# ABSTRACT

Title of Dissertation: TOWARDS UNIFYING MULTIMODAL: PERCEPTION, REASONING AND GENERATION

Jiuhai Chen
Doctor of Philosophy, 2026

Dissertation Directed by: Professor Tianyi Zhou and Professor Tom Goldstein
Department of Computer Science

This dissertation studies how to build a unified multimodal model that supports visual perception, multimodal reasoning, and visual generation. We present a sequence of models that progressively expand the frontier of unified multimodal learning, grounded in new architectures, training recipes, and open-source datasets.

Firstly, we introduce Florence-VL for visual perception, a multimodal family built on Florence-2's generative vision encoder. Unlike conventional vision encoder models CLIP, Florence-2 provides rich, multi-level visual features. We fuse these features via a novel depth-breadth fusion architecture and train in two stages: end-to-end pretraining followed by instruction tuning. Florence-VL's enriched embeddings yield state-of-the-art results across VQA, OCR, chart understanding, and knowledge-intensive benchmarks, and all code and weights are open-sourced.

Next, we introduce BLIP3-o, a unified foundation model for both image understanding and image generation. Unifying image understanding and generation has gained growing attention in recent research on multimodal models. Although design choices for image understanding have been

extensively studied, the optimal model architecture and training recipe for a unified framework with image generation remain underexplored. Motivated by the strong potential of autoregressive and diffusion models for high-quality generation and scalability, we conduct a comprehensive study of their use in unified multimodal settings, with emphasis on image representations, modeling objectives, and training strategies. Grounded in these investigations, we introduce a novel approach that employs a diffusion transformer to generate semantically rich CLIP image features, in contrast to conventional VAE-based representations. This design yields both higher training efficiency and improved generative quality. Furthermore, we demonstrate that a sequential pretraining strategy for unified models—first training on image understanding and subsequently on image generation—offers practical advantages by preserving image understanding capability while developing strong image generation ability. Finally, we carefully curate a high-quality instruction-tuning dataset BLIP3o-60k for image generation by prompting GPT-Image with a diverse set of captions covering various scenes, objects, human gestures, and more. Building on our innovative model design, training recipe, and datasets, we develop BLIP3-o, a suite of state-of-the-art unified multimodal models. BLIP3-o achieves superior performance across most of the popular benchmarks spanning both image understanding and generation tasks. *To facilitate future research, we fully open-source our models, including code, model weights, training scripts, and pretraining and instruction tuning datasets.*

Lastly, we present BLIP3o-NEXT, a fully open-source foundation model in the BLIP3 series that advances the next frontier of native image generation. BLIP3o-NEXT unifies text-to-image generation and image editing within a single architecture, demonstrating strong image generation and image editing capabilities. In developing the state-of-the-art native image generation model, we identify four key insights: (1) Most architectural choices yield comparable performance; an architecture can be deemed effective provided it scales efficiently and supports fast inference; (2) The successful application of

reinforcement learning can further push the frontier of native image generation; (3) Image editing still remains a challenging task, yet instruction following and the consistency between generated and reference images can be significantly enhanced through post-training and data engine; (4) Data quality and scale continue to be decisive factors that determine the upper bound of model performance. Building upon these insights, BLIP3o-NEXT leverages an Autoregressive + Diffusion architecture in which an autoregressive model first generates discrete image tokens conditioned on multimodal inputs, whose hidden states are then used as conditioning signals for a diffusion model to generate high-fidelity images. This architecture integrates the reasoning strength and instruction following of autoregressive models with the fine-detail rendering ability of diffusion models, achieving a new level of coherence and realism. Extensive evaluations of various text-to-image and image-editing benchmarks show that BLIP3o-NEXT achieves superior performance over existing models.

Overall, this dissertation contributes architectures, training strategies, and open-source resources that move toward a unified multimodal model family capable of visual perception, generation, and reasoning.

# TOWARDS UNIFYING MULTIMODAL: PERCEPTION, REASONING AND GENERATION

by

Jiuhai Chen

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
2026

Advisory Committee:
Professor Tom Goldstein, Chair/Co-Advisor
Professor Tianyi Zhou, Advisor
Professor Furong Huang
Professor Ruohan Gao
Professor Haizhao Yang

# Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I'd like to thank my advisor, Professor Tianyi Zhou for giving me an invaluable opportunity to work on challenging and extremely interesting projects over the past four years. He has always made himself available for help and advice and there has never been an occasion when I've knocked on his door and he hasn't given me time. It has been a pleasure to work with and learn from such an extraordinary individual.

I would also like to thank my co-advisor, Professor Tom Goldstein. Without his extraordinary advice, this thesis would have been a distant dream. Thanks are due to Professor Furong Huang, Professor Ruohan Gao and Professor Haizhao Yang for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript.

My PhD work would not have been possible without the collaboration and contributions of many talented individuals. I deeply appreciate the engaging discussions and enjoyable research projects shared with Zhiyang Xu, Xichen Pan, Jonas Mueller, Bin Xiao, David Wipf, Le Xue and Lichang Chen.

I owe my deepest thanks to my family - my mother and father who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times. Words cannot express the gratitude I owe them. It is impossible to remember all, and I apologize to those I've inadvertently left out. Lastly, thank you all!

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

## 1.1 Problem Formulation

Modern multimodal foundation models are expected to do more than recognize visual content: they can *perceive* visual signals, *reason* over multimodal inputs, and *generate* or *edit* visual content in a controllable way. This dissertation studies how to build a unified multimodal model that supports visual perception, multimodal reasoning, and visual generation, and presents a sequence of models that progressively expand the frontier of unified multimodal learning, grounded in new architectures, training recipes, and open-source datasets.

We formalize the unified multimodal learning goal as developing a model family that can: (1) transfer image + text context into representations sufficient for accurate and faithful multimodal reasoning, and (2) map multimodal inputs into high-quality images (and edited images), while preserving core image understanding capability. We pursue this goal via architectures and training strategies that strengthen perceptual representations, unify understanding and generation in a coherent framework, and leverage post-training (including reinforcement learning) to push generation and editing quality.

## 1.2 Research Questions

This dissertation is guided by the following research questions:

1. **Feature extraction and fusion:** Can a single vision foundation model generate diverse, task-relevant perceptual features, and how should we fuse and align them with language-model token spaces?

2. **Perceptual representation:** How can we design visual representations that preserve both global semantics and fine-grained details while remaining efficient for training and inference?

3. **Unified generation design:** In unified multimodal models, what architecture and image representation lead to scalable high-quality image generation ? In particular, when should we model low-level latents (e.g., VAE) versus higher-level semantic features (e.g., CLIP features)?

4. **Training strategy:** How should we train unified models so that image generation ability improves *without degrading* image understanding capability?

5. **Post-training for image generation/editing:** How far can post-training (e.g., instruction tuning and reinforcement learning) push image generation and image editing, and what are the key bottlenecks?

## 1.3 Thesis Roadmap

This dissertation answers the research questions through three stages of model development:

- **Stage 1: Florence-VL (Visual Perception).** Built on the generative vision encoder of Florence-2, this model addresses **RQ1** (feature extraction and fusion) and **RQ2** (perceptual representation). We investigate if a single vision model can generate diverse task-specific features and how to align these with language-model token spaces. Stronger perception is a prerequisite for

robust multimodal reasoning.Multi-level visual features and the Depth-Breadth Fusion (DBFusion) strategy preserve fine-grained details required for OCR and grounding.

- **Stage 2: BLIP3-o (Unified Multimodal Architecture).** Addressing **RQ3** (unified generation design) and **RQ4** (training strategy), BLIP3-o unifies image understanding and generation in a coherent framework. We conduct a systematic exploration of architecture and representation, comparing VAE latents against high-level semantic CLIP features and joint versus sequential training. CLIP image features offer more compact and informative representations, resulting in higher training efficiency. Furthermore, a sequential pretraining strateg (understanding first, then generation) preserves image understanding capability while developing strong generation ability.

- **Stage 3: BLIP3o-NEXT (Native Image Generation & Editing).** This stage targets **RQ5** (post-training for generation/editing) to advance the frontier of native image synthesis and editing. We examine how far post-training, including instruction tuning and Reinforcement Learning (RL), can push the quality and consistency of image generation and editing. Reinforcement learning (specifically GRPO) and high-quality data engines significantly push the frontiers of instruction following and text rendering. While most architectural choices yield comparable results if they scale efficiently, data quality remains a decisive factor for performance.

**Differences and Connections:** three stages form a progressive hierarchy where Florence-VL provides the foundational perception required for reasoning. BLIP3-o establishes a stable unified bridge using compatible semantic representations. Finally, BLIP3o-NEXT refines this unification with advanced post training to achieve high-fidelity synthesis and controllable editing. While Florence-VL focuses on feature fusion, BLIP3-o focuses on training recipes, and BLIP3o-NEXT focuses on quality

driven posttraining.

Overall, this dissertation contributes architectures, training strategies, and open-source resources that move toward a unified multimodal model family capable of visual perception, reasoning, and generation.

# Chapter 2: Florence-VL: Enhancing Vision-Language Models with Generative Vision Encoder and Depth-Breadth Fusion

## 2.1 Introduction

Recent progress in multimodal large language models (MLLMs) are largely driven by progress in large language models [2, 3]. However, when it comes to visual encoders, transformer-based models like CLIP or SigLIP remain the most commonly used choices. Despite CLIP and SigLIP's effectiveness, they come with limitations; for instance, their last-layer features usually provide an image-level semantic representation that captures the overall scene and context, but often overlook pixel or region-level details and low-level features that are critical to various downstream tasks. There is a much broader range of visual representation, such as the self-supervised DINOv2 model [4], diffusion model [5] and segmentation [6], [7] shows these different visual encoders can benefit well in some specific tasks.

In order to leverage distinctive representations of multiple vision encoders, some recent works such as [7, 8] adopt a mixture of vision encoders that specialize in different feature aspects or skills. However, integrating multiple vision encoders increases the computational expense for both model training and deployment. *Could a single vision model be designed to generate distinct visual features, each emphasizing different perceptual information in the input image?* In this paper, we propose Florence-VL, which leverages the generative vision foundation model Florence-2 [9] as the vision encoder. Florence-2 offers a prompt-based representation for various computer vision tasks, including captioning, object detection, grounding, and OCR. Its versatile visual representations can benefit different types of downstream tasks. For instance, OCR-based representations are advantageous for

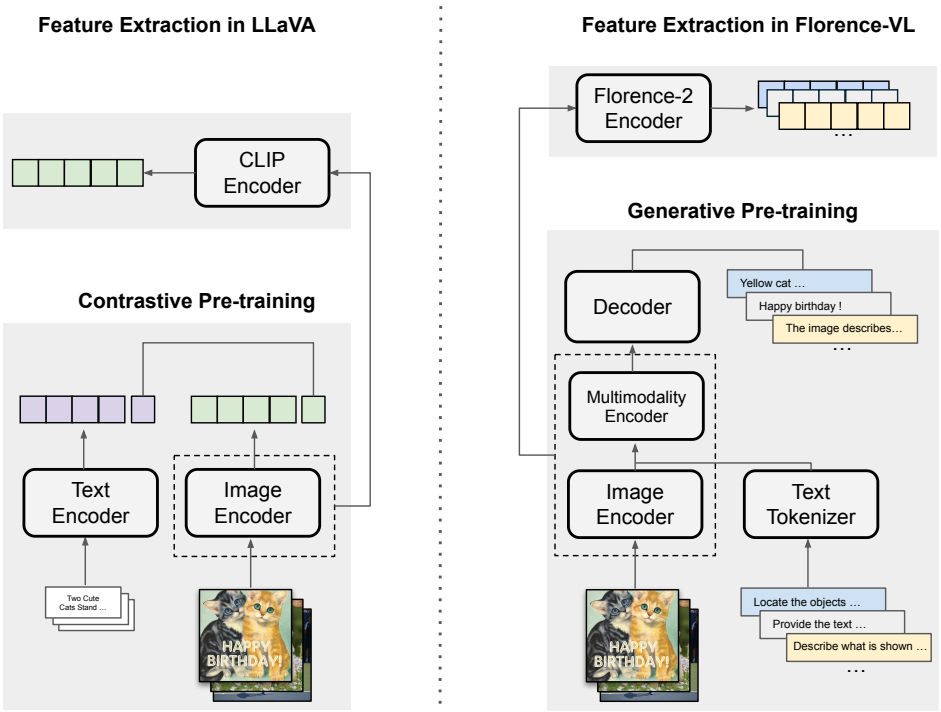



Figure 2.1: Comparison of LLaVA-style MLLMs with our Florence-VL. LLaVA-style models use CLIP, pretrained with contrastive learning, to generate a **single high-level image feature**. In contrast, Florence-VL leverages Florence-2, pretrained with generative modeling across various vision tasks such as image captioning, OCR, and grounding. This enables Florence-VL to flexibly extract **multiple task-specific image features** using Florence-2 as the image encoder.

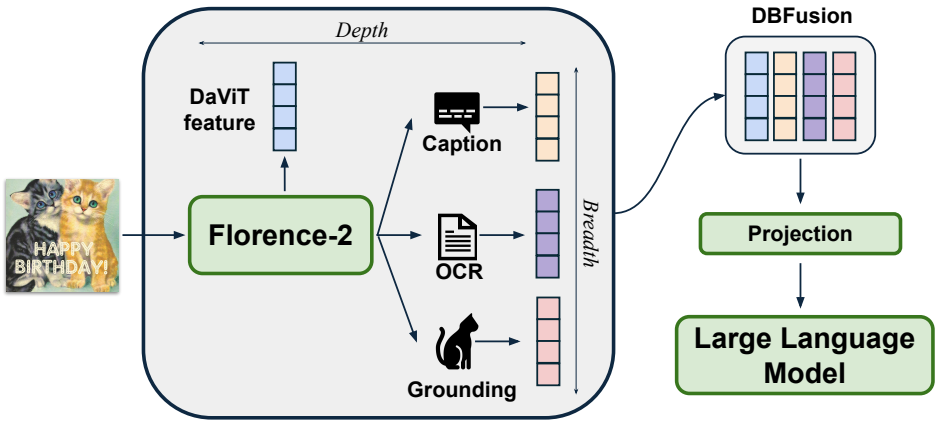



Figure 2.2: An overview of Florence-VL, which extracts visual features of different depths (levels of feature concepts) and breaths (prompts) from Florence-2, combines them using DBFusion, and project the fused features to an LLM's input space. Florence-VL is fully pretrained on image captioning data and then partially finetuned on instruction-tuning data.

tasks that require extracting textual information from images, and grounding-based representation can benefit for tasks that require the relationships between objects and their spatial contexts. However, to build a better MLLM, how to extract these diverse features and align them with a pretrained LLM remains underexplored.

