.

[242] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 345–362. Springer, 2020.

[243] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 352–361, 2020.

[244] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023.

[245] Haiam A Abdul-Azim and Elsayed E Hemayed. Human action recognition using trajectory-based representation. *Egyptian Informatics Journal*, 16(2):187–198, 2015.

[246] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 102–106, 2016.

[247] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[248] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[249] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022.

[250] Sai Saketh Rambhatla, Ishan Misra, Rama Chellappa, and Abhinav Shrivastava. Most: Multiple object localization with self-supervised transformers for object discovery. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15777–15788, 2023.

[251] Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3124–3134, 2023.

[252] Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22755–22764, 2023.

[253] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[254] Mohammad Abdelaziz Gowayyed, Marwan Torki, Mohamed Elsayed Hussein, and Motaz El-Saban. Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition. In *IJCAI*, volume 1, pages 1351–1357, 2013.

[255] Khoi D Nguyen, Quoc-Huy Tran, Khoi Nguyen, Binh-Son Hua, and Rang Nguyen. Inductive and transductive few-shot video classification via appearance and temporal alignments. In *European Conference on Computer Vision*, pages 471–487. Springer, 2022.

[256] Yifei Huang, Lijin Yang, and Yoichi Sato. Compound prototype matching for few-shot action recognition. In *European Conference on Computer Vision*, pages 351–368. Springer, 2022.

[257] Haifeng Xia, Kai Li, Martin Renqiang Min, and Zhengming Ding. Few-shot video classification via representation fusion and promotion learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19311–19320, 2023.

[258] Xiao Wang, Yan Yan, Hai-Miao Hu, Bo Li, and Hanzi Wang. Cross-modal contrastive learning network for few-shot action recognition. *IEEE Transactions on Image Processing*, 2024.

[259] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, brian ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024.

[260] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

[261] Shuaiyi Huang, Mara Levy, Zhenyu Jiang, Anima Anandkumar, Yuke Zhu, Linxi Fan, De-An Huang, and Abhinav Shrivastava. Ardup: Active region video diffusion for universal policies. *arXiv preprint arXiv:2406.13301*, 2024.

[262] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2847–2854. IEEE, 2012.

[263] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023.

[264] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[265] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[266] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems*, 36, 2024.

[267] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

[268] Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024.

[269] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[270] Dima Damen, Teesid Leelasawassuk, and Walterio Mayol-Cuevas. You-do, i-learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance. *Computer Vision and Image Understanding*, 149:98–112, 2016.

[271] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 287–295, 2015.

[272] Sanket Thakur, Cigdem Beyan, Pietro Morerio, Vittorio Murino, and Alessio Del Bue. Leveraging next-active objects for context-aware anticipation in egocentric videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8657–8666, 2024.

[273] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[274] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[275] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[276] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[277] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[278] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

[279] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[280] Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[281] Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[282] Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

[283] Mara Levy, Nirat Saini, and Abhinav Shrivastava. Wayex: Waypoint exploration using a single demonstration. In *International Conference on Robotics and Automaction(ICRA)*, 2024.

[284] Patrick M Pilarski, Michael R Dawson, Thomas Degris, Farbod Fahimi, Jason P Carey, and Richard S Sutton. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *2011 IEEE international conference on rehabilitation robotics*, pages 1–7. IEEE, 2011.

[285] Aaron Wilson, Alan Fern, and Prasad Tadepalli. A bayesian approach for policy learning from trajectory preference queries. *Advances in neural information processing systems*, 25, 2012.

[286] Christian Wirth, Johannes Fürnkranz, and Gerhard Neumann. Model-free preference-based reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[287] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024.

[288] Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[289] Jie Cheng, Gang Xiong, Xingyuan Dai, Qinghai Miao, Yisheng Lv, and Fei-Yue Wang. Rime: Robust preference-based reinforcement learning with noisy preferences. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[290] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

[291] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[292] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

[293] Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Conference on robot learning*, pages 519–528. PMLR, 2018.

[294] Erdem Bıyık, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*, 2020.

[295] Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning, 2022.

[296] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153, 2022.

[297] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 373–381, 2021.

[298] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*, 2022.

[299] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

[300] Wanqi Xue, Bo An, Shuicheng Yan, and Zhongwen Xu. Reinforcement learning from diverse human preferences. *arXiv preprint arXiv:2301.11774*, 2023.

[301] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[302] Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. Lmpriors: Pre-trained language models as task-specific priors. *arXiv preprint arXiv:2210.12530*, 2022.

[303] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pages 8657–8677. PMLR, 2023.

[304] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

[305] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. Structured attentions for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1291–1300, 2017.

[306] Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback, 2023.

[307] Mara Levy, Siddhant Haldar, Lerrel Pinto, and Abhinav Shirivastava. P3-po: Prescriptive point priors for visuo-spatial generalization of robot policies. In *International Conference on Robotics and Automaction(ICRA)*, 2024.

[308] Xiyang Wu, Tianrui Guan, Dianqi Li, Shuaiyi Huang, Xiaoyu Liu, Xijun Wang, Ruiqi Xian, Abhinav Shrivastava, Furong Huang, Jordan Lee Boyd-Graber, et al. Autohallusion: Automatic generation of hallucination benchmarks for vision-language models. *arXiv preprint arXiv:2406.10900*, 2024.

[309] Ruijie Zheng, Ching-An Cheng, Hal Daumé III, Furong Huang, and Andrey Kolobov. PRISE: LLM-style sequence compression for learning temporal action abstractions in control. In *Forty-first International Conference on Machine Learning*, 2024.

[310] Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[311] Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning, 2024.

[312] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. *arXiv preprint arXiv:2303.00001*, 2023.

[313] Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

[314] Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22270–22284, 2022.

[315] Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Preference transformer: Modeling human preferences using transformers for rl. *arXiv preprint arXiv:2303.00957*, 2023.

[316] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[317] Guoqing Chao, Kaiwen Zhang, Xiru Wang, and Dianhui Chu. Three-teaching: A three-way decision framework to handle noisy labels. *Applied Soft Computing*, 154:111400, 2024.

[318] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[319] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

[320] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference*

on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

[321] Gemini Team. Gemini: A family of highly capable multimodal models, 2024.