

---

# Bifrost-1: Bridging Multimodal LLMs and Diffusion Models with Patch-level CLIP Latents

---

Han Lin<sup>1</sup> Jaemin Cho<sup>1</sup> Amir Zadeh<sup>2</sup> Chuan Li<sup>2</sup> Mohit Bansal<sup>1</sup>

<sup>1</sup>UNC Chapel Hill <sup>2</sup>Lambda

## Abstract

There is growing interest in integrating high-fidelity visual synthesis capabilities into large language models (LLMs) without compromising their strong reasoning capabilities. Existing methods that directly train LLMs or bridge LLMs and diffusion models usually suffer from costly training since the backbone LLMs have not seen image representations during pretraining. We present BIFROST-1, a unified framework that bridges pretrained multimodal LLMs (MLLMs) and diffusion models using patch-level CLIP image embeddings as latent variables, which are natively aligned with the MLLM’s CLIP visual encoder. These patch-level image embeddings are integrated into the diffusion model with a lightweight adaptation of its ControlNet. To retain the original multimodal reasoning capabilities of MLLMs, we equip the MLLM with a visual generation branch initialized from the original MLLM parameters when predicting the patch-level image embeddings. By seamlessly integrating pretrained MLLMs and diffusion models with patch-level CLIP latents, our framework enables high-fidelity controllable image generation with significant training efficiency. Our experiments demonstrate that BIFROST-1 achieves comparable or better performance than previous methods in terms of visual fidelity and multimodal understanding, with substantially lower compute during training. We also provide comprehensive ablation studies showing the effectiveness of our design choices. Project page: <https://bifrost-1.github.io>.

## 1 Introduction

Humans can naturally process and generate multimodal information such as language, vision, and sound. This ability stems from our integrated cognitive system capable of reasoning, organization, and expression. Inspired by this, building a unified AI system that supports both multimodal understanding and generation has become an active area of research [48, 10, 80, 62, 73, 68]. On the one hand, large language models (LLMs) have achieved impressive results across diverse tasks, including natural language understanding [17, 40, 6, 54], mathematical reasoning [33, 58], and code generation [9, 46], demonstrating strong planning and reasoning abilities [49, 26]. On the other hand, recent advances in image and video generation models have enabled high-fidelity visual synthesis [50, 74, 19, 47]. Therefore, effectively and efficiently integrating visual generation capabilities into LLMs, while preserving their reasoning capabilities, has become an active research direction in the pursuit of unified AI systems that bridges language understanding and visual synthesis.

Existing LLM-based image generation methods fall into two main paradigms. First, single-architecture approaches (see Fig. 1 (a)) employ an LLM/MLLM that takes concatenated image and text tokens as inputs. The model is then finetuned, either by updating all LLM parameters [12, 67, 42, 41, 62, 71, 10, 68] or via an auxiliary visual generation branch [44, 37, 59], to generate images directly. While this offers unified reasoning and native image synthesis, it typically incurs very expensive computational costs to achieve high-fidelity results. Moreover, without careful

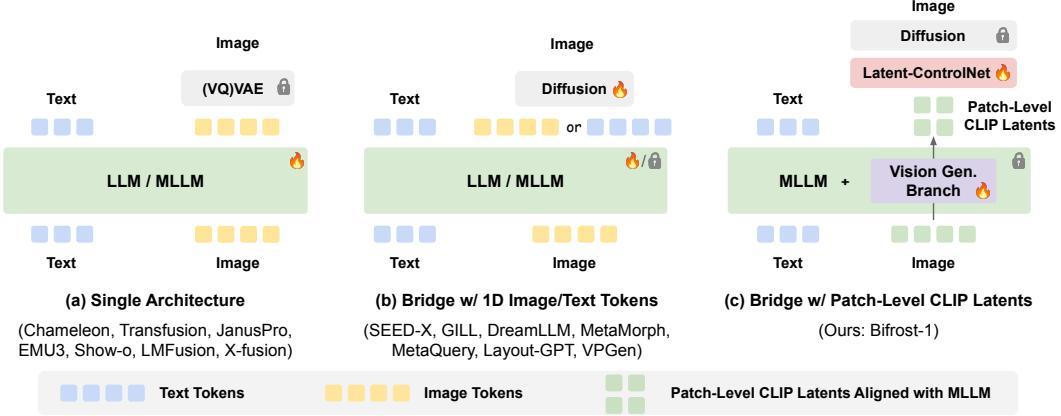


Figure 1: **Comparison of different approaches in using LLM for image generation.** (a) Single architecture handling both text and image tokens into the LLM/MLLM. (b) Bridging LLM and Diffusion model with a 1D sequence (image tokens or text tokens). (c) BIFROST-1 (ours), which bridges MLLM with diffusion models with 2D image tokens aligned with MLLM embeddings.

balancing of training data and objectives, such finetuning can risk degrading the LLM’s original language reasoning capabilities [71], thereby undermining a primary goal of the integration.

Second, bridging-based approaches (see Fig. 1 (b)) use both pretrained diffusion models as well as LLMs, teaching an LLM to guide a diffusion model in generating the final image. The guidance takes the form of detailed text descriptions (e.g., object layouts, extended prompts) generated by the LLM [13, 20, 14], or continuous query tokens also produced by the LLM [31, 23, 63, 51, 1]. Although generally more training-efficient than single-architecture methods, these approaches have their own limitations. Text-based representations can be too sparse to capture fine-grained details of complex scenes, while learning to generate effective 1D continuous query tokens often requires a large connector and end-to-end training, and may not sufficiently convey detailed 2D spatial information.

To address these limitations, we introduce **BIFROST-1** (Sec. 3). As illustrated in Fig. 1(c), BIFROST-1 uniquely bridges a pretrained multimodal LLM (MLLM) [5] and a diffusion model [32]. Our core innovation lies in using 2D patch-level CLIP [53] latents as the communicative medium. These latents are image embeddings that are natively aligned with the MLLM’s own CLIP visual encoder, enabling the MLLM to generate rich and precise spatial guidance for image synthesis. These patch-level latents are integrated into the diffusion model via a lightweight adaptation of its ControlNet [78]. To retain the original multimodal reasoning capabilities of MLLMs, we equip the MLLMs with a visual generation branch initialized from the original MLLM parameters when predicting the patch-level image embeddings. By seamlessly integrating pretrained MLLMs and diffusion models with patch-level CLIP latents, our framework enables high-fidelity controllable image generation with substantial training efficiency. In addition, by fully retaining the original parameters of the MLLM backbone, BIFROST-1 does not suffer from performance degradation of multimodal reasoning capabilities observed in some previous works.

We empirically demonstrate the effectiveness of BIFROST-1 through a series of experiments. First, in Sec. 5.1, we compare different architectures for bridging LLMs and diffusion models, showing that patch-level CLIP latents train more efficiently than the controlled variants with alternative methods. Next, in Sec. 5.2, we present the effective scaling of the number of CLIP latent tokens to improve image reconstruction quality. We also qualitatively demonstrate that the image reconstruction quality of BIFROST-1 is competitive or superior to strong baselines with a magnitude higher training computation cost. In Sec. 5.3, we compare BIFROST-1 with state-of-the-art methods on multimodal understanding and generation tasks, showcasing its competitive performance and the preservation of the MLLM backbone’s reasoning capabilities. Finally, in Sec. 5.4, we experiment with varying number of MLLM decoding steps, and observe that our method is robust as long as the number of decoding steps is larger than 8. In a nutshell, our contribution can be summarized as follows:

- We propose a novel framework (BIFROST-1) for unified multimodal generation and understanding by bridging pretrained MLLM and diffusion models with fine-grained patch-level CLIP latents.