To address this, we propose *Depth-Breadth Fusion* (**DBFusion**) to effectively selecting and utilizing diverse visual features. Visual features from different layers capture various levels of concepts, with the final layers typically representing higher-level concepts. Integrating lower-level features can therefore complement these high-level representations, which we refer to as the "Depth" of visual features. Additionally, since different downstream tasks need different perceptual information within images, a single image feature often falls short in capturing all relevant information. Thus, we leverage multiple image features, with each feature capturing different visual representations. We refer to this as the "Breadth" of visual features. For utilizing these diverse visual features, we find that a straightforward channel concatenation serves as a simple yet effective fusion strategy. Specifically,

we concatenate multiple features along the channel dimension, and these combined features, spanning various depths and breadths, are then projected as input embedding to LLMs.

We train Florence-VL on a novel recipe of open-sourced training data, which is composed of a large-scale detailed captioning dataset and a mix of instruction tuning datasets for whole-model pretraining and partial-model finetuning, respectively.

The resulted Florence-VL achieves significant advantages on 25 benchmarks covering vision-centric, knowledge-based, and OCR & Chart tasks, outperforming other advanced MLLMs like Cambrian [7]. Moreover, we provide quantitative analysis and visualization demonstrating that Florence-VL's visual representation achieves better alignment to LLMs than the widely adopted vision encoders such as CLIP and SigLIP [2].

## 2.2 Preliminary: Florence-2

Florence-2 [9] is a vision foundation model that utilizes a unified, prompt-based approach to handle various vision tasks with simple instructions, such as captioning, object detection, grounding, and segmentation. The architecture consists of a vision encoder DaViT [10] and a standard encoder-decoder model. It processes an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ (where $H$ and $W$ indicate height and width, respectively) into flattened visual token embeddings. The model then applies a standard encoder-decoder transformer architecture to process both visual and language token embeddings. It first generates prompt text embeddings $\mathbf{T} \in \mathbb{R}^{N_t \times D}$ using the language tokenizer and word embedding layer, with $N_t$ and $D$ representing the number and dimensionality of prompt tokens, respectively. The vision token embeddings are then concatenated with the prompt embeddings to create the input for the multi-modality encoder module, $\mathbf{X} = [\mathbf{V}, \mathbf{T}]$, where $\mathbf{V} \in \mathbb{R}^{N_v \times D}$ is produced by applying a linear

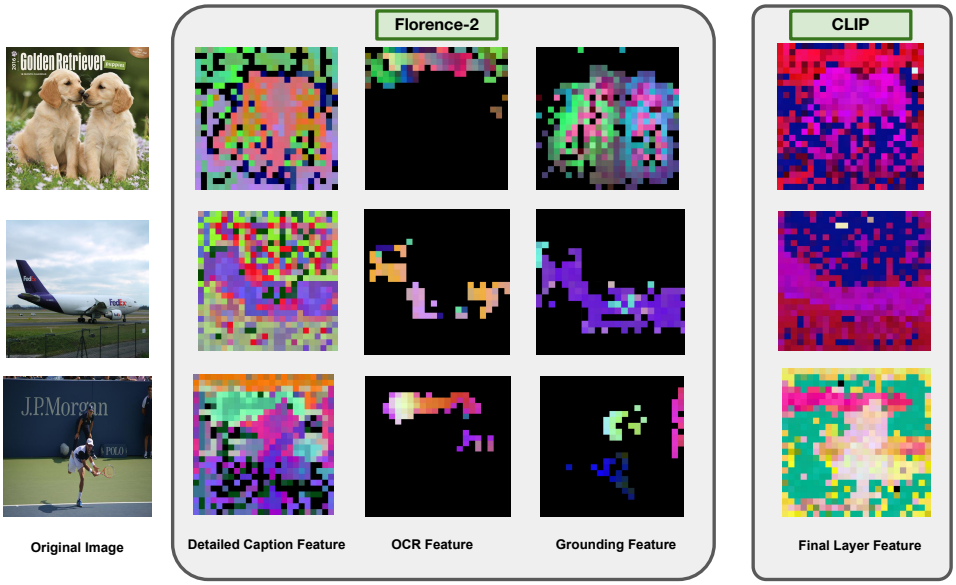



Figure 2.3: Visualization of the first three PCA components: we apply PCA to image features generated from Detailed Caption, OCR, and Grounding prompts, excluding the background by setting a threshold on the first PCA component. The image features derived from the Detailed Caption prompt (second column) capture the general context of the image, those from the OCR prompt (third column) focus primarily on text information, and those from the Grounding prompt (fourth column) highlight spatial relationships between objects. Additionally, we visualize the final layer features from OpenAI CLIP (ViT-L/14@336) in the last column, showing that CLIP features often miss certain region-level details, such as text information in many cases.

projection and LayerNorm layer to visual embedding from DaViT, with $N_v$ and $D$ representing the number and dimensionality of vision tokens, respectively. The linear projection and LayerNorm layer are used to ensure dimensionality alignment with $\mathbf{T}$. Encoder-decoder model will process the $\mathbf{X}$ and generate the desirable results, such as captions, object detections, grounding in textual form.

## 2.3 Method

### 2.3.1 Using Florence-2 as Vision Backbone

To address the limitations of existing vision backbones in MLLMs, specifically, last layer features typically yield an image-level representation that captures overall scene and context but often misses pixel- or region-level details, we utilize the vision foundation model Florence-2 as our visual encoder for extracting visual features. Unlike the CLIP pre-trained vision transformers that provide a single, universal image feature, Florence-2 can identify spatial details at different scales, by using different tasks prompts.

In MLLMs, effective image understanding requires capturing multiple levels of granularity, from global semantics to local details, and understanding spatial relationships between objects and entities within their semantic context. Florence-2, with its capability to manage diverse granularity levels, is an ideal vision encoder to address these core aspects of image comprehension. In the following section, we explore how to leverage Florence-2's strengths in integrating it into MLLMs.

### 2.3.2 Visual Features spanning Depth and Breadth

Breadth: since different downstream tasks require varying perceptual information from images, we consider expanding the breadth of visual representation. Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ and a

| | # Vis tok | MMBench (EN) | POPE | MM-Vet | MME-P | Seed-image | HallusionBench | LLaVA-bench | AI2D | MathVista | MMMU | OCRBench | ChartQA | DocVQA | InfoVQA | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Token Integration | 1728 | **66.6** | 88.7 | 34.1 | 1536.3 | **70.9** | 45.0 | 63.3 | 56.9 | **28.1** | **36.4** | 40.8 | 23.0 | 44.6 | **29.5** | 50.3 |
| Average Pooling | 576 | 65.7 | 88.8 | 32.3 | **1551.3** | 70.3 | 45.7 | 64.6 | 56.6 | 27.4 | 36.0 | 41.2 | **24.6** | **44.8** | 29.3 | 50.4 |
| Channel Integration | 576 | 66.1 | **89.4** | **35.2** | 1543.5 | 70.3 | **46.8** | **65.0** | **57.2** | 28.0 | 35.6 | **41.4** | 24.3 | 44.5 | 29.4 | **50.8** |

Table 2.1: Experiments for different fusion strategies. The vision token count is 1728 for token integration, which leads to longer training and inference times. The channel integration strategy shows better performance and training efficiency compared to the other two fusion methods.

task-specific prompt, such as "provide the text shown in the image", Florence-2 will process the image feature and prompt feature into $\mathbf{X} = [\mathbf{V}, \mathbf{T}]$ and then feed into the encoder-decoder transformer architecture. The encoder employs an attention mechanism to process $\mathbf{X}$, producing an output $\mathbf{X}' = [\mathbf{V}', \mathbf{T}']$. Due to the cross-attention between $\mathbf{V}$ and $\mathbf{T}$, the updated image feature $\mathbf{V}'$ becomes more focused on the prompt "provide the text shown in the image", specifically extracting more text information from the image.

We focus on three distinct tasks that contribute to image understanding, resulting in three different image embeddings $[\mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$, each tailored to a specific task:

- **Detailed Image Caption**: describe what is shown in the image with a paragraph. It enables the model to give a overall context of an image.
- **OCR**: provide the text shown in the image. It extracts more text information from the image.
- **Dense Region Caption**: locate the objects in the image, with their descriptions. It captures the spatial relationships between objects.

We visualize the image features with different task prompts, applying PCA to the visual embeddings and setting a threshold for the visualization. As illustrated in Figure 2.3, different image

embeddings emphasize distinct conceptual information within the images. Additionally, we also visualize the final layer image features from OpenAI CLIP in Figure 2.3, which often lacks certain region-level details in most cases.

Depth: we also integrate lower-level features using $\mathbf{V}$ from DaViT, combined with higher-level features $[\mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$ derived from the three prompts, allows us to capture multiple levels of conceptual detail.

### 2.3.3 Depth-Breadth Fusion

Since we have image feature with different level of granularity, feature fusion is commonly used. When dealing with multiple feature embeddings, such as $[\mathbf{V}, \mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$, the next question becomes how to fuse these features and align them with the language model space. To take advantage of all these four features, several approaches can be considered for this fusion process:

- **Token Integration**: This approach involves concatenating all features along the token dimension. However, this can make the visual token excessively long and complicate model training.
- **Average Pooling**: Alternatively, average pooling over all features can be used, but this method may result in information loss.
- **Channel Integration**: A more effective method is to concatenate features along the channel dimension, which does not increase the sequence length.

To quickly assess which feature fusion method provides the best overall performance, we use datasets from LLaVA-1.5 [2], which include 558K image captions for pre-training and 665K entries for instruction tuning. In the Table 2.1, the channel integration strategy shows better performance and

training efficiency compared to the other two fusion methods. Thus we choose channel integration simple yet effective fusion strategy.

### 2.3.4 Florence-VL

As shown in Figure 2.2, Florence-VL is composed of the vision foundation model Florence-2 and the large language model. After extracting multiple image features, we use MLP to project these features into the language model space. During the pretraining stage, we align Florence-2 with the language model using image detailed caption data. In the instruction tuning stage, we use diverse and high-quality instruction-tuning dataset to effectively adapt the model to downstream tasks.

## 2.4 Analysis on Different Vision Encoders

To demonstrate that Florence-2 is a superior vision encoder compared to others, we quantify the cross-modal alignment quality between various vision encoders and language models, allowing us to assess the impact of different vision encoders without requiring subsequent supervised fine-tuning and evaluations on benchmarks [11, 12]. Specifically, consider a pretrained MLLM $\mathscr{M} = (\mathscr{V}, \mathscr{L})$ where $\mathscr{V}$ is the vision encoder and $\mathscr{L}$ represents the language model, we input a set of image-text pairs, $(V,T) = (\{v_n\}_{n=1}^N, \{t_n\}_{n=1}^N)$, into the model. For the $n^{th}$ image-text pair, the vision encoder produces vision representations $f^{v_n} \in \mathbb{R}^{r_n \times d'}$, and the text representations $f^{t_n} \in \mathbb{R}^{s_n \times d}$ from last layer of the language decoder, where $r_n$ and $s_n$ are the number of tokens in the vision and text representations, and $d'$ and $d$ are the hidden state dimensions for the vision and text tokens. We apply the trainable projection $\mathscr{P}$ to $f^{v_n}$ to ensure dimensionality alignment with $f^{t_n}$, that is $\mathscr{P}(f^{v_n}) \in \mathbb{R}^{r_n \times d}$. We also apply average pooling along token dimension and normalize along the hidden dimension for both

$\mathscr{P}(f^{v_n})$ and $f^{t_n}$. For all image-text pairs, we concatenate all vision features along the first dimension to form a matrix $F^{v_n} \in \mathbb{R}^{N\times d}$, and similarly concatenate all text features into a matrix $F^{t_n} \in \mathbb{R}^{N\times d}$. Since we need to measure the modality gap between vision tokens and text tokens, we compute the divergence between these two token representations. Specifically, we optimize the trainable projection $\mathscr{P}$, which is used to bring these two representations closer together by minimizing a cross-entropy loss function:

$$\mathscr{L} = -\sum_{i,j} \mathscr{I}_n^{(i,j)} \log\left(\text{softmax}(F^{v_n} \times (F^{t_n})^T)_{i,j}\right)$$

, where $\mathscr{I}_n$ is the target (indicator) matrix. The multiplication of $F^{v_n}$ with the transpose of $F^{t_n}$ calculates the correlation between vision and text token representations. In short, the loss function is designed to minimize the distance between vision tokens and their corresponding text tokens by maximizing the likelihood that each vision token aligns correctly with its associated text token.

We use a set of image-text pairs $(V,T) = (\{v_n\}_{n=1}^N, \{t_n\}_{n=1}^N)$ from the LLaVA 1.5 pretraining image captioning datasets and select various vision encoders to assess how well we can optimize the alignment between the vision encoder and the language model. The vision encoders we evaluate include: Stable Diffusion [13], Dinov2 [4] (ViT-G/14, ViT-L/14, ViT-B/14), SigLIP, OpenAI CLIP, and our Florence-2 model. The chosen language model is Llama 3 8B Instruct. We plot the alignment loss in Figure 2.4, which clearly shows that Florence-2 vision encoder achieves the lowest alignment loss compared to the other vision encoders, demonstrating the best alignment with text embeddings. Additionally, SigLIP demonstrates competitive results, as noted in [7], which highlights SigLIP's strong benchmark performance relative to other vision encoders, aligning with the findings of our study.

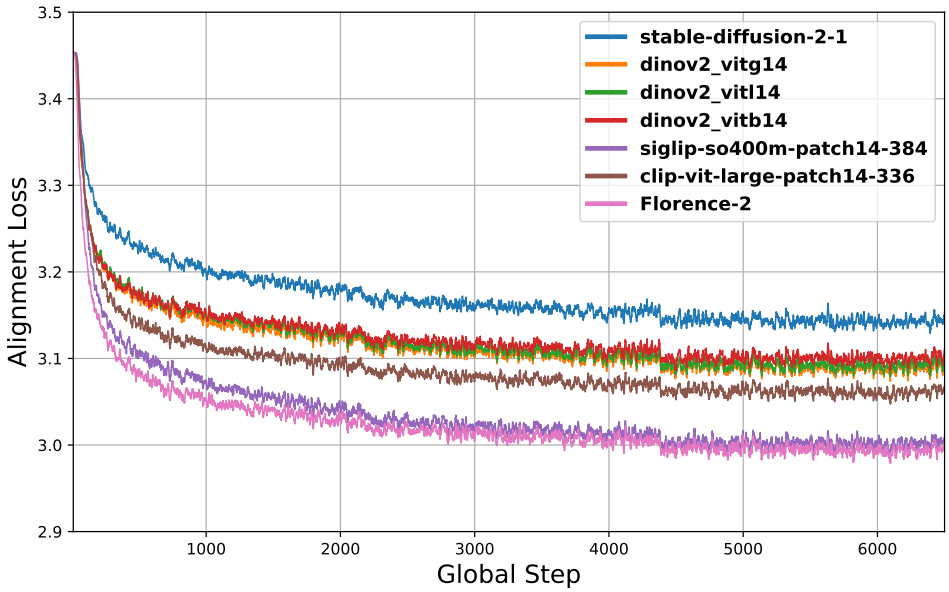



Figure 2.4: We plot the alignment loss for different vision encoders, which clearly shows that Florence-2 vision encoder achieves the lowest alignment loss compared to the other vision encoders, demonstrating the best alignment with text embeddings.

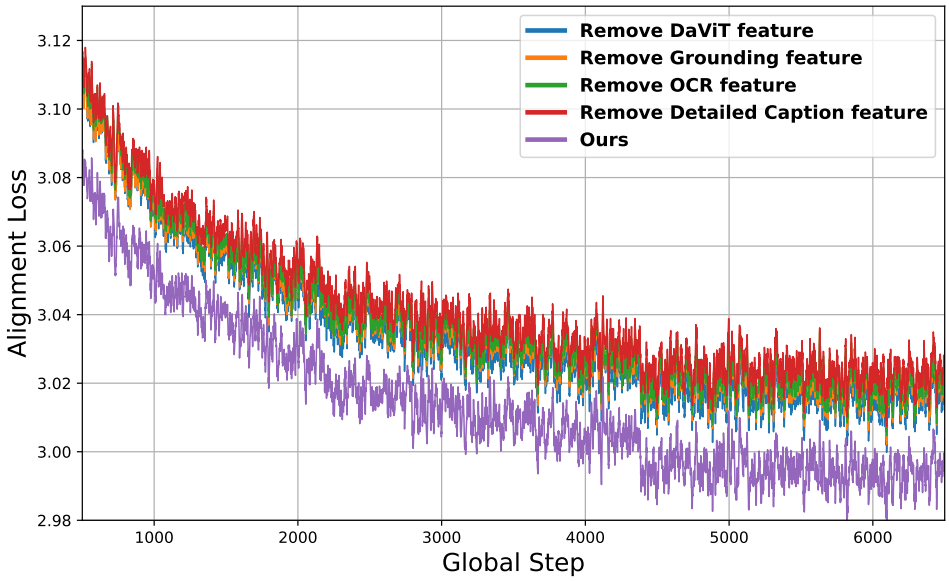



Figure 2.5: We plot the alignment loss for various feature combinations, removing one feature at a time from different depths and breadths. The results clearly show that our method achieves the lowest alignment loss compared to others, highlighting the importance of all features from different depths and breadths for optimal alignment.

| | # Vis tok. | General Benchmarks | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | VQAv2 | GQA | MMBench (EN) | MMBench (CN) | VizWiz | POPE | MM-Vet | MME-P | MME-C | Seed-image | HallusionBench | LLaVA-bench | MMStar |
| Vila 3B | - | 80.4 | 61.5 | 63.4 | 52.7 | 53.5 | 86.9 | 35.4 | 1442.4 | - | 67.9 | - | - | 40.3 |
| Phi 3.5-Vision | - | - | **63.5** | **75.5** | **64.2** | 58.2 | 82.2 | 46.5 | 1473.4 | **412.1** | 69.9 | 53.3 | 68.8 | **49.0** |
| Florence-VL 3B (ours) | 576 | **82.1** | 61.8 | 71.6 | 60.8 | **59.1** | **88.3** | **51.0** | **1498.7** | 403.9 | **70.6** | **58.1** | **71.1** | 44.9 |
| LLaVA next 8B | 2880 | - | **65.4** | 72.2 | - | 57.7 | 86.6 | 41.7 | 1595.1 | 379.3 | 72.7 | 47.7 | **76.8** | - |
| Vila 8B | - | 80.9 | 61.7 | 72.3 | 66.2 | 58.7 | 84.4 | 38.3 | 1577.0 | - | 71.4 | - | - | - |
| Mini-Gemini-HD 8B | 2880 | - | 64.5 | 72.7 | - | - | - | - | **1606.0** | - | 73.2 | - | - | - |
| Cambrain 8B | 576 | - | 64.6 | 75.9 | 67.9 | - | 87.4 | 48.0 | 1547.1 | - | 74.7 | 48.7 | 71.0 | **50.0** |
| Florence-VL 8B (ours) | 576 | **84.7** | 64.4 | **76.2** | **69.5** | **59.1** | **89.9** | **56.3** | 1560.0 | **381.1** | **74.9** | **57.3** | 74.2 | **50.0** |

(a) Results on general multimodal benchmarks.

| | # Vis tok. | Vision centric | | | Knowledge based | | | | OCR & Chart | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Realworldqa | CV-Bench* | MMVP | AI2D | MathVista | MMMU | SciQA-IMG | TextVQA | OCRBench | ChartQA | DocVQA | InfoVQA |
| Vila 3B | - | 53.3 | 55.2 | - | - | 30.6 | 34.1 | 67.9 | 58.1 | - | - | - | - |
| Phi 3.5 Vision | - | 53.5 | 69.3 | **67.7** | **77.4** | - | **43.3** | **89.0** | 61.1 | 59.8 | **72.0** | 75.9 | 40.7 |
| Florence-VL 3B (ours) | 576 | **60.4** | **70.2** | 64.7 | 73.8 | **52.2** | 41.8 | 84.6 | **69.1** | **63.0** | 70.7 | **82.1** | **51.3** |
| LLaVA next 8B | 2880 | 59.6 | 63.8 | 38.7 | 71.6 | 37.4 | 40.1 | 73.3 | 65.4 | 55.2 | 69.3 | 78.2 | - |
| Vila 8B | - | - | - | - | - | - | 36.9 | 79.9 | - | - | - | - | - |
| Mini-Gemini-HD 8B | 2880 | 62.1 | 62.6 | 18.7 | 73.5 | 37.0 | 37.3 | 75.1 | 70.2 | 47.7 | 59.1 | 74.6 | - |
| Cambrian 8B | 576 | **64.2** | 72.2 | 51.3 | 73.0 | 49.0 | 42.7 | 80.4 | 71.7 | 62.4 | 73.3 | 77.8 | - |
| Florence-VL 8B (ours) | 576 | **64.2** | **73.4** | **73.3** | **74.2** | **55.5** | **43.7** | **85.9** | **74.2** | **63.4** | **74.7** | **84.9** | **51.7** |

(b) Results on Vision centric, Knowledge based, and OCR & Chart benchmarks.

Table 2.2: Results on general multimodal benchmarks, Vision centric, Knowledge based, and OCR & Chart benchmarks.

## 2.5 Experiments

Implementation Details: in order to build a state-of-the-art MLLM, we use images from CC12M [14], Redcaps [15], and Commonpool [16] during the pretraining stage, with detailed captions sourced from PixelProse [17]. For the instruction tuning stage, we also curate our high quality instruction tuning datasets, sourcing from Cambrian-7M [7], Vision Flan [18], ShareGPT4V [19], along with additional data from Docmatix [20] to improve chart and diagram comprehension [21]. The detail of

training datasets and experiment details can be found in the appendix.

Evaluation: we evaluate the performance of different MLLM models on 25 benchmarks with four different categories:

- General multimodal benchmarks: VQAv2 [22], GQA [23], MMBench (EN and CN) [24], VisWiz [25], POPE [26], MM-Vet [27], MME Perception [28], MME Cognition [28], Seed-Bench [29], HallusionBench, LLaVA in the Wild [2] and MMStar [30].
- OCR & Chart benchmark: TextVQA [31], OCRBench [32], ChartQA [33], DocVQA [34] and InforVQA [35].
- Knowledge based benchmark: AI2D [36], MathVista [37], MMMU [38] and ScienceQA [39].
- Vision Centric benchmark: MMVP [40], RealworldQA [41] and CV-Bench [7].

Baselines: we select two language backbones: Phi-3.5-mini-Instruct and LLama-3-8B-Instruct. For baseline comparisons among small models, we chose Vila 1.5 3B [42] and Phi 3.5-Vision-Instruct [43]. For the larger models, we select the baselines: LLaVA Next 8B [44], Vila 8B [42], Mini-Gemini-HD 8B [45] and Cambrain 8B [7], using LLama 3 8B Instruct as the language backbone.

Results: in the Table 2.2, we present the results of Florence-VL compared to various baselines across a range of benchmarks, along with the number of visual tokens used. For the smaller-sized model, our model outperforms Vila 3B, and surpasses Phi 3.5 Vision on 12 out of 24 tasks. Notably, Phi 3.5 Vision utilizes 500 billion vision and text tokens [43], with its training data being proprietary and significantly larger than ours. Nonetheless, our Florence-VL 3B remains competitive with this model. For the larger-sized model, our model shows a significant improvement over other baselines on

| | LLM | GQA | MMBench (EN) | MMBench (CN) | VizWiz | POPE | MM-Vet | MME-P | MME-C | HallusionBench | LLaVA-bench | MMStar |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LLaVA 1.5 3B | Phi 3.5 | 61.4 | **69.4** | 60.6 | 38.4 | 86.2 | 35.4 | 1399.5 | 284.6 | 44.5 | **68.0** | 40.6 |
| Florence-VL 3B | Phi 3.5 | **62.7** | 68.7 | **61.7** | **42.6** | **89.9** | 35.4 | **1448.5** | **299.6** | **45.5** | 64.9 | **40.8** |
| LLaVA 1.5 7B | Vicuna 1.5 | 62.0 | 64.8 | **57.6** | 50.0 | 85.9 | 30.6 | 1510.7 | 294.0 | 44.8 | 64.2 | 30.3 |
| Florence-VL 7B | Vicuna 1.5 | **62.7** | **66.1** | 55.8 | **54.5** | **89.4** | **35.2** | **1543.5** | **316.4** | **46.8** | **65.0** | **36.8** |
| LLaVA 1.5 8B | Llama 3 | 62.8 | **71.4** | 65.5 | 49.3 | 84.8 | 34.2 | 1539.4 | 292.5 | 45.7 | **71.0** | 38.5 |
| Florence-VL 8B | Llama 3 | **63.8** | 71.1 | **65.8** | **54.0** | **88.4** | **36.4** | **1584.1** | **346.8** | **46.8** | 66.2 | **39.1** |

(a) Results on general multimodal benchmarks.

| | LLM | Realworldqa | MMVP | AI2D | MathVista | MMMU | SciQA-IMG | TextVQA | OCRBench | ChartQA | DocVQA | InfoVQA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LLaVA 1.5 3B | Phi 3.5 | 54.4 | 2.0 | 63.3 | 30.6 | **40.7** | **72.0** | 43.7 | 30.4 | 16.4 | 28.1 | 26.4 |
| Florence-VL 3B | Phi 3.5 | **58.4** | **6.0** | **64.9** | 30.6 | 39.6 | 68.7 | **61.6** | **40.3** | **21.8** | **46.1** | **29.6** |
| LLaVA 1.5 7B | Vicuna 1.5 | 54.8 | 6.0 | 54.8 | 26.7 | 35.3 | **66.8** | 58.2 | 31.4 | 18.2 | 28.1 | 25.8 |
| Florence-VL 7B | Vicuna 1.5 | **60.4** | **12.3** | **57.2** | **28.0** | **35.6** | 66.5 | **62.8** | **41.4** | **24.3** | **44.5** | **29.4** |
| LLaVA 1.5 8B | Llama 3 | 55.7 | 7.3 | 60.2 | 29.3 | 39.4 | **76.5** | 45.4 | 34.6 | 15.4 | 28.6 | 26.4 |
| Florence-VL 8B | Llama 3 | **59.9** | **8.3** | **62.4** | **31.8** | **39.9** | 73.6 | **68.0** | **41.1** | **23.4** | **44.4** | **29.0** |

(b) Results on Vision centric, Knowledge based, and OCR & Chart benchmarks.

Table 2.3: We compare LLaVA 1.5 with our model (Florence-VL 3B/7B/8B) across multiple multimodal benchmarks. The key difference between them lies in the vision encoders used (CLIP for LLaVA vs. Florence-2 for our model), while we maintain the same training data and backbone LLMs for both. The results show that our models significantly outperform LLaVA 1.5 with the same training data.

most benchmarks. Notably, our model significantly outperforms Cambrain-8B, which utilizes multiple vision encoders and combines their image features, whereas we achieve superior results using only a single vision encoder.

## 2.6 Discussion

Results using LLaVA 1.5 Data: since we curate our training data when building our MLLMs, we disentangle the effects of training data and model architecture to clearly demonstrate our method's effectiveness. Specifically, to highlight the advantages of our model architecture, we use the **exact same** pretraining and instruction dataset as LLaVA 1.5 [2]. We test different language backbones, including Phi-3.5-mini-Instruct, Vicuna 1.5 7B, and LLama-3-8B-Instruct. As shown in Tables 2.3, our model design significantly outperforms the LLaVA architectures when trained on the same dataset. Notably, for OCR & Chart tasks, Florence-VL significantly outperforms LLaVA 1.5, demonstrating that OCR image features are essential for effective text-based image understanding.

Study on Depth Features Impacts: we aim to examine the impact of image features from different depths. For the feature set $[\mathbf{V}, \mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$, we first remove all higher-level features $[\mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$ and retain only the lower-level feature $[\mathbf{V}]$. We then evaluate the performance across different benchmarks, and as shown in Table 2.4, using only the lower-level feature $[\mathbf{V}]$ performs worse than our complete method. Next, we remove the lower-level feature $[\mathbf{V}]$ and keep only the higher-level features $[\mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$. The alignment loss, displayed in Figure 2.5, clearly indicates that excluding the lower-level features (i.e., removing DaViT features) results in a higher alignment loss compared to our method. Therefore, both ablation studies confirm that features from different depths are essential for achieving optimal performance.