- By using CLIP latents that are natively aligned with the MLLM, we achieve significantly lower training costs compared to recent works with a single architecture or those that bridge using 1D latents or query tokens.
- We show qualitatively and quantitatively that BIFROST-1 achieves comparable performance with previous SoTAs on image reconstruction quality and text-to-image generation benchmarks, while well-preserving MLLM’s original multimodal understanding ability.

## 2 Related Work

**Unified multimodal LLM architectures for visual generation.** The success of large language models (LLMs) such as T5 [54] and GPT-3 [6] has driven efforts toward extending language modeling paradigms to multimodal settings. Early works like VL-T5 [11], SimVLM [69], and Flamingo [3] adapt pretrained LLMs for visual understanding by framing perception tasks as text generation. More recent models also incorporate image generation capabilities into LLMs using either autoregressive objectives [12, 67, 42, 41, 62, 71, 10, 68, 59, 44] or denoising diffusion objective [73, 80], as described in Fig. 1 (a). While these approaches enable flexible, end-to-end training, they often inherit their initialization from text-only LMs and require significant resources to learn high-quality image generation from scratch. This limitation has motivated a complementary line of research: bridging pretrained LLMs with diffusion models, as discussed next below.

**Bridging LLM and diffusion models for visual generation.** As described in Fig. 1 (b), several methods decompose the generation process by using LLMs to produce detailed intermediate textual representations, such as object layouts or detailed scene descriptions, which guide a pretrained diffusion model [13, 20, 14]. However, complex visual scenes with rich object interactions are often hard to describe purely with text. An alternative strategy is to replace intermediate text with continuous visual representations to guide a diffusion model [31, 23, 63, 51]. For example, Tong et al. [63] starts with an LLM and conduct large-scale training to equip it with a SigLIP [77] visual encoder. The image tokens generated autoregressively by the LLM are supervised using a cosine similarity loss with ground-truth SigLIP embeddings. These predicted image tokens are then injected into diffusion models via cross-attention. However, teaching LLMs to generate visual representations still requires more text-image annotations and computational resources (see Table 1), and it is also prone to degraded text-generation quality unless the model is carefully balanced and trained on high-quality text-text and vision-text data, along with extensive compute. Hence, we explore the use of multimodal LLMs (MLLMs) that are already strongly aligned with CLIP embeddings encoding spatial information, by teaching them to generate 2D CLIP embeddings to guide diffusion models through latent ControlNet. As shown in Sec. 5.1, we find our method of bridging MLLM and Diffusion models with 2D CLIP embeddings and latent ControlNet is significantly more data-efficient in broad training budget scenarios compared to the above baselines.

## 3 BIFROST-1

We introduce **BIFROST-1**, a novel unified multimodal framework that effectively bridges pretrained MLLMs and diffusion models with patch-level CLIP latents which are natively aligned with the MLLM. This enables MLLMs to generate rich and precise spatial guidance for image synthesis while achieving significant training efficiency. In the following, we present preliminaries about MLLMs and ControlNet (Sec. 3.1), how BIFROST-1 bridges MLLMs and diffusion models via patch-level CLIP latents (Sec. 3.2), and training and inference strategy (Sec. 3.3).

### 3.1 Preliminaries

**Visual encoding and decoding with MLLMs.** For image understanding tasks, given an RGB image  $x \in \mathbb{R}^{3 \times H \times W}$ , recent MLLMs commonly employ visual encoders that convert the image into a set of  $d$ -dimensional continuous (e.g., CLIP [53]) or discrete (e.g., VQVAE [64]) visual embeddings:  $z = \mathcal{E}(x) \in \mathbb{R}^{d \times H' \times W'}$ , where  $H' < H$  and  $W' < W$  due to spatial downsampling [5, 81, 2, 15]. These visual embeddings are aligned with the LLMs’ embedding space via learned projection layers. For image generation tasks, to enable the MLLM to produce images, the visual embeddings generated by the LLM backbone are passed to a visual generation decoder  $\mathcal{D}$  to render the final image:  $x = \mathcal{D}(z)$ . Typically, VQVAEs or diffusion models are employed for the decoder, whose visual

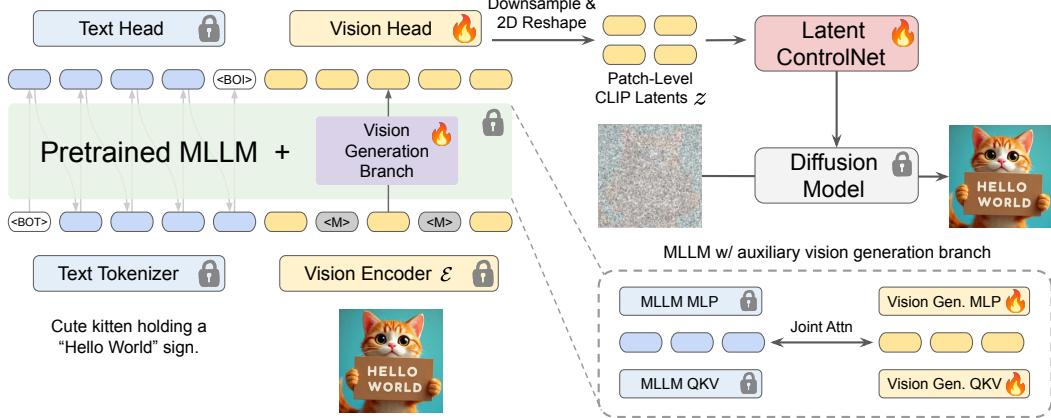


Figure 2: **Overview of BIFROST-1.** BIFROST-1 equips the backbone MLLM with a visual generation branch, which is a trainable copy of a pretrained MLLM parameters (i.e., QKV, MLP, normalization layers) and a newly added vision head (i.e., a linear layer). The visual generation branch outputs patch-level CLIP latents, which are then downsampled and reshaped into 2D (HxW), provided to latent ControlNet, and finally guiding image generation of a pretrained diffusion model. During training, a portion of the image patches is randomly replaced with learnable mask tokens  $\langle M \rangle$ . During inference, we start with fully masked image tokens and autoregressively predict them.

representation is different from CLIP embeddings; thus, learning to align MLLMs and decoders usually requires significant computation. In this work, we extend unCLIP [55] that employs a model to convert a single (pooled) CLIP text embedding to a single (pooled) CLIP image embedding by teaching MLLM to generate patch-level CLIP latents which are natively aligned with the visual encoder  $\mathcal{E}$  of the pretrained MLLM. We find this significantly facilitates image generation training.

**ControlNets.** ControlNet [79] is designed to add spatial controls with image guidance inputs (e.g., depth, sketch, segmentation maps, *etc.*) to diffusion models. Specifically, given a pretrained diffusion model  $\mathcal{F}_\theta$ , ControlNet adopts a similar architecture  $\mathcal{F}_{\theta'}$ , whose parameters are initialized from  $\theta$  (i.e., ‘trainable copy’). ControlNet takes the diffusion timestep  $t$ , text prompt  $c_{\text{text}}$ , control image  $c_f$  (e.g., depth map), and the noisy latents  $z_t$  as inputs, and outputs features guide the backbone diffusion  $\mathcal{F}_\theta$  for the final image generation. In this work, we propose latent ControlNet that uses patch-level CLIP image embeddings instead of a control image to guide diffusion models.

### 3.2 Bridging MLLMs and Diffusion Models with Patch-Level CLIP Latents

**BIFROST-1 design summary.** We have two main goals: (1) to preserve the multimodal understanding capability of the MLLM, while (2) efficiently teaching MLLM to generate latent tokens to guide diffusion models. As illustrated in Fig. 2, BIFROST-1 achieves these goals by having a new visual generation branch initialized from the original MLLM parameters and using patch-level CLIP image embeddings as latent bridge tokens.

**Learning to unmask image patch embeddings with a visual generation branch.** To teach an MLLM image generation, we first encode the images using the MLLM’s native visual encoder to get patch-level image embeddings and concatenate them with text tokens. Following MAR [36], we replace parts of the input image embeddings with a learnable mask token ( $\langle M \rangle$ ) and let the MLLM to predict the masked image embeddings. The mask ratio is randomly sampled from a truncated normal distribution with a mean of 1.0 and a standard deviation of 0.25, constrained to the range [0.7, 1.0]. For this image embedding prediction task, we introduce a visual generation branch (Fig. 2 left), whose parameters are initialized from MLLM parameters (i.e., attention QKV projections, MLP projection layers, and normalization layers) following LMFusion [59]. Just like text head, which is a linear layer toward text embedding space, we use a simple linear layer as a vision generation head. By reusing majority of parameters from the pretrained MLLM and randomly initializing only a single linear layer as the vision head, we avoid the costly process of realigning image embeddings.

**Attention across modalities.** In Fig. 3, we illustrate how different image and text input tokens can attend to each other. In summary, we apply causal masking for text, and we apply full attention for image tokens. All the previous modalities are fully visible to future modality tokens.

**Latent ControlNet.** To effectively guide diffusion models with patch-level CLIP image embeddings, we create latent ControlNet (Fig. 2 top right), by modifying the original ControlNet architecture from a backbone image diffusion model (*e.g.*, FLUX.1-dev [32]) with the following two changes:

- We replace the input linear projection layer ( $\mathbb{R}^{3 \times d} \rightarrow \mathbb{R}^{d' \times d}$ ), where  $d$  is the embedding dimension of the backbone image diffusion model, and  $d'$  is the embedding dimension of the MLLM.
- To further reduce the number of visual tokens generated by the MLLM, we introduce a lightweight downsampling 2D convolution block to reduce the spatial resolution of the visual embeddings by a factor of 2 before passing to latent ControlNet.

During training, we update only the newly added input linear projection, the 2D downsampling convolution blocks, and the ControlNet for 4 MM-DiT blocks and 1 Single-DiT block, compared to the full FLUX.1-dev model which contains 19 MM-DiT blocks and 38 Single-DiT blocks in total.

### 3.3 Training and Inference

**Decoupled training.** The MLLM visual prediction branch and latent ControlNet can be jointly trained or separately. We adopt the second, a decoupled training strategy for the following two reasons: (1) Since we only train small parameters of latent ControlNet, it converges much faster than the MLLM’s vision branch. In initial experiments, we find that decoupled training allows us to allocate more compute to the MLLM’s vision branch, improving overall performance. (2) Since recent diffusion models have large parameters (*e.g.*, FLUX.1-dev has 12B parameters), end-to-end training significantly increases memory requirements. For the MLLM visual prediction branch, we use the mean squared error (MSE) loss for image patch embedding prediction. For the latent ControlNet, we use the original flow-matching loss used in FLUX ControlNet.

**Inference of image patch embeddings tokens via masked autoregression.** During inference, the MLLM is given a text prompt along with fully masked CLIP image embeddings. Following MAR [36], we first randomly sample a generation order of image patches, then iteratively denoise them into clean CLIP image embeddings. Once all image patches are obtained, we give them latent ControlNet to guide the diffusion model.

## 4 Experiment Setup

**Latent ControlNet implementation details.** For the latent ControlNet design, we build upon the official FLUX.1-dev ControlNet training implementation from the Hugging Face Diffusers library.<sup>1</sup> Specifically, we train a ControlNet consisting of 4 MM-DiT [19] (DoubleStream) blocks and 1 Single-DiT [52] (SingleStream) block (*i.e.*, `num_double_layers=4`, `num_single_layers=1`) with a total batch size of 48. All other training hyperparameters, including learning rate, Adam optimizer, and weight decay, are kept identical to the original codebase without any tuning. During latent ControlNet inference, we also retain all default hyperparameters unchanged (*e.g.*, `num_inference_steps=28`, `controlnet_conditioning_scale=0.7`, `guidance_scale=3.5`).

**Baselines.** For ImageNet experiments, we compare BIFROST-1 with two main variants, using the same model initialization and computational budget: (1) the first variant replaces the MLLM-aligned

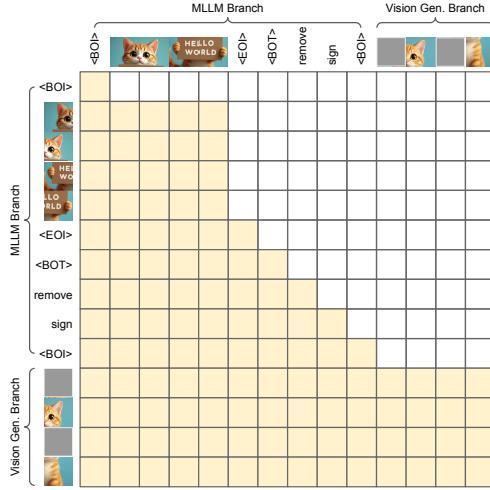


Figure 3: Attention mask. X-axis is input and Y-axis is output.

<sup>1</sup><https://github.com/huggingface/diffusers/tree/main/examples/controlnet>

Table 1: Comparison of different architectures for bridging LLMs and diffusion models in image generation on ImageNet 256×256.

Method		FID↓	sFID↓	IS↑
Backbone diffusion model				
FLUX.1-dev [32]		51.20	224.82	<b>157.46</b>
2D Learnable query tokens instead of CLIP latent				
MLLM + <b>2D Learnable Query Tokens</b> + Latent ControlNet + FLUX.1-dev		118.69	129.14	9.15
No pre-aligned visual encoder instead of pre-aligned visual encoder				
MLLM + <b>SigLIP</b> + Latent ControlNet + FLUX.1-dev		274.16	304.94	2.69
VAE instead of CLIP latent				
MLLM + <b>FLUX VAE</b> + Latent ControlNet + FLUX.1-dev		284.51	361.45	1.11
Cross-attention instead of Latent ControlNet				
MLLM + Patch-level CLIP Latent + <b>cross-attention</b> + FLUX.1-dev		76.32	208.11	26.35
BIFROST-1 (Ours)				
<b>MLLM + Patch-level CLIP Latent</b> + Latent ControlNet + FLUX.1-dev		<b>25.77</b>	<b>53.67</b>	98.57

CLIP-based embeddings with non-MLLM-aligned FLUX VAE features for visual generation, and (2) the second variant uses 2D query tokens (extended from MetaQuery [51]) instead of masked autoregression for visual generation. For SoTA comparison experiments, we compare our method with baselines for unified multimodal understanding and generation including DreamLLM [18], Chameleon [62], Show-o [73], VILA-U [72], EMU3 [68], MetaMorph [63], MetaQuery [51], Token-Flow [24], Transfusion [80], LMFusion [59], Janus [71], JanusFlow [43], and JanusPro [10].