Study on Breadth Features Impacts: in Table 2.5, we analyze the impact of each feature from different breadths by individually removing one feature at a time from $[\mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$. For instance, to assess the effect of the caption feature, we retain only the OCR and grounding features. The results in Table 2.5

| Features used | MMBench (EN) | POPE | MM-Vet | MME-P | Seed-image | HallusionBench | LLaVA-bench | AI2D | MathVista | MMMU | OCRBench | ChartQA | DocVQA | InfoVQA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $[\mathbf{V}]$ | 64.3 | 86.1 | 31.1 | 1510.7 | 66.0 | 44.8 | 64.2 | 54.7 | 26.7 | 35.2 | 31.2 | 18.3 | 27.9 | 25.7 |
| $[\mathbf{V}, \mathbf{V}'_{t_1}, \mathbf{V}'_{t_2}, \mathbf{V}'_{t_3}]$ | **66.1** | **89.4** | **35.2** | **1543.5** | **70.3** | **46.8** | **65.0** | **57.2** | **28.0** | **35.6** | **41.4** | **24.3** | **44.5** | **29.4** |

Table 2.4: The comparison between keeping only the lower-level feature $[\mathbf{V}]$ and our method, which includes both lower- and higher-level features, clearly demonstrates that maintaining both types of features achieves better performance.

| | GQA | MMBench (EN) | MMBench (CN) | VizWiz | POPE | MM-Vet | MME-P | MME-C | Seed-image | HallusionBench | LLaVA-bench | MMStar | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Florence-VL 7B | 62.7 | 66.1 | 55.8 | 54.5 | **89.4** | **35.2** | **1543.5** | 316.4 | 70.3 | **46.8** | 65.0 | **36.8** | **58.3** |
| Remove Caption Feature $\mathbf{V}'_{t_1}$ | 62.2 | 64.9 | 56.1 | 53.5 | 89.3 | 31.8 | 1477.8 | **354.3** | 69.0 | 44.9 | **65.2** | 36.0 | 57.6 |
| Remove OCR Feature $\mathbf{V}'_{t_2}$ | 62.0 | 65.6 | 55.4 | 56.0 | 88.8 | 30.2 | 1506.3 | 345.4 | 67.6 | 45.4 | 62.6 | 35.2 | 57.3 |
| Remove Grounding Feature $\mathbf{V}'_{t_3}$ | **63.0** | **66.6** | **56.8** | **56.5** | 88.8 | 32.9 | 1494.8 | 338.9 | **70.8** | 44.7 | 65.1 | 36.2 | 58.2 |

Table 2.5: Ablation study was conducted by removing one high level image feature at a time, demonstrating that all high-level features are essential for maintaining optimal performance.

show that combining all three features yields the best average benchmark performance. Additionally, we plot the alignment loss when each feature is removed individually, as shown in Figure 2.5. This further demonstrates incorporating all three features from different breadths is essential for effectively extracting visual information.

## 2.7 Conclusion

In conclusion, Florence-VL uses Florence-2 as a versatile vision encoder, which provides diverse, task-specific visual representations across multiple computer vision tasks like captioning, OCR, and Grounding. By leveraging Depth-Breadth Fusion (DBFusion), we incorporate a range of visual features from different layers (”Depth”) and prompts (”Breadth”) to create enriched representations

that meet varied perceptual demands of downstream tasks. Our fusion strategy, based on channel concatenation, effectively combines these diverse features, which are then projected as input to the language model.

Through training on a novel data recipe that includes detailed captions for pretraining and diverse instruction tuning data, Florence-VL demonstrates superior alignment between the vision encoder and the LLM, outperforming other models across 25 benchmarks covering vision-centric, knowledge-based, and OCR & Chart tasks. Our analysis underscores the effectiveness of Florence-2's generative capabilities in enhancing MLLM alignment and versatility for a wide range of applications.

For future work, several avenues could further enhance the capabilities and efficiency of Florence-VL. One direction involves improving the DBFusion strategy by exploring more sophisticated fusion techniques that could dynamically adapt the Depth-Breadth balance based on specific downstream task requirements. Additionally, while Florence-2 provides diverse visual representations, future research could explore adaptive vision encoders that select features on-the-fly, optimizing computational efficiency without compromising performance.

# Chapter 3: BLIP3-o: A Family of Fully Open Unified Multimodal Models—Architecture, Training and Dataset

## 3.1 Introduction

Recent advances have demonstrated the potential for unified multimodal representation learning that supports both image understanding and image generation within a single model [46–52]. In this field, despite extensive studies on image understanding, the optimal architecture and training strategy for image generation remain underexplored. The previous debate revolves around two approaches: the first approach quantizes continuous visual features into discrete tokens and models them as a categorical distribution [53, 54]; the second approach generates intermediate visual features or latent representations via the autoregressive model and then conditions on these visual features to generate images through the diffusion model [51, 52]. The recently released GPT-4o image generation [55] was implied to adopt a hybrid architecture with autoregressive and diffusion models following the second approach [55, 56]. The wide range of image representations used in industry inspired our systematic study of design choices in unified models. Specifically, our investigation focuses on three key design axes: (1) **image representations** – whether to encode the images into low-level pixel features (e.g., from VAE-based encoders) or high-level semantic features (e.g., from CLIP image encoders); (2) **training objectives** – Mean Squared Error (MSE) versus Flow Matching [57, 58], and what their impacts on training efficiency and generation quality; (3) **training strategies** – one can use simultaneous multitask training on image understanding and generation like Metamorph [51], or else use sequential training like LMFusion [59] and MetaQuery [52], in which the model is first trained for understanding and then extended for generation.

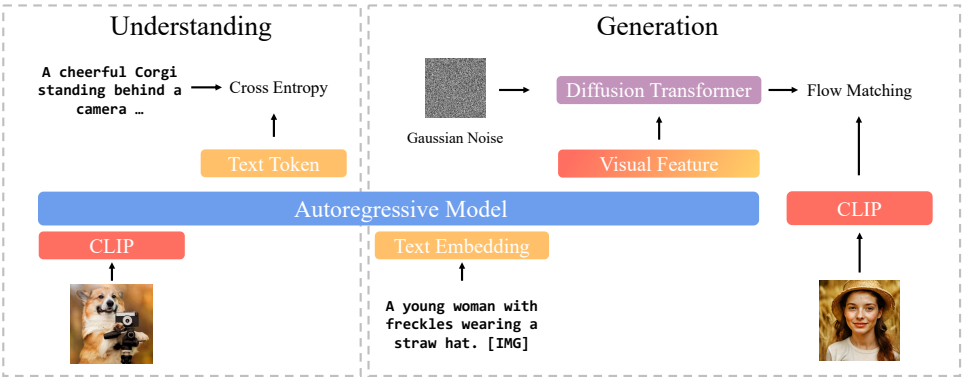



Figure 3.1: The architecture of BLIP3-o. For image understanding part, we use CLIP to encode the image and compute the cross entropy loss between the target text token and predicted text token. For image generation part, autoregressive model first generates a sequence of intermediate visual features, which are then used as conditioning inputs to a diffusion transformer that generates CLIP image features to approximate the ground-truth CLIP features. By using CLIP encoder, image understanding and image generation share the same semantic space, effectively unifying these two tasks.

Our findings reveal that CLIP image features offer more compact and informative representations than VAE features, resulting in both faster training and higher image generation quality. Flow matching loss proves to be more effective than MSE loss, enabling more diverse image sampling and yielding better image quality. Furthermore, we find that a sequential training strategy—first training the autoregressive model on image understanding tasks, then freezing it during training on image generation—achieves the best overall performance. Based on these findings, we develop BLIP3-o, a herd of state-of-the-art unified multimodal models. BLIP3-o leverages diffusion transformer and flow matching on CLIP features (Figure 3.1) and is sequentially trained on image understanding and image generation tasks. To further improve visual aesthetic and instruction following abilities, we carefully curate a 60k high-quality instruction-tuning dataset BLIP3o-60k for image generation, by prompting GPT-4o with a diverse set of prompts spanning scenes, objects, human gestures and more. We observe that supervised instruction tuning on BLIP3o-60k significantly enhances the alignment of BLIP3-o with human preference and improves the aesthetic quality.


Figure 3.2: Visualization results of BLIP3-o 8B at 1024×1024 resolution.

In our experiments, BLIP3-o achieves superior performance across most of the popular benchmarks for image understanding and image generation, with the 8B model scoring 1682.6 on MME-P, 50.6 on MMMU and 0.84 on GenEval. *To support further research and keep the mission of open-source foundation model research like BLIP-3, we fully open-source our models including model weights, code, pretraining and instruction-tuning datasets, and evaluation pipelines. We hope that our work will support the research community and drive continued progress in the unified multimodal domain.*

## 3.2 Unified Multimodal for Image Generation and Understanding

Recent OpenAI's GPT-4o [55] has demonstrated state-of-the-art performance in image understanding, generation and editing tasks. Emerging hypotheses of its architecture [56] suggest a hybrid

pipeline structured as:

$$\textbf{Tokens} \longrightarrow \textbf{[Autoregressive Model]} \longrightarrow \textbf{[Diffusion Model]} \longrightarrow \textbf{Image Pixels}$$

indicating that autoregressive and diffusion models may be jointly leveraged to combine the strengths of both modules. Motivated by this hybrid design, we adopt an autoregressive + diffusion framework in our study. However, the optimal architecture in this framework remains unclear. The autoregressive model produces continuous intermediate visual features intended to approximate ground-truth image representations, raising two key questions. First, what should serve as the ground-truth embeddings: should we use a VAE or CLIP to encode images into continuous features? Second, once the autoregressive model generates visual features, how do we optimally align them with the ground-truth image features, or more generally, how should we model the distribution of these continuous visual features: via a simple MSE loss, or by employing a diffusion-based approach? Thus, we conduct a comprehensive exploration of various design choices in the following section.

## 3.3 Image Generation in Unified Multimodal

In this section, we discuss the design choices involved in building the image generation model within a unified multimodal framework. We begin by exploring how images can be represented as continuous embeddings through encoder–decoder architectures, which play a foundational role in learning efficiency and generation quality.

### 3.3.1 Image Encoding and Reconstruction

Image generation typically begins by encoding an image into a continuous latent embedding using an encoder, followed by a decoder that reconstructs the image from this latent embedding. This encoding-decoding pipeline can effectively reduce the dimensionality of the input space in image generation, facilitating efficient training. In the following, we discuss two widely used encoder–decoder paradigms.

Variational Autoencoders: Variational Autoencoders (VAEs) [60, 61] are a class of generative models that learn to encode images into a structured, continuous latent space. The encoder approximates the posterior distribution over the latent variables given the input image, while the decoder reconstructs the image from samples drawn from this latent distribution. Latent diffusion models build on this framework by learning to model the distribution of compressed latent representations, rather than raw image pixels. By operating in the VAE latent space, these models significantly reduce the dimensionality of the output space, thereby lowering computational costs and enabling more efficient training. After the denoising steps, the VAE decoder maps the generated latent embeddings into raw image pixels.

CLIP Encoder with Diffusion Decoder: CLIP [62] models have become foundational encoders for image understanding tasks [63], owing to its strong ability to extract rich, high-level semantic features from images through contrastive training on large-scale image–text pairs. However, leveraging these features for image generation remains a non-trivial challenge, as CLIP was not originally designed for reconstruction tasks. Emu2 [47] presents a practical solution by pairing a CLIP-based encoder with a diffusion-based decoder. Specifically, it uses EVA-CLIP to encode images into continuous visual embeddings and reconstructs them via a diffusion model initialized from SDXL-base [64]. During

training, the diffusion decoder is fine-tuned to use the visual embeddings from EVA-CLIP as conditions to recover the original image from Gaussian noise, while the EVA-CLIP remains frozen. This process effectively combines the CLIP and diffusion models into an image autoencoder: the CLIP encoder compresses an image into semantically rich latent embeddings, and the diffusion-based decoder reconstructs the image from these embeddings. Notably, although the decoder is based on diffusion architecture, it is trained with a reconstruction loss rather than probabilistic sampling objectives. Consequently, during inference, the model performs deterministic reconstruction.

Discussion: these two encoder–decoder architectures, i.e., VAEs and CLIP-Diffusion, represent distinct paradigms for image encoding and reconstruction, each offering specific advantages and trade-offs. VAEs encode the image into low-level pixel features and offer better reconstruction quality. Furthermore, VAEs are widely available as off-the-shelf models and can be integrated directly into image generation training pipelines. In contrast, CLIP-Diffusion requires additional training to adapt the diffusion models to various CLIP encoders. However, CLIP-Diffusion architectures offer significant benefits in terms of image compression ratio. For example, in both Emu2 [47] and our experiments, each image regardless of its resolution can be encoded into a fixed length of 64 continuous vectors, providing both compact and semantically rich latent embeddings. By contrast, VAE-based encoders tend to produce a longer sequence of latent embeddings for higher-resolution inputs, which increases the computational burden in the training procedure.

### 3.3.2 Modeling Latent Image Representation

After obtaining continuous image embeddings, we proceed to model them using autoregressive architectures. Given a user prompt (e.g., *"A young woman with freckles wearing a straw hat."*), we

first encode the prompt into a sequence of embedding vectors $\mathbf{C}$ using the autoregressive model's input embedding layer, and append a learnable query vector $\mathbf{Q}$ to $\mathbf{C}$, where $\mathbf{Q}$ is randomly initialized and optimized during training. As the combined sequence $[\mathbf{C};\mathbf{Q}]$ is processed through the autoregressive transformer, $\mathbf{Q}$ learns to attend to and extract relevant semantic information from the prompt $\mathbf{C}$. The resulting $\mathbf{Q}$ is interpreted as the intermediate visual features or latent representation generated by the autoregressive model, and is trained to approximate the ground-truth image feature $\mathbf{X}$ (obtained from VAE or CLIP).

In the following, we introduce two training objectives: Mean Squared Error (MSE) and Flow Matching, for learning to align $\mathbf{Q}$ with the ground-truth image embedding $\mathbf{X}$.

MSE Loss: the Mean Squared Error (MSE) loss is a straightforward and widely used objective for learning continuous image embeddings [46, 47]. Given the predicted visual features $\mathbf{Q}$ produced by the autoregressive model and the ground-truth image features $\mathbf{X}$, we first apply a learnable linear projection to align the dimensionality of $\mathbf{Q}$ with that of $\mathbf{X}$. The MSE loss is then formulated as:

$$\mathscr{L}_{\text{MSE}} = \|\mathbf{X} - \mathbf{W}\mathbf{Q}\|_2^2,$$

where $\mathbf{W}$ denotes the learnable projection matrix.

Flow Matching: note that using MSE loss only aligns the predicted image features $\mathbf{Q}$ with the mean value of the target distribution. An ideal training objective would model the probability distribution of continuous image representation.

We propose to use flow matching [65], a diffusion framework that can sample from a target continuous distribution by iterative transporting samples from a prior distribution (e.g., Gaussian).

Given a ground-truth image feature $X_1$ and the condition $\mathbf{Q}$ encoded by an autoregressive model, at each training step, we sample a timestep $t \sim \mathscr{U}(0,1)$, and noise $X_0 \sim \mathscr{N}(0,1)$. Then diffusion transformer learns to predict the velocity $V_t = \frac{dX_t}{dt}$ at the timestep $t$ conditioned on $\mathbf{Q}$, in the direction of $X_1$. Following previous work [58], we compute $X_t$ by a simple linear interpolation between $X_0$ and $X_1$:

$$X_t = tX_t + (1-t)X_0,$$

and the analytical solution of $V_t$ can be expressed as:

$$\mathbf{V}_t = \frac{d\mathbf{X}_t}{dt} = \mathbf{X}_t - \mathbf{X}_0.$$

Finally, the training objective is defined as:

$$\mathscr{L}_{\text{Flow}}(\theta) = \mathbb{E}_{(X_1,\mathbf{Q})\sim\mathscr{D},t\sim\mathscr{U}(0,1),X_0\sim\mathscr{N}(0,1)}\left[\|V_\theta(X_t,\mathbf{Q},t) - V_t\|^2\right],$$

where $\theta$ is the diffusion transformer's parameters, and $V_\theta(X_t,\mathbf{Q},t)$ denotes the predicted velocity based on an instance $(X_1,\mathbf{Q})$, timestep $t$, and noise $X_0$.