**Training details.** We train BIFROST-1 on ImageNet [16] for the experiments in Sec. 5.1 and Sec. 5.2. Specifically, the latent ControlNet and BIFROST-1 MLLM in Sec. 5.1 are trained for 2 epochs and 16 epochs respectively, and the latent ControlNet in Sec. 5.2 is trained for only 1 epoch (~25M training steps). For SoTA comparison experiments in Sec. 5.3, we measure the training cost by the total number of training images passed to the model, calculated as the product of the number of training steps and the batch size per step. Our models with Qwen2.5-VL 3B/7B are trained on 9M and 62M images respectively from the BLIP3-o [8] training dataset.<sup>2</sup> All experiments on ImageNet (Sec. 5.1 and Sec. 5.2) are trained on a single GH200 GPU, and the SoTA comparison experiments in Sec. 5.3 are trained on 16 GB200 GPUs.

**Evaluation datasets and metrics.** For ImageNet, following previous works [36, 56], we evaluate our model on Fréchet Inception Distance (FID) [27], sFID [45], and Inception Score (IS) [57]. Then for open prompt evaluation, we follow previous works [10, 51, 63, 59] and report FID scores on MJHQ-30K [35] and 30k randomly sampled images from MSCOCO [38] validation set for visual aesthetic quality, and GenEval [25] and DPG-Bench [28] for prompt alignment, respectively. We use the original prompts from all these four evaluation datasets directly without any prompt rewriting.

## 5 Results and Discussion

We empirically demonstrate the effectiveness of BIFROST-1 in various experiments. Specifically, we compare different architectures for bridging LLMs and diffusion models (Sec. 5.1), training efficiency of different numbers of CLIP latent tokens for image reconstruction (Sec. 5.2), BIFROST-1 with SoTAs on multimodal understanding and generation tasks (Sec. 5.3), and experiments with different MLLM decoding steps (Sec. 5.4).

### 5.1 Comparison of Design Choices for Bridging LLMs and Diffusion Models

We compare different choices for bridging LLMs and diffusion models in terms of image generation task on ImageNet 256x256. We generate 10K images with classes randomly sampled from 1k categories and compute visual quality metrics. In all settings, the MLLM visual generation branch and the Latent ControlNet are trained for 16 and 2 epochs, respectively.

<sup>2</sup><https://huggingface.co/datasets/BLIP3o/BLIP3o-Pretrain-Long-Caption>

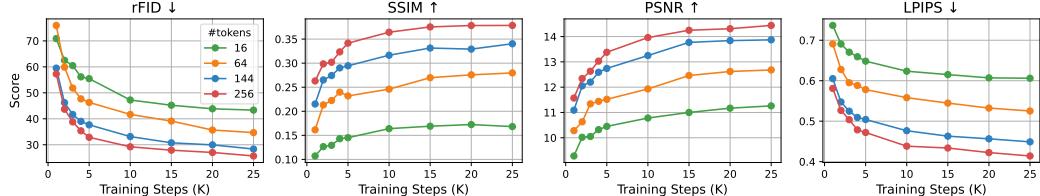


Figure 4: Image reconstruction scores with different numbers of 2D CLIP latent tokens used within BIFROST-1 on ImageNet for around one training epoch ( $\sim 26$ K steps). Results indicate that using more tokens achieves better data efficiency.

**BIFROST-1 vs. Backbone diffusion model.** In the first and last rows of Table 1, we compare BIFROST-1 and its backbone diffusion model (FLUX.1-dev [32]). We find that BIFROST-1 improves FID and sFID, while it hurts IS from the original backbone. The improvement over the baseline mainly comes from adding a few trainable ControlNet blocks to the backbone diffusion model, which enables better adaptation to the data distribution.

**Patch-level CLIP latent vs. 2D learnable query tokens.** MetaQuery [51] is a recent method that bridges LLMs and diffusion models by learning a connector (24-layer transformer encoder) that projects a frozen LLM’s hidden representations on a finite number of learnable query tokens, where the connector outputs guide a diffusion model via cross attentions. Inspired by this, we implement a 2D version of MetaQuery and compare it with BIFROST-1. To fairly compare methods with similar additional parameters, instead of learning a heavy connector module, we directly reshape the LLM representations of learned query tokens as inputs to the latent ControlNet. As shown in Table 1, 2D learnable query tokens hurt performance in all 4 metrics, indicating that learning to align representations between MLLM and diffusion from scratch requires much more computation than our patch-level CLIP latent.

**Patch-level CLIP latents vs. VAE latents.** Here, we compare BIFROST-1 with a variant that replaces CLIP latents (that are natively aligned with MLLM) with VAE latents (that are not originally aligned with MLLMs). Specifically, we substitute the CLIP visual encoder with the FLUX VAE encoder, and replace our visual decoder (i.e., Latent ControlNet + FLUX diffusion model) with the FLUX VAE decoder. Linear projection layers are applied to align the dimensions of the VAE-encoded features with the feature dimension of the MLLM backbone. All other components of our framework and the training strategy are kept the same to ensure a fair comparison. As shown in the 3rd row of Table 1, using VAE features significantly slows down learning. Additionally, this highlights that by leveraging a pretrained diffusion model as the image renderer, BIFROST-1 effectively reduces the burden on the MLLM side to directly generate high-quality images.

**MLLM’s native visual encoder vs. non-aligned external visual encoder.** In addition, we also compare using the latents from MLLM’s native visual encoder (i.e., CLIP) with an external visual encoder (i.e., SigLIP, as used in MetaMorph [63]). As we can see in the 4th row of Table 1, although using SigLIP achieves better image generation quality than using VAE, it is still significantly worse than using the MLLM’s native visual encoder. This indicates the training efficiency of adopting MLLM’s native visual encoder compared with non-aligned external visual encoders.

**Diffusion model guidance strategy: Latent ControlNet vs. Cross-attention** Our method directly injects 2D image tokens generated by the MLLM into the DiT via latent ControlNet. By adding 2D ControlNet latents on top of the noisy DiT latents, our method can enforce spatial structures more explicitly and effectively. In contrast, several previous works [63, 51, 8] use cross-attention to condition on image tokens. We conducted an additional experiment comparing these two conditioning strategies. For a fair comparison, we unfreeze all parameters in DiT when conditioning it via cross-attention and use 64 MLLM-generated image tokens for both methods. As shown in the 5th row of Table 1, our latent ControlNet achieves better image generation quality (FID, sFID, IS), demonstrating its superior efficiency and effectiveness.

## 5.2 Image Reconstruction with Patch-level CLIP Latent

In the following, we demonstrate the effectiveness of CLIP Latent-based image representation in image reconstruction experiments. Concretely, we experiment with scaling the number of latent tokens and compare BIFROST-1 with SoTA models.



Figure 5: Visual samples for image reconstruction with different numbers of patch-level CLIP tokens generated from MLLM. The Latent ControlNet models with varying numbers of tokens are trained for only 1 epoch on the ImageNet training split.

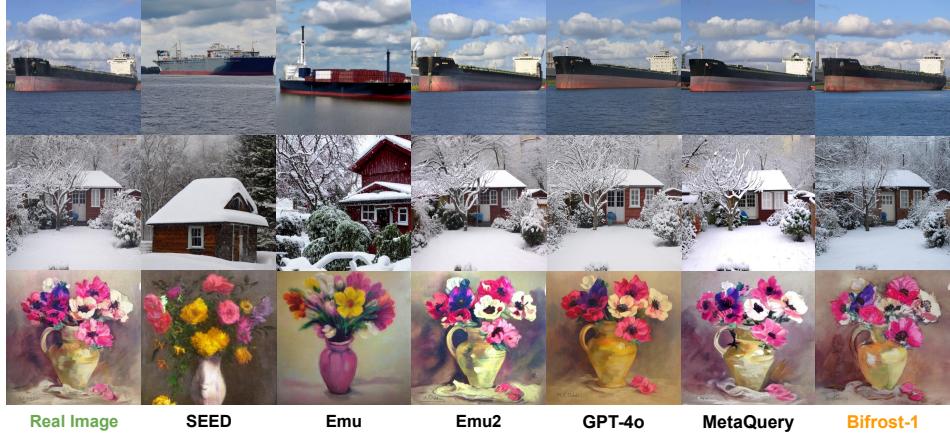


Figure 6: Image reconstruction results.

**Scaling the number of CLIP latent tokens.** In Fig. 4 we show that image reconstruction quality scales well with the number of patch-level CLIP latent tokens input to the Latent ControlNet. With only 1 epoch of training ( $\sim 26K$  steps) on the ImageNet dataset, images represented by 256 ( $= 14 \times 14$ ) CLIP latent tokens not only achieve higher reconstruction accuracy, measured by rFID, SSIM, PSNR, and LPIPS, but also converge faster compared to those using fewer CLIP latent tokens. In Fig. 5, we show qualitative examples of images generated with different numbers of CLIP latent tokens.

**Comparison with SoTA methods.** In Fig. 6, we qualitatively compare the image reconstruction quality of our latent ControlNet with various unified models, including SEED [22], EMU [61], EMU2 [60], GPT-4o [29], and MetaQuery [51]. Specifically, we first encode the images using the visual encoder in Qwen2.5-VL, then use the latent ControlNet to reconstruct the images based solely on this visual information, without providing any text prompt as additional guidance. Trained exclusively on the ImageNet dataset for 3 epochs without exposure to any other open-world images, the reconstructions from the BIFROST-1 latent ControlNet achieve quality that is competitive with or superior to strong baselines such as GPT-4o and MetaQuery, demonstrating the efficiency and robustness of our Latent ControlNet design.

### 5.3 Comparison with SoTA Models

**BIFROST-1 fully inherits strong visual understanding capabilities of backbone MLLM.** As BIFROST-1 keeps the original parameters in the MLLM completely frozen, the trainable vision branch is agnostic to the choice of MLLM. This allows BIFROST-1 to fully inherit the visual understanding

Table 2: Comparison with state-of-the-art methods on multimodal understanding benchmarks.

Method	Base (M)LLM	MME-P↑	MMB↑	SEED↑	MMMU↑	MM-Vet↑
DreamLLM [18]	Vicuna 7B	-	-	-	-	36.6
Chameleon [62]	From Scratch 7B	-	-	-	22.4	8.3
Show-o-512 [73]	Phi-1.5 1.3B	1097.2	-	-	26.7	-
VILA-U [72]	LLaMA-2 7B	1401.8	-	59.0	-	33.5
EMU3 [68]	From Scratch 7B	-	58.5	68.2	31.6	37.2
MetaMorph [63]	LLaMA-3 8B	-	75.2	71.8	-	-
MetaQuery-L [51]	Qwen2.5-VL 3B	1574.3	78.6	73.8	53.1	63.2
MetaQuery-XL [51]	Qwen2.5-VL 7B	1685.2	83.5	76.9	58.6	66.6
TokenFlow-XL [24]	Qwen-2.5 14B	1551.1	76.8	72.6	43.2	48.2
Transfusion [80]	From Scratch 7B	-	-	-	-	-
LMFusion [59]	LLaVA-Next 8B	1603.7	72.1	72.5	41.7	-
Janus [71]	DeepSeek-LLM 1.5B	1338.0	69.4	63.7	30.5	34.3
JanusFlow [43]	DeepSeek-LLM 1.5B	1333.1	74.9	70.5	29.3	30.9
JanusPro-1B [10]	DeepSeek-LLM 1.5B	1444.0	75.5	68.3	36.3	39.8
JanusPro-7B [10]	DeepSeek-LLM 7B	1567.1	79.2	72.1	41.0	50.0
BLIP3-o 4B [8]	Qwen2.5-VL 3B	1527.7	78.6	73.8	46.6	60.1
BLIP3-o 8B [8]	Qwen2.5-VL 7B	1682.6	83.5	77.5	50.6	66.6
<b>BIFROST-1 (Ours)</b>	<b>Qwen2.5-VL 7B</b>	<b>1685.2</b>	<b>83.5</b>	<b>76.9</b>	<b>58.6</b>	<b>66.6</b>

Table 3: Comparison with state-of-the-art methods on multimodal generation benchmarks.

Method	Base (M)LLM	Train Steps×B.S.	COCO FID↓	MJHQ FID↓	GenEval↑	DPG-Bench↑
EMU [68]	LLaMA 13B	-	11.66	-	-	-
DreamLLM [18]	Vicuna 7B	-	8.46	-	-	-
Chameleon [62]	From Scratch 7B	-	26.74	-	0.39	-
Show-o-512 [73]	Phi-1.5 1.3B	-	9.24	15.18	0.68	-
VILA-U [72]	LLaMA-2 7B	-	-	7.69	-	-
EMU3 [68]	From Scratch 7B	-	12.80	-	0.66	80.60
MetaMorph [63]	LLaMA-3 8B	-	11.8	-	-	-
MetaQuery-L [51]	Qwen2.5-VL 3B	-	8.87	6.35	0.78	81.10
MetaQuery-XL [51]	Qwen2.5-VL 7B	200M	8.69	6.02	0.80	82.05
TokenFlow-XL [24]	Qwen-2.5 14B	-	-	-	0.63	73.38
Transfusion [80]	From Scratch 7B	-	8.70	-	0.63	-
LMFusion [59]	LLaVA-Next 8B	-	8.20	-	-	-
Janus [71]	DeepSeek-LLM 1.5B	100M	8.53	10.10	0.61	-
JanusFlow [43]	DeepSeek-LLM 1.5B	211M	-	9.51	0.63	80.09
JanusPro-1B [10]	DeepSeek-LLM 1.5B	200M	-	14.33	0.73	82.63
JanusPro-7B [10]	DeepSeek-LLM 7B	194M	-	13.48	0.80	84.19
BLIP3-o 4B [8]	Qwen2.5-VL 3B	-	-	-	0.81	79.36
BLIP3-o 8B [8]	Qwen2.5-VL 7B	-	-	-	0.84	81.60
<b>BIFROST-1 (Ours)</b>	<b>Qwen2.5-VL 3B</b>	<b>9M</b>	<b>23.02</b>	<b>15.24</b>	<b>0.61</b>	<b>76.41</b>
<b>BIFROST-1 (Ours)</b>	<b>Qwen2.5-VL 7B</b>	<b>62M</b>	<b>34.35</b>	<b>16.21</b>	<b>0.81</b>	<b>77.67</b>