Discussion: unlike discrete tokens, which inherently support sampling-based strategies for exploring diverse generation paths, continuous representations lack this property. Specifically, under an MSE-based training objective, the predicted visual features $\mathbf{Q}$ become nearly deterministic for a given prompt. As a result, the output images, regardless of whether the visual decoder is based on VAEs or CLIP + Diffusion architectures, remain almost identical across multiple inference runs. This determinism highlights a key limitation of the MSE objective: it constrains the model to produce a single,

fixed output for each prompt, thereby limiting generation diversity. In contrast, the flow matching framework enables the model to inherit the stochasticity of the diffusion process. This allows the model to generate diverse image samples conditioned on the same prompt, facilitating broader exploration of the output space. However, this flexibility comes at the cost of increased model complexity. Flow matching introduces additional learnable parameters compared to MSE. In our implementation, we use a diffusion transformer (DiT), and empirically find that scaling its capacity yields significant performance improvements.

### 3.3.3 Design Choices

The combination of different image encoder–decoder architectures and training objectives gives rise to a range of design choices for image generation models. These design choices, illustrated in Figure 3.3, significantly influence both the quality and controllability of the generated images. In this section, we summarize and analyze the trade-offs introduced by different encoder types (e.g., VAEs vs. CLIP encoders) and loss functions (e.g., MSE vs. Flow Matching).

CLIP + MSE: following Emu2 [47], Seed-X [46] and Metamorph [51], we use CLIP to encode images into 64 fixed-length semantic-rich visual embeddings. The autoregressive model is trained to minimize the Mean Squared Error (MSE) loss between the predicted visual features $\mathbf{Q}$ and the ground-truth CLIP embedding $\mathbf{X}$, as illustrated in Figure 3.3(a). During inference, given a text prompt $\mathbf{C}$, the autoregressive model predicts the latent visual features $\mathbf{Q}$, which is subsequently passed to a diffusion-based visual decoder to reconstruct the real image.

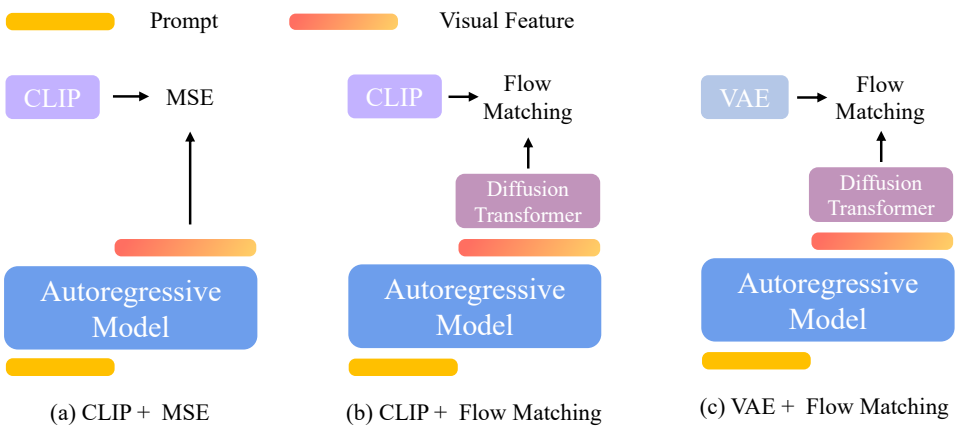



Figure 3.3: Three design choices for image generation in unified multimodal model. All designs use a **Autoregressive + Diffusion** framework but vary in their image generation components. For the flow matching loss, we keep the autoregressive model frozen and only fine-tune the image generation module to preserve the model's language capabilities.

CLIP + Flow Matching: as an alternative to MSE loss, we employ flow matching loss to train the model to predict ground-truth CLIP embeddings, as illustrated in Figure 3.3(b). Given a prompt **C**, the autoregressive model generates a sequence of visual features **Q**. These features are used as conditions to guide the diffusion process, yielding a predicted CLIP embedding to approximate the ground-truth CLIP features. In essence, the inference pipeline involves two diffusion stages: the first uses the conditioning visual features **Q** to iteratively denoise into CLIP embeddings. And the second converts these CLIP embeddings into real images by diffusion-based visual decoder. This approach enables stochastic sampling at the first stage, allowing for greater diversity in image generation.

VAE + Flow Matching: we can also use flow matching loss to predict the ground truth VAE features seen in Figure 3.3(c), which is similar to MetaQuery [52]. At inference time, given a prompt **C**, the autoregressive model produces visual features **Q**. Then, conditioning on **Q** and iteratively removing noise at each step, the real images are generated by the VAE decoder.

VAE + MSE: because our focus is on autoregressive + diffusion framework, we exclude VAE + MSE approaches, as they do not incorporate any diffusion module.

Implementation Details: to compare various design choices, we use Llama-3.2-1B-Instruct as autoregressive model. Our training data consists of CC12M [14], SA-1B [6], and JourneyDB [66], amounting to approximately 25 million samples. For CC12M and SA-1B, we utilize the detailed captions generated by LLaVA, while for JourneyDB we use the original captions. The detailed description of image generation architecture using flow matching loss is provided in Section 3.5.1.

Results: we report the FID score [67] on MJHQ-30k [68] for visual aesthetic quality, along with GenEval [69] and DPG-Bench [70] metrics for evaluating prompt alignment. We plot the results for each design choice at approximately every 3,200 training steps. Figure 3.4 shows that CLIP + Flow Matching achieves the best prompt alignment scores on both GenEval and DPG-Bench, while VAE + Flow Matching produces the lowest (best) FID, indicating superior aesthetic quality. However, FID has inherent limitations: it quantifies stylistic deviation from the target image distribution and often overlooks true generative quality and prompt alignment. In fact, our FID evaluation of GPT-4o on the MJHQ-30k dataset produced a score of around 30.0, underscoring that FID can be misleading in the image generation evaluation. In general, our experiments demonstrate CLIP + Flow Matching as the most effective design choice.

Discussion: in this section, we present a comprehensive evaluation of various design choices for image generation within a unified multimodal framework. Our results clearly show that CLIP's features produce more compact and semantically rich representations than VAE features, yielding higher train-

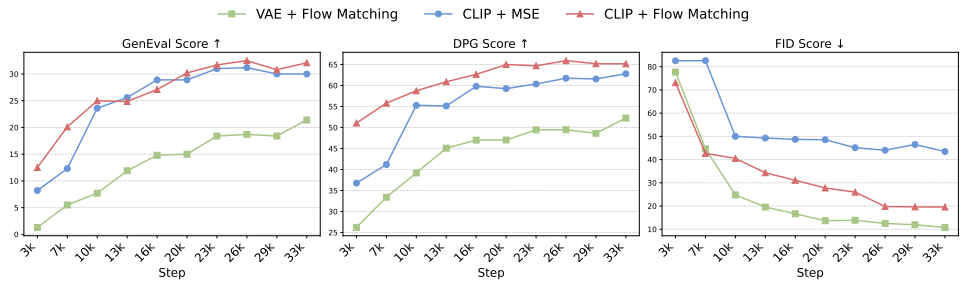
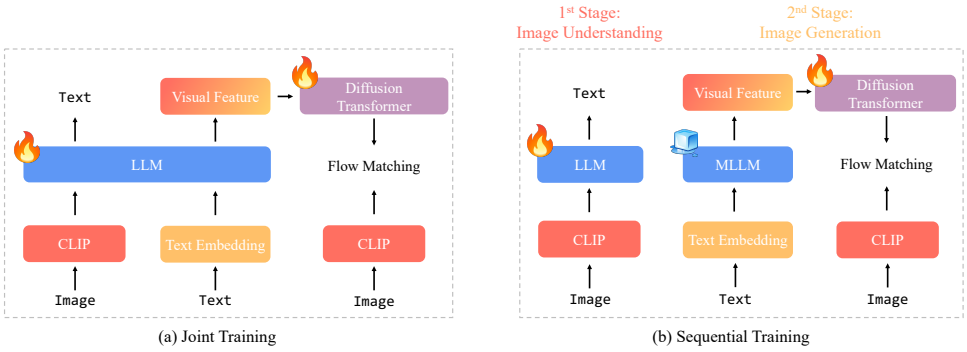



Figure 3.4: Comparison of different design choices.




(a) Joint Training (b) Sequential Training

Figure 3.5: Joint Training vs. Sequential Training: Joint training performs multitask learning by mixing image-understanding and image-generation data, updating both the autoregressive backbone and the generation module simultaneously. Sequential training separates the process: first, the model is trained only on image-understanding tasks; then the autoregressive backbone is frozen and only the image-generation module is trained in a second stage.

ing efficiency. Autoregressive models more effectively learn these semantic-level features compared to pixel-level features. Furthermore, flow matching proves to be a more effective training objective for modeling the image distribution, resulting in greater sample diversity and enhanced visual quality.

## 3.4 Training Strategies for Unified Multimodal

Building on our image generation study, the next step is to develop a unified model that can perform both image understanding and image generation. We use CLIP + Flow Matching for the image generation module. Since image understanding also operates in CLIP's embedding space, we align both tasks within the same semantic space, enabling their unification. In this context, we discuss two training strategies to achieve this integration.

### 3.4.1 Joint Training Versus Sequential Training

Joint Training: joint training of image understanding and image generation has become a common practice in recent works such as Metamorph [51], Janus-Pro [50], and Show-o [48]. Although these methods adopt different architectures for image generation, all perform multitask learning by mixing data for image generation and understanding.

Sequential Training: instead of training image understanding and generation together, we follow a two-stage approach. In the first stage, we train only the image understanding module. In the second stage, we freeze the MLLM backbone, and train only the image generation module.

### 3.4.2 Discussion

In joint training setting, although image understanding and generation tasks possibly benefit each other as demonstrated by Metamorph [51], two critical factors influence their synergistic effect: (i) the total data size and (ii) the data ratio between image understanding and generation data. In contrast, sequential training offers greater flexibility: It lets us freeze the autoregressive backbone

and maintain the image understanding capability. We can dedicate all training capacity to image generation, avoiding any inter-task effects in joint training. Also motivated by MetaQuery [52], we will choose sequential training to construct our unified multimodal model and defer joint training to future work.

## 3.5 BLIP3-o: Our State-of-the-Art Unified Multimodal

Based on our findings, we adopt CLIP + Flow Matching and sequential training to develop our own state-of-the-art unified multimodal model BLIP3-o.

### 3.5.1 Model Architecture

We develop two different size models: 8B parameter model trained on proprietary data and 4B parameter model using **only open source data**. Given the existence of strong open source image understanding models, such as Qwen 2.5 VL [71], we skip image understanding training stage and build our image generation module directly on Qwen 2.5 VL. In the 8B model, we freeze the Qwen2.5-VL-7B-Instruct backbone and train the diffusion transformers, totaling 1.4 B trainable parameters. The 4B model follows the same image generation architecture but uses Qwen2.5-VL-3B-Instruct as backbone. For the diffusion transformer architecture, we leverage the architecture of the Lumina-Next model [72] for our DiT. More details about the model architecture can be seen in the Appendix.

### 3.5.2 Training Recipe

Stage 1: Pretraining for Image Generation. For 8B model, we combine around 25 million open-source data with an additional 30 million proprietary images. Each image is paired with a detailed

caption. To improve generalization to varying prompt lengths, we also include around 10% (6 million) shorter captions. For the fully open-source 4B model, we use 25 million publicly available images with detail captions and around 10% (3 million) short captions. More details about the pretraining data can be seen in Appendix. **To support the research community, we release 25 million detailed captions and 3 million short captions.**

Stage 2: Instruction Tuning for Image Generation. After the image generation pre-training stage, we observe several failures in the model generation: complex human gestures, knowledge-based objects and simple text. Although these categories were intended to be covered during pretraining, the limited size of our pretraining corpus meant they weren't adequately addressed. To remedy this, we perform instruction tuning focused specifically on these domains. For each category, we prompt GPT-4o to generate roughly 10k prompt–image pairs, creating a targeted dataset that improves the model's ability to handle these cases. To improve visual aesthetics quality, we also expand our data with prompts drawn from JourneyDB [66] and DALL·E 3. This process yields a curated collection of approximately 60k high quality prompt–image pairs. We also release this 60k instruction tuning dataset.

### 3.5.3 Results

For baseline comparison, we include the following unified models: EMU2 Chat [47], Chameleon [53], Seed-X [46], VILA-U [73], LMfusion [59], Show-o [48], EMU3 [54], MetaMorph [51], Token-Flow [74], Janus [49], and Janus-Pro [50].

Image Understanding: in the image understanding task, we evaluate the benchmark performance on VQAv2 [22], MMBench [24], SeedBench [29], MM-Vet [27], MME-Perception and MME-Cognition [28],

MMMU [38], TextVQA [31], and RealWorldQA [41]. As shown in Table 3.1, our BLIP3-o 8B achieves the best performance in most benchmarks.

| Model | VQAv2 | MMBench | SEED | MM-Vet | MME-P | MME-C | MMMU | RWQA | TEXTVQA |
|---|---|---|---|---|---|---|---|---|---|
| EMU2 Chat 34B | - | - | 62.8 | 48.5 | - | - | 34.1 | - | 66.6 |
| Chameleon 7B | - | 19.8 | 27.2 | 8.3 | 202.7 | - | 22.4 | 39.0 | 0.0 |
| Chameleon 34B | - | 32.7 | - | 9.7 | 604.5 | - | 38.8 | 39.2 | 0.0 |
| Seed-X 17B | 63.4 | 70.1 | 66.5 | 43.0 | 1457.0 | - | 35.6 | - | - |
| VILA-U 7B | 79.4 | 66.6 | 57.1 | 33.5 | 1401.8 | - | 32.2 | 46.6 | 48.3 |
| LMFusion 16B | - | - | 72.1 | - | 1603.7 | 367.8 | 41.7 | 60.0 | - |
| Show-o 1.3B | 69.4 | - | - | - | 1097.2 | - | 27.4 | - | - |
| EMU3 8B | 75.1 | 58.5 | 68.2 | 37.2 | 1243.8 | 266.1 | 31.6 | 57.4 | 64.7 |
| MetaMorph 8B | - | 75.2 | 71.8 | - | - | - | 41.8 | 58.3 | 60.5 |
| TokenFlow-XL 14B | 77.6 | 76.8 | 72.6 | 48.2 | 1551.1 | 371.1 | 43.2 | 56.6 | 77.6 |
| Janus 1.3B | 77.3 | 75.5 | 68.3 | 34.3 | 1338.0 | - | 30.5 | - | - |
| Janus Pro 7B | - | 79.2 | 72.1 | 50.0 | 1567.1 | - | 41.0 | - | - |
| BLIP3-o 4B | 75.9 | 78.6 | 73.8 | 60.1 | 1527.7 | 632.9 | 46.6 | 60.4 | 78.0 |
| BLIP3-o 8B | **83.1** | **83.5** | **77.5** | **66.6** | **1682.6** | **647.1** | **50.6** | **69.0** | **83.1** |

Table 3.1: Results on image understanding benchmarks. We highlight the best results in **bold**.