capabilities of the MLLM and easily scale to backbone MLLMs with larger sizes and stronger performance. Table 2 compares the Qwen2.5-VL 7B [5] model with other unified models on image understanding benchmarks, including MME-P [21], MMB [39], SEED [34], MMMU [76], and MM-Vet [75]. Baseline models such as JanusPro [10] and MetaMorph [63] fully fine-tune the LLM backbone and therefore tend to lose the original planning and reasoning abilities that these LLMs acquired through large-scale text pretraining.

**BIFROST-1 achieves competitive visual generation quality.** In Fig. 7, we compare BIFROST-1 with the recent SoTA methods (e.g., JanusPro and MetaQuery) on MJHQ30k prompts. Table 3 compares BIFROST-1 with other unified models on image generation benchmarks, including COCO and MJHQ for visual quality and GenEval and DPG-Bench for prompt following ability. Trained only with 25M image-text pairs for less than two epochs, BIFROST-1 matches the performance with models trained with much higher compute, including Janus on GenEval benchmark, and outperforms TokenFlow-XL on DPG-Bench. In addition, we would like to highlight that FID scores heavily depend on the choice of diffusion backbones. As also observed in MetaQuery[51], diffusion models fine-tuned on aesthetic datasets (e.g., FLUX.1-dev) typically achieve worse FID scores compared to models that have not undergone extensive aesthetic fine-tuning (e.g., SD1.5, SANA).



Figure 7: Qualitative comparison with SoTA methods for image generation on MJHQ30k [35].

Table 4: Inference clock time and image generation quality with different VLM decoding steps.

# Decoding Steps	64 (default)	32	16	8	4	2	1
Clock Time (seconds)	5.21s	2.63s	1.37s	0.66s	0.35s	0.19s	0.09s
FID ↓	18.64	<b>18.49</b>	18.79	18.89	19.97	24.90	60.35
sFID ↓	43.64	<b>43.58</b>	44.52	45.89	50.86	67.23	217.50
IS ↑	156.01	<b>158.93</b>	156.96	158.87	150.56	124.78	83.22

#### 5.4 MLLM Inference Time with Varying Decoding Steps

To further assess the effect of decoding steps, we conducted experiments with varying numbers of decoding passes on ImageNet. The results are summarized in Tab. 4. We show the inference clock time in seconds, as well as image quality metrics including FID, sFID, and IS. As shown, our method remains robust as long as the number of decoding steps is greater than 8. And we can observe that there is a clear trade-off between inference speed and image quality when using fewer decoding steps. Importantly, we also highlight that the MLLM decoding time is significantly smaller than the runtime of the diffusion-based image generation model. For instance, FLUX.1-dev with 12B parameters together with latent ControlNets takes 14.79 seconds to generate a single image with the default 28 denoising steps. Therefore, the MLLM decoding time is not a major bottleneck. From a practical standpoint, users can flexibly select the number of decoding steps based on whether they favor faster inference or higher image quality for their particular application.

## 6 Conclusion

We present BIFROST-1, a unified and efficient framework for both multimodal understanding and generation. BIFROST-1 bridges pretrained MLLMs with diffusion models by introducing patch-level CLIP image embeddings as latent variables, which are naturally aligned with the CLIP visual encoder of the MLLM. This design enables high-fidelity, controllable image generation while maintaining strong training efficiency. A key advantage of BIFROST-1 is that it fully preserves the original parameters of the MLLM backbone, ensuring that its multimodal planning and reasoning capabilities remain intact. Although our current model is trained with significantly fewer computational resources than SoTA methods and does not yet surpass them in performance, future work focusing on scaling to larger datasets and more powerful MLLM backbones can further realize its potential.

## **Acknowledgments**

This work was supported by DARPA ECOLE Program No. HR00112390060, NSF-AI Engage Institute DRL-2112635, ARO Award W911NF2110220, ONR Grant N00014-23-1-2356, and a Bloomberg Data Science PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

## References

- [1] I. AI, B. Gong, C. Zou, D. Zheng, H. Yu, J. Chen, J. Sun, J. Zhao, J. Zhou, K. Ji, L. Ru, L. Wang, Q. Guo, R. Liu, W. Chai, X. Xiao, and Z. Huang. Ming-lite-uni: Advancements in unified architecture for natural multimodal interaction, 2025.
- [2] M. AI. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, April 2025. Accessed: 2025-05-12.
- [3] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning, 2022.
- [4] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024.
- [5] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [7] S. Changpinyo, P. Sharma, N. Ding, and R. 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.
- [8] J. Chen, Z. Xu, X. Pan, Y. Hu, C. Qin, T. Goldstein, L. Huang, T. Zhou, S. Xie, S. Savarese, L. Xue, C. Xiong, and R. Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset, 2025.
- [9] M. Chen et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- [10] X. Chen, Z. Wu, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, and C. Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.
- [11] J. Cho, J. Lei, H. Tan, and M. Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, 2021.
- [12] J. Cho, J. Lu, D. Schwenk, H. Hajishirzi, and A. Kembhavi. X-lxmert: Paint, caption and answer questions with multi-modal transformers. In *EMNLP*, 2020.
- [13] J. Cho, A. Zala, and M. Bansal. Visual programming for step-by-step text-to-image generation and evaluation. *Advances in Neural Information Processing Systems*, 36:6048–6069, 2023.

[14] S. Datta, A. Ku, D. Ramachandran, and P. Anderson. Prompt expansion for adaptive text-to-image generation. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3449–3476, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics.

[15] G. DeepMind. Gemini 2.5 pro, March 2025. Large language model.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

[18] R. Dong, C. Han, Y. Peng, Z. Qi, Z. Ge, J. Yang, L. Zhao, J. Sun, H. Zhou, H. Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. In *The Twelfth International Conference on Learning Representations*, 2023.

[19] P. Esser, S. Kulal, A. Blattmann, R. Entezari, and J. Müller. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[20] W. Feng, W. Zhu, T.-j. Fu, V. Jampani, A. Akula, X. He, S. Basu, X. E. Wang, and W. Y. Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023.

[21] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[22] Y. Ge, Y. Ge, Z. Zeng, X. Wang, and Y. Shan. Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041*, 2023.

[23] Y. Ge, S. Zhao, J. Zhu, Y. Ge, K. Yi, L. Song, C. Li, X. Ding, and Y. Shan. Seed-x: Multi-modal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.

[24] M. Geyer, O. Bar-Tal, S. Bagon, and T. Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *The Twelfth International Conference on Learning Representations*, 2023.