Image Generation: in the image generation benchmark, we report GenEval [69] and DPG-Bench [70] to measure prompt alignment, WISE [75] to evaluate world knowledge reasoning capability. As shown in Table 3.2, BLIP3-o 8B achieves a GenEval score of 0.84, a WISE score of 0.62, but scores lower on DPG-Bench. Because model-based evaluation for DPG-Bench can be unreliable, we complement these results with a human study on all DPG-Bench prompts in the next section. Furthermore, we also find our instruction tuning dataset BLIP3o-60k yields immediate gains: using only 60k prompt–image pairs, both prompt alignment and visual aesthetics improve markedly, and many generation artifacts are quickly reduced. Although this instruction tuning dataset cannot fully resolve some difficult cases, such as complex human gestures generation, it nonetheless delivers a substantial boost in overall image quality.

| Model | GenEval | DPG-Bench | WISE |
|---|---|---|---|
| Chameleon 7B | 0.39 | – | - |
| Seed-X 17B | 0.51 | – | - |
| LLaVAFusion 16B | 0.63 | – | - |
| Show-o 1.3B | 0.68 | 67.27 | 0.35 |
| EMU3 8B | 0.66 | 80.60 | 0.39 |
| TokenFlow-XL 14B | 0.63 | 73.38 | - |
| Janus 1.3B | 0.61 | 79.68 | 0.18 |
| Janus Pro 7B | 0.80 | **84.19** | 0.35 |
| BLIP3-o 4B | 0.81 | 79.36 | 0.50 |
| BLIP3-o 8B | **0.84** | 81.60 | **0.62** |

Table 3.2: Image generation benchmark results.

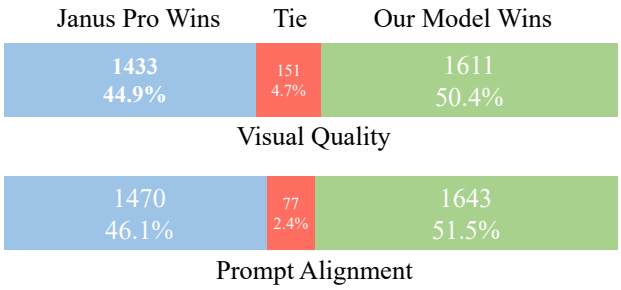



Figure 3.6: Human study results for DPG-Bench between Janus Pro and our model.

### 3.5.4 Human Study

In this section, we conduct a human evaluation comparing BLIP3-o 8B and Janus Pro 7B on about 1,000 prompts drawn from the DPG-Bench. We assess two metrics: visual quality and prompt alignment. The details about each metric can be seen in the Appendix. Each metric was assessed in two separate rounds, resulting in roughly 3,000 judgments per criterion. As illustrated in Figure 3.6, BLIP3-o outperforms Janus Pro on both visual quality and prompt alignment, even though Janus Pro achieves a higher DPG score in Table 3.2. The $p$-values for Visual Quality and Prompt Alignment are 5.05e-06 and 1.16e-05, respectively, indicating that our model significantly outperforms Janus Pro with high statistical confidence.

## 3.6 Conclusion

We present the first systematic exploration of hybrid autoregressive and diffusion architectures for unified multimodal modeling, evaluating three critical aspects: image representation, training objective and training strategy. Building on these insights, we introduce BLIP3-o, a family of state-of-

the-art unified models enhanced with a 60k instruction tuning dataset BLIP3o-60k that substantially improves prompt alignment and visual aesthetics. We are actively working on applications for the unified model, including iterative image editing, visual dialogue, and step-by-step visual reasoning.

# Chapter 4: BLIP3o-NEXT: Next Frontier of Native Image Generation

## 4.1 Introduction

Recently image generation has become a cornerstone of modern artificial intelligence, empowering models to not only synthesize photorealistic images from textual descriptions but also edit existing images with remarkable precision and semantic consistency [52, 55, 76–79]. These capabilities have redefined how machines interpret, represent, and communicate visual information, enabling a wide range of creative and practical applications, from content creation and design to scientific visualization and simulation.

In this paper, we introduce BLIP3o-NEXT, a novel image generation foundation model that advances the frontier of multimodal learning through innovations in model architecture, multi-task pretraining, reinforcement learning, and comprehensive data engineering. BLIP3o-NEXT adopts a hybrid Autoregressive + Diffusion architecture [52, 76]. The autoregressive model [71] takes a user prompt together with a set of reference images (for image editing) and generates a sequence of discrete image tokens conditioned on the multimodal input. The diffusion model [80] subsequently leverages the final hidden states of generated discrete tokens as conditioning signals to synthesize the final image. This architecture combines the semantic compositionality and global structure understanding captured by the autoregressive model with the fine-grained detail rendering capability of diffusion models, enabling BLIP3o-NEXT to produce images that are both semantically coherent and photorealistically detailed.

The autoregressive model is trained on three primary tasks: (1) text-to-image generation, (2) input image reconstruction, and (3) image editing, which together equip the model with the fundamental capabilities required for diverse downstream tasks. In the post-training stage, beyond fine-tuning on

carefully curated high-quality datasets [19, 76, 77], we perform extensive reinforcement learning (RL) to further enhance the model's text rendering quality and instruction following ability. We propose an efficient RL framework specifically tailored for the BLIP3o-NEXT architecture by leveraging the discrete image tokens. This design allows seamless integration with existing RL infrastructures originally developed for language models, while achieving performance comparable to recent flow-based RL approaches.

Taking advantage of the Autoregressive + Diffusion design, BLIP3o-NEXT exhibits great flexibility in integrating the VAE features [81] of reference images for editing tasks. Through empirical experiments, we find that concatenating VAE features to (1) hidden states generated by the autoregressive model as conditions, and (2) the random noise as initial input in the diffusion process, achieves the highest consistency and visual fidelity. Additionally, powered by the strong multimodal reasoning and understanding capability of its autoregressive backbone, BLIP3o-NEXT can faithfully follow user instructions for image editing.

In summary, BLIP3o-NEXT makes the following key contributions:

- **A novel and scalable Autoregressive + Diffusion architecture** that advances the next frontier of native image generation.
- **An efficient reinforcement learning method for image generation** that can be seamlessly integrated with existing RL infrastructures for language models, improving text rendering and instruction following abilities.
- **Systematic studies on improving consistency in image editing**, including strategies for integrating VAE features from reference images.

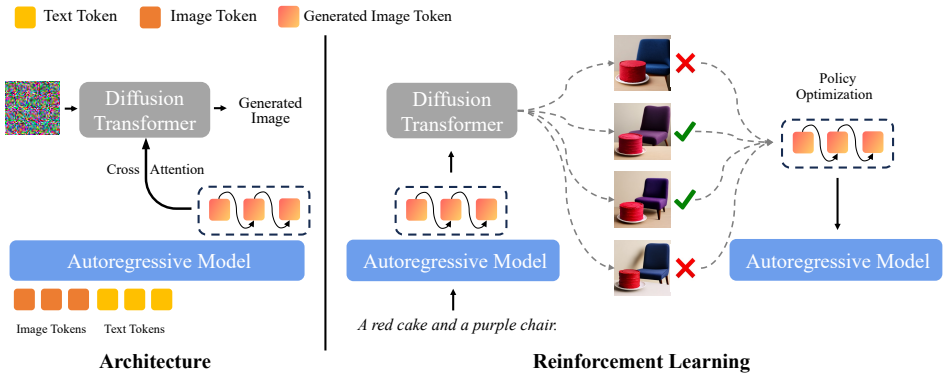



Figure 4.1: The architecture of BLIP3o-NEXT (left) and its reinforcement learning pipeline (right). BLIP3o-NEXT adopts an Autoregressive (AR) + Diffusion design, where the AR module autoregressively generates image conditions for the diffusion model. The model is jointly optimized with both AR and diffusion objectives. During reinforcement learning, rollouts are rendered from the diffusion transformer, and policy optimization is performed directly on the AR model, enabling seamless integration with existing RL infrastructures originally developed for language models.

- **Strong performance across diverse benchmarks**, comprehensive evaluation on text-to-image generation benchmarks and image-editing benchmarks reveals that BLIP3o-NEXT consistently outperform existing models.

*To support future research and uphold the open-source philosophy of the BLIP3 family, we fully release BLIP3o-NEXT, including pretrained and post-trained model weights, datasets, detailed training and inference code, and evaluation pipelines, ensuring complete reproducibility. We hope that our work will foster continued progress in native image generation.*

## 4.2 Overview of Architectures for Native Image Generation

### 4.2.1 Autoregressive + Diffusion

In recent developments in native image generation, approaches that integrate autoregressive models with diffusion models continue to achieve state-of-the-art performance. In these frameworks,

the autoregressive backbone encodes the input prompt, whether text or other modalities, and produces conditions for the diffusion model to generate the final image. Early explorations include Emu2 [47] and Seed-X [46]. Following the success of GPT-4o [55], subsequent works such as MetaQuery [52], UniGen [82], and BLIP3-o [76] push the direction further, while recent models like Qwen-Image [79], Gemini Nano Banana and Seedream [83] continue to push the performance frontier. The advantage of Autoregressive + Diffusion is its **simplicity** and **scalability**, and also leveraging the strengths of existing autoregressive models, such as transferring instruction following and reasoning capabilities into image generation procedure.

Within this framework, the question is how to derive the conditioning signal from the autoregressive model for the diffusion model. Existing approaches can be broadly grouped into the following categories: (1) The autoregressive model directly generates continuous embeddings (e.g., CLIP representations), which are then used as conditions for the diffusion model, similar to Emu2 [47] and MetaMorph [51]. (2) The autoregressive model compresses all input information, including text and images, into a fixed number of learnable query, as demonstrated by Seed-X [46], MetaQuery [52], and BLIP3-o [76]. (3) The autoregressive model encodes both image and text inputs, and the hidden states from the autoregressive model are directly used as the conditioning signal for the diffusion model, as in OmniGen2 [77], Qwen-Image [79], and MANZANO [84].

Since continuous embeddings and learnable query tokens generate conditioning signals by sampling from AR models, they tend to perform better on image generation tasks requiring reasoning capability compared with directly using hidden states [85]. However, both continuous embeddings and learnable query compress all conditioning information into a fixed number of tokens, which inevitably limits their representational capacity. For example, Emu2 [47] and BLIP3-o [76] usually use 64 tokens. In contrast, using hidden states offers a more flexible and efficient way for encoding the

condition.

### 4.2.2 Mixture of Transformers

LMFusion [59] and BAGEL [78] adopt a different architecture for native image generation. At a high level, they employ two transformer experts to separately process understanding and generation information, while tokens from different modalities interact through shared multimodal self-attention within each transformer block. Although this structure facilitates richer information sharing across modalities, it faces challenges in flexibility and scalability, and suffers from high inference latency.

### 4.2.3 BLIP3o-NEXT Architecture

In BLIP3o-NEXT, we still adopt an Autoregressive + Diffusion architecture, where an autoregressive model first predicts discrete image tokens from multimodal inputs, and the hidden states of these tokens are subsequently used to condition a diffusion model for high-fidelity image synthesis. Unlike previous approaches, where the autoregressive model takes in and generates continuous image embeddings [46, 51, 52, 76], our autoregressive model operates on discrete visual tokens. Specifically, each image is first encoded using the SigLIP2 model [86], and the resulting continuous embeddings are quantized into a finite vocabulary of tokens, yielding 729 discrete tokens per image given the image resized to 384x384. Conditioned on a text prompt (or reference images for image editing), the autoregressive model is trained through next-token prediction over discrete image tokens. The diffusion model is then applied on top of the autoregressive outputs to refine fine-grained details. Specifically, the hidden states of the predicted tokens serve as conditioning signals for the diffusion model to diffuse the final VAE features, which produce the final high-fidelity images, as illustrated in Figure 4.1 (left).

During training, BLIP3o-NEXT is optimized with two objectives:

$$\mathscr{L} = \mathscr{L}_{\text{CE}} + \lambda \, \mathscr{L}_{\text{diff}}, \tag{4.1}$$

where $\mathscr{L}_{\text{CE}}$ denotes the cross-entropy loss over text and discrete image tokens in autoregressive model, $\mathscr{L}_{\text{diff}}$ is the diffusion loss, and $\lambda$ is a balancing weight. Due to computational constraints, we initialize the autoregressive model with Qwen3 [87] and the diffusion transformer with SANA1.5 [80], resulting in a total of approximately 3B parameters. For training, we employ BLIP3o-Pretrain [76] as the pretraining corpus and BLIP3o-60K [76] as the instruction tuning dataset.

### 4.2.4 Discussion

From GPT-4o, where an autoregressive model produces discrete image tokens and is refined by a diffusion model, to more recent models such as Qwen-Image [79], architectures are increasingly dominated by diffusion backbones. For example, Qwen-Image integrates a 7B vision language model with a 20B diffusion transformer. In practice, GPT-4o exhibits higher inference latency, whereas advances in accelerating diffusion transformers [88–90] have enabled diffusion-centric models to achieve both greater scalability and faster inference efficiency.

In practice, we observe that most architectures deliver comparable performance, with minor design variations yielding marginal differences. Therefore, an architecture can be considered effective, as long as it remains simple, scalable, and supports fast inference.

## 4.3 Image Generation with Reinforcement Learning

Reinforcement learning (RL) has been extensively explored in large language models, but remains relatively underexplored in the image generation domain. GPT-4o [55] is the representative native image generation model to successfully integrate RL. In this work, we primarily discuss two directions for applying RL to image generation: (1) applying RL to the autoregressive mode under the BLIP3o-NEXT framework, and (2) applying RL to the diffusion model, for example Flow-GRPO[91].

### 4.3.1 RL for Autoregressive Model

Applying RL to the autoregressive model is a natural extension of language modeling. The key advantage of this approach is its compatibility with existing RL frameworks developed for language models, allowing most training and inference infrastructures to be easily adapted for image generation training.

We use the Group Relative Policy Optimization (GRPO) algorithm [92] under BLIP3o-NEXT framework. For each text prompt $p \sim \mathscr{D}$, the previous policy $\pi_{\theta_{old}}$, i.e., the autoregressive model, samples a group of $G$ trajectories $\{o_1, \ldots, o_G\}$. Each of the trajectory consists of 729 discrete image tokens for an image. These trajectories are then decoded by a frozen diffusion model to generate the corresponding images $\{I_1, I_2, \ldots, I_G\}$, each assigned a reward score $\{r_1, \ldots, r_G\}$ by the reward function. The rewards are normalized across the group to compute the advantages $A_i$, following the GRPO procedure. During training, the diffusion model remains frozen, while the policy model $\pi_\theta$ is

| Model | Single Obj. | Two Obj. | Counting | Colors | Position | Color Attri. | Overall |
|---|---|---|---|---|---|---|---|
| FLUX.1-dev (12B) [93] | 0.98 | 0.93 | 0.75 | 0.93 | 0.68 | 0.65 | 0.82 |
| OmniGen2 (7B) [77] | 1.00 | 0.95 | 0.64 | 0.88 | 0.55 | 0.76 | 0.80 |
| Qwen-Image (27B) [79] | 0.99 | 0.92 | 0.89 | 0.88 | 0.76 | 0.77 | 0.87 |
| Metaqueries XL (7B) [52] | – | – | – | – | – | – | 0.80$^{\dagger}$ |
| BAGEL (14B) [78] | 0.98 | 0.95 | 0.84 | 0.95 | 0.78 | 0.77 | 0.88$^{\dagger}$ |
| BLIP3o (8B) [76] | – | – | – | – | – | – | 0.84 |
| BLIP3o-NEXT-GRPO-GenEval (3B) | 0.99 | 0.95 | 0.88 | 0.90 | 0.92 | 0.79 | 0.91 |

Table 4.1: Quantitative results on GenEval benchmarks. ($^{\dagger}$ denotes the rewritten prompts.)

optimized by maximizing the following objective:

$$\mathscr{L}_{GRPO}(\theta) = \mathbb{E}_{p\sim\mathscr{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(\cdot|p)} \tag{4.2}$$
$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|p)}{\pi_{\theta_{old}}(o_i|p)}A_i,\ \text{clip}\left(\frac{\pi_\theta(o_i|p)}{\pi_{\theta_{old}}(o_i|p)},1-\varepsilon,1+\varepsilon\right)A_i\right) \quad -\beta\,\mathbb{D}_{KL}\left(\pi_\theta\,\|\,\pi_{\theta_{ref}}\right)\right).$$

Here, $\varepsilon$ and $\beta$ are hyperparameters, $\pi_{\theta_{ref}}$ is a fixed reference policy, and $\mathbb{D}_{KL}\left(\pi_\theta\,\|\,\pi_{\theta_{ref}}\right)$ is a KL divergence penalty that constrains policy updates relative to the reference policy. RL training can be seen in Figure 4.1 right.

### 4.3.2 RL for Diffusion Model

Instead of autoregressive model, RL can also be applied to the diffusion model. Qwen-Image [79] is the representative model to successfully apply both DPO [94] and Flow-GRPO [91] within a native image generation model.

After the DPO stage, Qwen-Image uses additional fine-grained reinforcement learning with GRPO, following the Flow-GRPO framework [91]. Conditioned on the hidden state of the text prompt $p$ from the autoregressive module, the flow model generates a set of $G$ candidate images $\{x_0^i\}_{i=1}^{G}$ along

with their trajectories $\{x_T^i, x_{T-1}^i, \ldots, x_0^i\}_{i=1}^G$. The training objective of Flow-GRPO is consistent with Equation 4.2.

A key distinction from RL in autoregressive models is trajectory sampling. In Flow-GRPO, trajectories $\{x_T^i, \ldots, x_0^i\}_{i=1}^G \sim \pi_\theta$ are sampled according to the flow matching dynamics:

$$dx_t = v_t \, dt, \tag{4.3}$$

where $v_t = v_\theta(x_t, t, p)$ is the velocity predicted by the model. However, this deterministic formulation lacks stochasticity and thus fails to support adequate exploration. To address this, Flow-GRPO reformulates trajectory sampling as a stochastic differential equation (SDE), injecting randomness into the process:

$$dx_t = \left( v_t + \frac{\sigma_t^2}{2t}(x_t + (1-t)v_t) \right) dt + \sigma_t dw_t, \tag{4.4}$$

where $dw_t$ denotes Brownian motion and $\sigma_t$ controls the magnitude of randomness.

In addition to Flow-GRPO, DanceGRPO [95] extends GRPO to a broader range of visual generation tasks, including text-to-image, text-to-video, and image-to-video. It reformulates both diffusion sampling and rectified flows within the framework of stochastic differential equations, enabling GRPO to generalize across different architectures and training paradigms. MixGRPO [96] further improves computational efficiency by combining SDE and ODE based formulations.

### 4.3.3 Reward Models

Reward functions can be broadly divided into two categories. (1) Verifiable rewards, such as GenEval [69] for the composition of multiple objects and OCR-based evaluation for visual text ren-

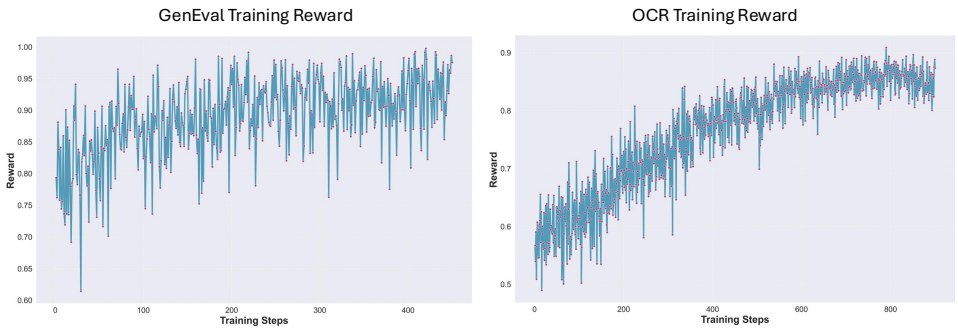



Figure 4.2: Training reward for GenEval and visual text rendering.

dering. (2) Model-based rewards, including PickScore [97], ClipScore [98], HPSv2.1 [99], ImageReward [100], and UnifiedReward [101], which assess image quality, image–text alignment, and human preference.

### 4.3.4 Experiments

In our experiments, we only apply RL for autoregressive model and focus on two verifiable reward tasks. (1) Multiple object composition: training prompts are adopted from Flow-GRPO [91], and evaluation is performed with the GenEval evaluator. (2) Visual text rendering: training prompts are also sourced from Flow-GRPO [91], and evaluation relies on PaddleOCR [102]. Figure 4.2 clearly illustrates the increasing reward trend throughout training. Figures 4.3 and 4.4 present qualitative comparisons before and after GRPO training, demonstrating noticeable improvements in both object composition and text rendering quality.

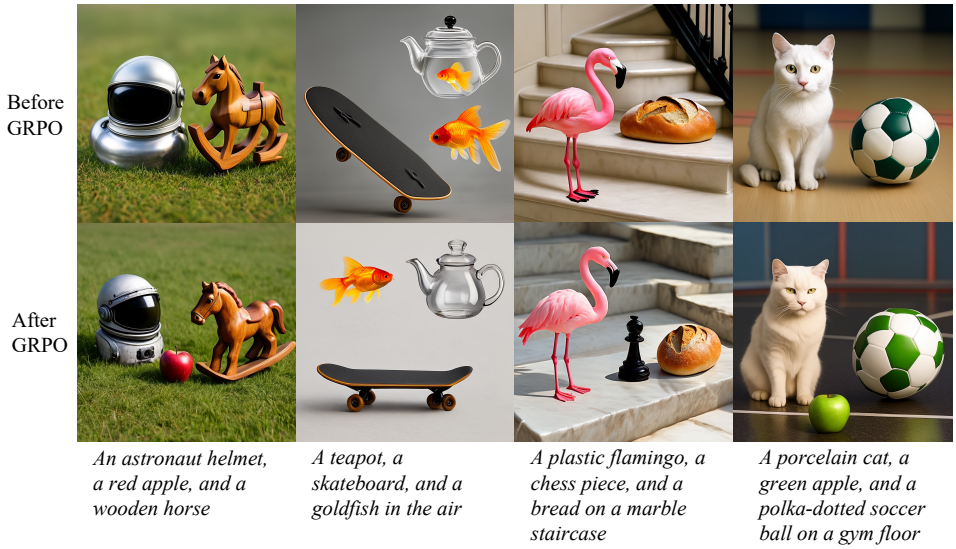



Figure 4.3: Qualitative results of multiple object composition before and after GRPO.


Figure 4.4: Qualitative results of text rendering before and after GRPO.

### 4.3.5 Discussion

The model architecture determines whether RL is applied to the autoregressive or diffusion model. For example, in Qwen-Image, the autoregressive module mainly serves as a text encoder, while the diffusion model performs most of the image generation. RL is more effective when applied to the diffusion model for Qwen-Image. In contrast, within the BLIP3o-NEXT framework, the autoregressive model is responsible for producing image tokens, and the diffusion model primarily functions as an image decoder. The autoregressive model plays a central role in image generation, making it the natural target for RL optimization. Empirically, without diffusion acceleration, applying RL to diffusion model is slower due to the lack of KV cache support and the need for multiple time steps.

The central challenge in applying reinforcement learning to native image generation lies in metric design rather than the RL algorithm itself. The key open problem is to develop reward models that can effectively capture and balance multiple dimensions, including image quality, instruction following, and human preference alignment.

## 4.4 Image Editing

To incorporate reference images as multimodal conditions for image generation, a common strategy is to feed them into the autoregressive backbone. Recent models such as MetaQuery [52], BLIP3-o [76], OmniGen2 [77] and Qwen-Image [79] are built on top of existing vision–language models such as Qwen-VL [71], which natively support image inputs as the conditions. Within the BLIP3o-NEXT framework, this is achieved by converting the reference image into quantized image tokens and concatenating them with the input text prompt tokens.

The primary challenge in image editing is maintaining consistency between the generated and

reference images. To address this issue, we adopt the following strategies to enhance consistency.

Image Reconstruction Task: we perform an image reconstruction task, where the reference image is provided as input and the text prompt is "*Keep the image unchanged.*". This task encourages the model to faithfully reconstruct visual details and align the generative process with the conditioning image.

Conditioning on VAE Latents: while the reference image is represented through a semantic vision encoder to autoregressive model, such representations often lack fine-grained pixel-level information. To address this limitation, we incorporate low-level, detail-preserving VAE latents as additional conditioning signals for the diffusion model. Specifically, we explore two complementary methods for integrating these VAE latents: (1) cross-attention conditioning and (2) noise-space injection, both designed to enhance visual consistency with respect to the reference image. Empirically, we find that combining both approaches yields the best consistency.

**(1) VAE features as cross-attention inputs in DiT.** After extracting the reference image's VAE features, we flatten the spatial dimensions (height and width) and project the channel dimension to match the cross-attention input size of the DiT. The resulting projected features are then concatenated with the multimodal context tokens produced by the autoregressive model. These concatenated tokens serve as the input to the DiT's cross-attention layers, enabling the model to incorporate both semantic and low-level cues, as shown in Figure 4.5 left.

**(2) VAE features as noise-space injection.** Alternatively, we concatenate the extracted VAE features with the diffusion noise tensor along the height dimension, effectively augmenting the noise input with adjacent reference image VAE features. This composite input is then fed into the DiT during denoising. After the model predicts the refined noise, we compute the loss only over the region

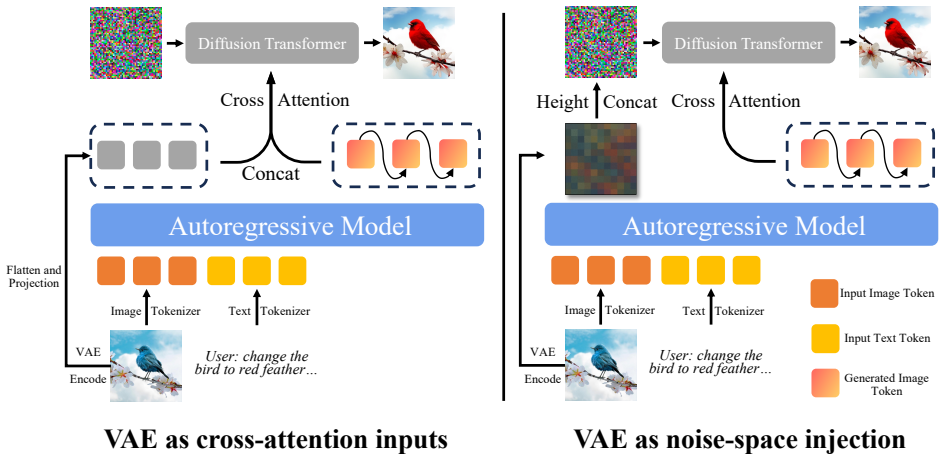



Figure 4.5: Comparison of VAE feature integration strategies in BLIP3o-NEXT. In the *VAE as cross-attention inputs* setup, the flattened VAE tokens are appended to the multimodal tokens produced by the autoregressive model. Empirically, we find that combining both methods yields the best visual consistency.

corresponding to the original random noise, using it as the flow-matching loss objective as shown in Figure 4.5 right.

### 4.4.1 Experiments

In our image editing experiments, we adopt a multitask learning setup that jointly trains the model on both image reconstruction and image editing objectives. The training data is curated from various open-source datasets, including BLIP3-o [76], OmniGen2 [77], ShareGPT4V [19], and Awesome-Nano-Banana [110]. To increase the data scale and stabilize training, we repeat selected subsets to construct a larger dataset ensemble. The final training corpus contains approximately 10 million samples, including repeated entries. Table 4.2 presents the results on the ImgEdit [1] benchmark, where our 3B model achieves performance comparable to larger models such as BAGEL and OmniGen2. Fig-

| **Model** | Add | Adjust | Extract | Replace | Remove | Background | Style | Hybrid | Action | Overall ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| MagicBrush [103] | 2.84 | 1.58 | 1.51 | 1.97 | 1.58 | 1.75 | 2.38 | 1.62 | 1.22 | 1.90 |
| Instruct-Pix2Pix [104] | 2.45 | 1.83 | 1.44 | 2.07 | 1.50 | 1.44 | 3.55 | 1.20 | 1.46 | 1.88 |
| AnyEdit [105] | 3.18 | 2.95 | 1.88 | 2.47 | 2.23 | 2.24 | 2.85 | 1.56 | 2.65 | 2.45 |
| OmniGen [106] | 3.67 | 3.39 | 1.71 | 2.94 | 2.43 | 3.41 | 4.19 | 2.24 | 3.38 | 2.93 |
| ICEdit [107] | 3.39 | 3.39 | 1.73 | 3.15 | 2.93 | 3.08 | 3.62 | 2.09 | 3.06 | 2.95 |
| StepX-Edit [108] | 3.88 | 3.14 | 1.76 | 3.40 | 2.41 | 3.16 | 4.63 | 2.64 | 2.52 | 3.06 |
| BAGEL [78] | 3.76 | 3.04 | 1.70 | 3.43 | 3.04 | 3.40 | 4.64 | 2.64 | 2.62 | 3.25 |
| UniWorld-V1 [109] | 3.82 | 3.64 | 2.27 | 3.47 | 3.24 | 2.99 | 4.21 | 2.96 | 2.74 | 3.26 |
| OmniGen2 [77] | 3.57 | 3.06 | 1.77 | 3.74 | 3.02 | 3.57 | 4.81 | 2.52 | 4.68 | 3.44 |
| FLUX.1 Kontext (Pro) [93] | 4.15 | 2.35 | 4.56 | 3.57 | 3.42 | 4.56 | 4.57 | 3.63 | 4.63 | 4.00 |
| GPT Image 1 [55] | 4.61 | 4.33 | 2.90 | 4.35 | 3.66 | 4.57 | 4.93 | 3.96 | 4.89 | 4.20 |
| Qwen-Image [79] | 4.38 | 4.16 | 3.43 | 4.66 | 4.14 | 4.38 | 4.81 | 3.82 | 4.69 | 4.27 |
| BLIP3o-NEXT (3B) | 4.00 | 3.78 | 2.39 | 4.05 | 2.61 | 4.30 | 4.64 | 2.67 | 4.13 | 3.62 |

Table 4.2: Image editing benchmark results for ImgEdit [1], with all metrics evaluated using GPT-4.1. The "Overall" score is computed as the average across all task categories. While our 3B model still lags behind GPT-Image and Qwen-Image, it achieves comparable performance to BAGEL and OmniGen2.

ure 4.6 illustrates qualitative results on consistency. The comparison between models with and without VAE latents demonstrates that incorporating VAE latents effectively enhances consistency. However, since the VAE in SANA uses a downsampling ratio of 32 to accelerate training and inference [80], the generated images still exhibit slight inconsistencies with the reference images.