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

[26] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[27] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[28] X. Hu, R. Wang, Y. Fang, B. Fu, P. Cheng, and G. Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[29] A. Hurst et al. Gpt-4o system card. <https://arxiv.org/abs/2410.21276>, 2024. arXiv:2410.21276 [cs.CL].

[30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[31] J. Y. Koh, D. Fried, and R. Salakhutdinov. Generating images with multimodal language models. *NeurIPS*, 2023.

[32] B. F. Labs. Flux.1-dev. <https://huggingface.co/black-forest-labs/FLUX.1-dev>, 2024.

[33] A. Lewkowycz et al. Minerva: Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022.

[34] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[35] D. Li, A. Kamko, E. Akhgari, A. Sabet, L. Xu, and S. Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024.

[36] T. Li, Y. Tian, H. Li, M. Deng, and K. He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[37] C. Liao, L. Liu, X. Wang, Z. Luo, X. Zhang, W. Zhao, J. Wu, L. Li, Z. Tian, and W. Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025.

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[39] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[40] Y. Liu et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[41] J. Lu, C. Clark, S. Lee, Z. Zhang, S. Khosla, R. Marten, D. Hoiem, and A. Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. *arXiv preprint arXiv:2312.17172*, 2023.

[42] J. Lu, C. Clark, R. Zellers, R. Mottaghi, and A. Kembhavi. UNIFIED-IO: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2023.

[43] Y. Ma, X. Liu, X. Chen, W. Liu, C. Wu, Z. Wu, Z. Pan, Z. Xie, H. Zhang, X. yu, L. Zhao, Y. Wang, J. Liu, and C. Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation, 2024.

[44] S. Mo, T. Nguyen, X. Huang, S. S. Iyer, Y. Li, Y. Liu, A. Tandon, E. Shechtman, K. K. Singh, Y. J. Lee, B. Zhou, and Y. Li. X-fusion: Introducing new modality to frozen large language models, 2025.

[45] C. Nash, D. Barrett, and D. Rezende. Generating images with sparse representations. In *International Conference on Machine Learning (ICML)*, pages 7937–7947. PMLR, 2021.

[46] E. Nijkamp et al. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.

[47] OpenAI. Improving image generation with better captions, 2023. Accessed: 2025-05-11.

[48] OpenAI. Introducing gpt-4o: Image, text, and audio generation, 2024. Accessed: 2025-05-11.

[49] OpenAI. Openai o1 team page, 2024. Accessed: 2025-05-11.

[50] OpenAI. Video generation models as world simulators, 2024. Accessed: 2025-05-11.

[51] X. Pan, S. N. Shukla, A. Singh, Z. Zhao, S. K. Mishra, J. Wang, Z. Xu, J. Chen, K. Li, F. Juefei-Xu, J. Hou, and S. Xie. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

[52] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[53] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[54] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[55] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[56] S. Ren, Q. Yu, J. He, X. Shen, A. Yuille, and L.-C. Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025.

[57] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[58] W. Shi et al. Prm800k: A prompt-rewarded dataset for math reasoning. *arXiv preprint arXiv:2305.17126*, 2023.

[59] W. Shi, X. Han, C. Zhou, W. Liang, X. V. Lin, L. Zettlemoyer, and L. Yu. Lmfusion: Adapting pretrained language models for multimodal generation, 2025.

[60] Q. Sun, Y. Cui, X. Zhang, F. Zhang, Q. Yu, Y. Wang, Y. Rao, J. Liu, T. Huang, and X. 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.

[61] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang. Emu: Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

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

[63] S. Tong, D. Fan, J. Zhu, Y. Xiong, X. Chen, K. Sinha, M. Rabbat, Y. LeCun, S. Xie, and Z. Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.

[64] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning, 2018.

[65] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[66] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, and T. Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.

[67] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *CoRR*, abs/2202.03052, 2022.

[68] X. Wang, X. Zhang, Z. Luo, Q. Sun, Y. Cui, J. Wang, F. Zhang, Y. Wang, Z. Li, Q. Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[69] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao. Simvlm: Simple visual language model pretraining with weak supervision, 2022.

[70] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[71] C. Wu, X. Chen, Z. Wu, Y. Ma, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, C. Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[72] Y. Wu, Z. Zhang, J. Chen, H. Tang, D. Li, Y. Fang, L. Zhu, E. Xie, H. Yin, L. Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[73] J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[74] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[75] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *Forty-first International Conference on Machine Learning*, 2023.

[76] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. 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.

[77] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[78] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[79] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[80] C. Zhou, L. Yu, A. Babu, K. Tirumala, M. Yasunaga, L. Shamis, J. Kahn, X. Ma, L. Zettlemoyer, and O. Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model, 2024.

[81] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, Z. Gao, E. Cui, X. Wang, Y. Cao, Y. Liu, X. Wei, H. Zhang, H. Wang, W. Xu, H. Li, J. Wang, N. Deng, S. Li, Y. He, T. Jiang, J. Luo, Y. Wang, C. He, B. Shi, X. Zhang, W. Shao, J. He, Y. Xiong, W. Qu, P. Sun, P. Jiao, H. Lv, L. Wu, K. Zhang, H. Deng, J. Ge, K. Chen, L. Wang, M. Dou, L. Lu, X. Zhu, T. Lu, D. Lin, Y. Qiao, J. Dai, and W. Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

## Appendix

### A Formal Description of the BIFROST-1 MLLM Architecture

As illustrated in Fig. 2, we initialize the visual generation branch from the pretrained MLLM by creating a trainable copy of the MLP and attention QKV projection layers [65]. The only component randomly initialized is the vision head, which is a simple linear projection layer. By reusing the majority of parameters from the pretrained MLLM, we avoid the costly process of realigning image embeddings.

Specifically, we use blue color to denote the original *frozen* MLLM-specific modules, which handle language modeling (**LM**) and image understanding (**Img-U**) tasks. These modules remain unchanged during training. In addition, we use yellow color to represent the newly introduced *trainable* modules for the image generation task (**Img-G**), which are initialized as trainable copies of the corresponding blue modules.

During the image generation training process, we first obtain input hidden states for each task. Text inputs  $x^{\text{Text}}$  are projected through a linear embedding layer to produce  $h_{\text{in}}^{\text{Text}}$ . Images used for image understanding ( $x^{\text{Img-U}}$ ) and image generation ( $x^{\text{Img-G}}$ ) are both passed through the frozen MLLM-embedded visual encoder  $\mathcal{E}^{\text{Und}}$ , producing hidden states  $h_{\text{in}}^{\text{Img-U}}$  and  $h_{\text{in}}^{\text{Img-G}}$  respectively:

$$h_{\text{in}}^{\text{Text}} = \text{Linear}_{\text{Text}}(x^{\text{Text}}) \quad h_{\text{in}}^{\text{Img-U}} = \mathcal{E}^{\text{Und}}(x^{\text{Img-U}}) \quad h_{\text{in}}^{\text{Img-G}} = \mathcal{E}^{\text{Und}}(x^{\text{Img-G}})$$

During the image generation inference process,  $h_{\text{in}}^{\text{Img-G}}$  is initialized from 2D learnable mask tokens.

By using patch-level CLIP image embeddings that are natively aligned with the MLLM’s visual encoder to represent visual signals in vision generation tasks, we eliminate the need for any additional alignment between the visual generation representation and the MLLM.