### 4.4.2 Future Exploration

Besides image reconstruction and conditioning on VAE latents, we also want to highlight the following strategies:

Reinforcement Learning for Image Editing: applying RL to image editing offers a promising direction for improving instruction following and consistency between the generated output and the reference input. While model-based reward functions can effectively guide instruction following [111], the


Figure 4.6: Qualitative results for image editing comparing models with and without VAE latent condition.

design of reward models for measuring consistency remains relatively underexplored.

Designing System Prompts for Inpainting and Subject-Driven Generation: another avenue of exploration is the design of system prompts that explicitly distinguishes between inpainting and subject-driven generation tasks. Inpainting tasks prioritize spatial and background consistency, whereas subject-driven generation emphasizes maintaining consistency in the subject's appearance. Therefore, incorporating task-specific system prompts can effectively guide the model to handle these two categories of editing more appropriately.

Prompt Rewriting: in practice, we use prompt rewriting to enrich prompt details and improve instruction following abilities for both image generation and editing tasks.

While Autoregressive + Diffusion architectures enable strong instruction following, editing consistency still lags behind, even with VAE feature injected into the diffusion model. Bridging this gap calls for advances in data engineering and scaling reinforcement learning specifically tailored for image editing.

## 4.5 Training Recipe and Evaluation

Data quality remains a decisive factor in determining the overall performance of the model. Although different models have different data engine pipelines, they generally share several key components. (1) Diversity: We categorize image topics into domains such as Environments, Business, Cities, Food & Drink, Nature, Objects, Pets, Wildlife, and Lifestyle. The dataset integrates publicly available sources, including CC12M [14], SA-1B [6], and JourneyDB [66], supplemented with additional proprietary images. (2) Filtering: We perform extensive data cleaning; for example, we remove

extremely low-resolution or corrupted images, and we exclude samples containing watermarks, etc. (3) Captioning: Dense captions are generated using Qwen-VL-2.5 [71], samples with overly long captions (exceeding 120 tokens) or low CLIP-based image-text alignment scores are discarded. (4) Synthetic data: To enrich training diversity, we further construct synthetic datasets, particularly for text-rendering tasks, and distill data from frontier models.

Regarding the evaluation, although there are numerous benchmarks for evaluating image generation performance [69, 70, 75], there remains a significant need for more specialized benchmarks, particularly for image editing. Such benchmarks are essential for assessing a model's instruction following ability and the consistency between the generated images and the reference inputs.

## 4.6 Conclusion

In this work, we introduced BLIP3o-NEXT, a fully open-source foundation model that advances the frontier of native image generation by unifying text-to-image synthesis and image editing within a single architecture. Through extensive exploration, we identify four central insights: architectural simplicity coupled with scalability is often sufficient; reinforcement learning holds strong potential to further improve generation quality; post-training remains crucial for instruction following and editing consistency; and high-quality, large-scale data continues to set the performance ceiling. Building on these principles, BLIP3o-NEXT integrates the instruction following and reasoning capabilities of autoregressive modeling with the fine-grained rendering ability of diffusion, yielding coherent and high-fidelity visual outputs. Experimental results across diverse benchmarks demonstrate that BLIP3o-NEXT achieves superior performance in both generation and editing tasks, confirming the effectiveness of the Autoregressive + Diffusion paradigm.

Looking forward, we believe the findings presented here point toward promising directions for the next frontier of foundation models, where unified architectures, reinforcement learning, and scalable post-training jointly drive progress in controllable, instruction-aligned, and high-quality native image generation systems.

# Chapter 5: Related Work

LLMs have significantly advanced the development of MLLMs, including models like LLaVA [2], MiniGPT-4 [3], Qwen-VL [112], and Vila [42]. Most of these models integrate a language-supervised vision encoder, such as CLIP or SigLIP, with a language model backbone. Beyond these, there is a wider range of visual models available, including self-supervised models [4], segmentation models [6], and diffusion models [5]. Departing from conventional vision encoder designs, our work introduces an innovative approach by using the generative vision foundation model Florence-2 as the vision encoder.

While other studies, such as Cambrian [7], Brave [113] and MouSi [114] have explored the advantages of combining multiple visual signals, our approach avoids the added complexity and cost of using multiple vision encoders. Instead, we use a single vision model to generate multiple visual features, which each one emphasizing different perceptual information in the input image. This approach allows us to achieve superior performance with a single vision encoder, surpassing models that rely on multiple vision encoders, such as Cambrian [7].

Recent studies have highlighted unified multimodal, capable of both image understanding and generation, as a promising avenue of research. For example, SEED-X [46], Emu-2 [47], and Meta-Morph [51] train image features via regression losses, while Chameleon [53], Show-o [48], EMU3 [54], and Janus [49, 50] adopt an autoregressive discrete token prediction paradigm. In parallel, Dream-

LLM [115] and Transfusion [116] leverage diffusion objectives for visual generation. To our knowledge, we present the first systematic study of design choice in the autoregressive and diffusion framework. Regrading to the unifed model training strategy, LMFusion [59] builds on the frozen MLLM backbone while incorporating transformer modules for image generation using Transfusion [116]. Concurrent work MetaQuery [52] also uses learnable queries to bridge frozen pre-trained MLLMs with pre-trained diffusion models, but the diffusion models are in VAE + Flow Matching strategy instead of the more efficient CLIP + Flow Matching one in our BLIP3-o.

# Chapter 6: Conclusion and Future Work

## 6.1 Conclusion

This dissertation studied how to build a unified multimodal model family that can *perceive* visual content, *reason* over multimodal evidence, and *generate* (and ultimately *edit*) visual content. Across three stages of development, we advanced this goal with architectures and training recipes that systematically strengthen visual representations, scale native image generation, and improve instruction following and controllability through post-training.

First, we showed that stronger perception is a prerequisite for robust multimodal reasoning: rich, multi-level visual features and principled feature fusion better preserve fine-grained details required by OCR, grounding, and knowledge-intensive understanding. Second, we demonstrated that understanding and generation can be unified effectively when the model is designed around compatible representations and trained with stable recipes—including sequential training that preserves understanding while introducing generation capability. Third, we illustrated that post-training, including reinforcement learning, can further push generation and editing quality by optimizing for instruction adherence, consistency, and human-aligned preferences.

Overall, the dissertation contributes a coherent view of unified multimodal learning: strong perception enables faithful reasoning, and faithful reasoning enables controllable generation. The presented models, training strategies, and resources provide a practical path toward general-purpose mul-

timodal systems that can interpret the visual world and act on it through visual synthesis and editing.

## 6.2 Future Work

Despite rapid progress, unified multimodals remain an open frontier. We highlight several directions that can further advance perception, reasoning, and generation.

A key next step is developing representations that simultaneously support fine-grained recognition, long-range spatial relationships, and generation-friendly structure. Promising directions include hybrid continuous–discrete representations and multi-resolution representations that preserve both global layout and local details.

Reinforcement learning and preference optimization are powerful but can be data-hungry and brittle. A promising direction is combining RL with stronger automatic supervision: self-critique, self-distillation, and verified rewards derived from OCR, detection, and geometry constraints. Another direction is improving robustness and safety during post-training, including reducing reward hacking and maintaining broad generalization across prompts and domains.

Finally, unification must remain practical. Future work should explore model efficiency (token compression and caching), training efficiency (data curation and curriculum), and deployment constraints (latency and memory). The long-term goal is a unified system that is not only capable, but also scalable, reliable, and efficient enough for broad real-world adoption.

# Bibliography


[1] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[2] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[3] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[4] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[5] 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.

[6] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[7] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

[8] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024.

[9] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.

[10] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European conference on computer vision*, pages 74–92. Springer, 2022.

[11] Qidong Huang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Deciphering cross-modal alignment in large vision-language models with modality integration rate. *arXiv preprint arXiv:2410.07167*, 2024.

[12] Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. Large language model evaluation via matrix entropy. *arXiv preprint arXiv:2401.17139*, 2024.

[13] 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 (CVPR)*, pages 10684–10695, June 2022.

[14] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[15] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people, for the people. *arXiv preprint arXiv:2111.11431*, 2021.

[16] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.

[17] Vasu Singla, Kaiyu Yue, Sukriti Paul, Reza Shirkavand, Mayuka Jayawardhana, Alireza Ganjdanesh, Heng Huang, Abhinav Bhatele, Gowthami Somepalli, and Tom Goldstein. From pixels to prose: A large dataset of dense image captions, 2024.

[18] Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning, 2024.

[19] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023.

[20] HuggingFaceM4/Docmatix. https://huggingface.co/datasets/huggingfacem4/docmatix. `https://huggingface.co/datasets/HuggingFaceM4/Docmatix`, 2024.

[21] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[22] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[23] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[24] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[25] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[26] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[27] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[28] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024.

[29] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[30] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[31] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[32] Yuliang Liu, Zhang Li, Biao Yang, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. On the hidden mystery of ocr in large multimodal models, 2024.

[33] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[34] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

[35] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[36] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.

[37] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[38] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[39] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

[40] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.

[41] x.ai. Grok 1.5v: The next generation of ai. `https://x.ai/blog/grok-1.5v`, 2023. Accessed: 2024-07-26.

[42] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024.

[43] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[44] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[45] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

[46] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.

[47] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context

learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024.

[48] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[49] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[50] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[51] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.

[52] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

[53] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[54] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[55] Introducing 4o image generation. `https://openai.com/index/introducing-4o-image-generation/`, 2025.

[56] Zhiyuan Yan, Junyan Ye, Weijia Li, Zilong Huang, Shenghai Yuan, Xiangyang He, Kaiqing Lin, Jun He, Conghui He, and Li Yuan. Gpt-imgeval: A comprehensive benchmark for diagnosing gpt4o in image generation. *arXiv preprint arXiv:2504.02782*, 2025.

[57] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[58] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *CoRR*, abs/2209.03003, 2022.

[59] Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. Llamafusion: Adapting pretrained language models for multimodal generation. *arXiv preprint arXiv:2412.15188*, 2024.

[60] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[61] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[62] 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.

[63] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[64] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[65] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *CoRR*, abs/2210.02747, 2022.

[66] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, Jifeng Dai, Yu Qiao, Limin Wang, and Hongsheng Li. Journeydb: A benchmark for generative image understanding. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[67] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[68] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024.

[69] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[70] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[71] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[72] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Lirui Zhao, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. *arXiv preprint arXiv:2406.18583*, 2024.

[73] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[74] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. *arXiv preprint arXiv:2412.03069*, 2024.

[75] Yuwei Niu, Munan Ning, Mengren Zheng, Bin Lin, Peng Jin, Jiaqi Liao, Kunpeng Ning, Bin Zhu, and Li Yuan. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

[76] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[77] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[78] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[79] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[80] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformer, 2024.

[81] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[82] Rui Tian, Mingfei Gao, Mingze Xu, Jiaming Hu, Jiasen Lu, Zuxuan Wu, Yinfei Yang, and Afshin Dehghan. Unigen: Enhanced training & test-time strategies for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.14682*, 2025.

[83] Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

[84] Yanghao Li, Rui Qian, Bowen Pan, Haotian Zhang, Haoshuo Huang, Bowen Zhang, Jialing Tong, Haoxuan You, Xianzhi Du, Zhe Gan, et al. Manzano: A simple and scalable unified multimodal model with a hybrid vision tokenizer. *arXiv preprint arXiv:2509.16197*, 2025.

[85] Hao Tang, Chenwei Xie, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Unilip: Adapting clip for unified multimodal understanding, generation and editing. *arXiv preprint arXiv:2507.23278*, 2025.

[86] Jiaming Han, Hao Chen, Yang Zhao, Hanyu Wang, Qi Zhao, Ziyan Yang, Hao He, Xiangyu Yue, and Lu Jiang. Vision as a dialect: Unifying visual understanding and generation via text-aligned representations. *arXiv preprint arXiv:2506.18898*, 2025.

[87] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[88] Huiyang Shao, Xin Xia, Yuhong Yang, Yuxi Ren, Xing Wang, and Xuefeng Xiao. Rayflow: Instance-aware diffusion acceleration via adaptive flow trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18113–18123, 2025.

[89] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in Neural Information Processing Systems*, 37:117340–117362, 2024.

[90] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

[91] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[92] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[93] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[94] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[95] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[96] Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

[97] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

[98] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[99] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023.

[100] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[101] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[102] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025.

[103] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[104] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[105] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135, 2025.

[106] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025.

[107] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025.

[108] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[109] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

[110] awesome-nano-banana. `https://github.com/JimmyLv/awesome-nano-banana`, 2025.

[111] Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhu Chen. Editreward: A human-aligned reward model for instruction-guided image editing. *arXiv preprint arXiv:2509.26346*, 2025.

[112] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[113] Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. Brave: Broadening the visual encoding of vision-language models. *arXiv preprint arXiv:2404.07204*, 2024.

[114] Xiaoran Fan, Tao Ji, Changhao Jiang, Shuo Li, Senjie Jin, Sirui Song, Junke Wang, Boyang Hong, Lu Chen, Guodong Zheng, et al. Mousi: Poly-visual-expert vision-language models. *arXiv preprint arXiv:2401.17221*, 2024.

[115] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023.

[116] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.