For attention processing, we use the frozen MLLM attention layers to compute Q, K, and V matrices for the text and image understanding tokens  $h_{\text{in}}^{\text{Text}}$  and  $h_{\text{in}}^{\text{Img-U}}$ , and use the newly added visual generation branch to process the image generation tokens  $h_{\text{in}}^{\text{Img-G}}$ :

$$h_Q^{\text{Text}}, h_K^{\text{Text}}, h_V^{\text{Text}} = \text{QKV}_{\text{MLLM}}(h_{\text{in}}^{\text{Text}}) \quad h_Q^{\text{Img-U}}, h_K^{\text{Img-U}}, h_V^{\text{Img-U}} = \text{QKV}_{\text{MLLM}}(h_{\text{in}}^{\text{Img-U}}) \\ h_Q^{\text{Img-G}}, h_K^{\text{Img-G}}, h_V^{\text{Img-G}} = \text{QKV}_{\text{Img-G}}(h_{\text{in}}^{\text{Img-G}})$$

For language modeling and image understanding tasks, we replicate the standard MLLM attention structure by attending over their respective modalities only. For image generation task, we enable cross-module attention, allowing image generation queries to attend jointly over all token types. Specifically:

$$h_O^{\text{Text}} = \text{O}_{\text{MLLM}}(\text{Attn}(h_Q^{\text{Text}}, [h_K^{\text{Text}} \circ h_V^{\text{Text}}], [h_V^{\text{Text}} \circ h_V^{\text{Text}}])) \\ h_O^{\text{Img-U}} = \text{O}_{\text{MLLM}}(\text{Attn}(h_Q^{\text{Img-U}}, [h_K^{\text{Img-U}} \circ h_K^{\text{Text}}], [h_V^{\text{Img-U}} \circ h_V^{\text{Text}}])) \\ h_O^{\text{Img-G}} = \text{O}_{\text{Img-G}}(\text{Attn}(h_Q^{\text{Img-G}}, [h_K^{\text{Img-G}} \circ h_K^{\text{Img-U}} \circ h_K^{\text{Text}}], [h_V^{\text{Img-G}} \circ h_V^{\text{Img-U}} \circ h_V^{\text{Text}}]))$$

where  $\circ$  denotes concatenation, and  $\text{O}(\cdot)$  denotes linear output projection layer. We apply a causal mask to text and image understanding tokens, and a bidirectional mask to image generation tokens.

For the MLP layers, we follow the same branching: text and image-understanding tokens  $h_{\text{in}}^{\text{Text}}$  and  $h_{\text{in}}^{\text{Img-U}}$  are passed through the frozen MLLM MLP, while image generation tokens  $h_{\text{in}}^{\text{Img-G}}$  use the trainable MLP from the generation branch:

$$h_{\text{MLP}}^{\text{Text}} = \text{MLP}_{\text{MLLM}}(h_O^{\text{Text}}) \quad h_{\text{MLP}}^{\text{Img-U}} = \text{MLP}_{\text{MLLM}}(h_O^{\text{Img-U}}) \quad h_{\text{MLP}}^{\text{Img-G}} = \text{MLP}_{\text{Img-G}}(h_O^{\text{Img-G}})$$

Finally, task-specific heads convert the hidden states to output predictions. For language modeling and image understanding, we apply a linear projection to  $h_{\text{MLP}}^{\text{Text}}$ , and for image generation, we project  $h_{\text{MLP}}^{\text{Img-G}}$  using the vision generation head:

$$h_{\text{out}}^{\text{Text}} = \text{TextHead}(h_{\text{MLP}}^{\text{Text}}) \quad h_{\text{out}}^{\text{Img-G}} = \text{VisionHead}(h_{\text{MLP}}^{\text{Img-G}})$$

## B Broader Impacts

**BIFROST-1** is motivated by the fact that training a unified multimodal generation and understanding model that can perform native generation with high visual quality usually requires huge computational cost. By bridging pretrained MLLM with pretrained diffusion models, training **BIFROST-1** can be significantly faster. Therefore, we believe that our work can be a strong contribution to efficient unified model training. While our framework can benefit numerous applications in image generation, similar to other image generation frameworks, it can also be used for potentially harmful purposes (e.g., creating false information or misleading images). Therefore, it should be used with caution in real-world applications.

## C Safeguards

**BIFROST-1** is built upon pretrained MLLM (*i.e.*, Qwen2.5-VL) and diffusion models (*i.e.*, FLUX.1-dev) with strong safeguards, and trained on publically available image datasets (*i.e.*, MSCOCO and SA1B) that removes unsafe concepts. Therefore, our model avoids the high risk for misuse.

## D Limitations

Note that **BIFROST-1** is designed as a bridging method that connects existing MLLMs with diffusion-based image generation models. As such, its performance, output quality, and potential visual artifacts are inherently influenced by the capabilities and limitations of the underlying backbone models it relies on. For instance, if the diffusion model used as the visual backbone struggles with generating complex, rare, or previously unseen scenes and objects, then **BIFROST-1**, which builds upon this foundation, may also exhibit suboptimal image generation results. This dependency highlights the importance of selecting strong and well-generalized base models when applying **BIFROST-1** to real-world or open-domain generation tasks.

## E License

We use standard licenses from the community and provide the following links to the licenses for the datasets, codes, and models that we used in this paper. For further information, please refer to the specific link.

PyTorch [4]: [BSD-style](#)

HuggingFace Transformers [70]: [Apache License 2.0](#)

HuggingFace Diffusers [66]: [Apache License 2.0](#)

FLUX.1-dev [32]: [Non-Commercial License](#)

Qwen2.5-VL [5]: [Non-Commercial License](#)

MSCOCO dataset [38]: [CC BY 4.0](#)

CC12M dataset [7]: [Permissive Custom License](#)

SA1B dataset [30]: [SA-1B Dataset Research License](#)

MJHQ30k dataset [35]: [Playground v2 Community License](#)



Figure 8: Visualization examples from MJHQ30k dataset.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer **[Yes]** , **[No]** , or **[NA]** .
- **[NA]** means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "**[Yes]**" is generally preferable to "**[No]**", it is perfectly acceptable to answer "**[No]**" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "**[No]**" or "**[NA]**" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer **[Yes]** to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: **[Yes]**

Justification: The abstract and introduction clearly state the claims we made.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: We include limitation sections in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our model architecture is well-illustrated in Sec. 3, and experiment setup is shown in Sec. 4 in detail.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: **[Yes]**

Justification: We use publically available open-sourced datasets and benchmarks for model training and evaluation. We include our code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: **[Yes]**

Justification: We show all training and test details in Sec. 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: **[No]**

Justification: Due to the large training cost for each run, we follow previous works [71, 43, 10, 51, 63, 80, 59] and only train the model once and inference once on each evaluation task.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: **[Yes]**

Justification: We describe training compute details in Sec. 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: We conform to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: We discussed broader impacts in the appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: Our model aims at bridging pretrained MLLM and diffusion models which already have safeguards. We include safeguards information in the appendix.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We include licenses in the appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](#) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We include instructions about code and how to use the model in the README.md file in the supplementary material.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

#### **15. Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### **16. Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: When preparing this manuscript, we used LLM only for editing (grammar checking) purposes.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.