LMFusion: Adapting Pretrained Language Models for Multimodal Generation

Weijia Shi [*w] **Xiaochuang Han** [*w] **Chunting Zhou** [f] **Weixin Liang** [s] **Xi Victoria Lin** [f]
Luke Zettlemoyer [wf] **Lili Yu** [f]

[w]University of Washington [f]FAIR at Meta [s]Stanford University

swj0419@uw.edu xhan77@uw.edu liliyu@meta.com

Abstract

We present LMFusion, a framework for empowering pretrained text-only large language models (LLMs) with multimodal generative capabilities, enabling them to understand and generate both text and images in arbitrary sequences. LMFusion leverages existing Llama-3's weights for processing texts autoregressively while introducing additional and parallel transformer modules for processing images with diffusion. During training, the data from each modality is routed to its dedicated modules: modality-specific feedforward layers, query-key-value projections, and normalization layers process each modality independently, while the shared self-attention layers allow interactions across text and image features. By freezing the text-specific modules and only training the image-specific modules, LMFusion preserves the language capabilities of text-only LLMs while developing strong visual understanding and generation abilities. Compared to methods that pretrain multimodal generative models from scratch, our experiments demonstrate that LMFusion improves image understanding by 20% and image generation by 3.6% while maintaining Llama-3's language capabilities. We also show that this framework can adapt existing vision-language models with multimodal generation ability.

1 Introduction

Over the past few years, we have seen significant progress in multimodal generative models capable of understanding and generating interleaved text and images in arbitrary sequences [1, 2, 3]. Models like Transfusion [4], Chameleon [5], and Unified-IO [6, 7] demonstrate the potential of unified architectures that seamlessly handle both image and text modalities. However, these models typically train from scratch, demanding significant computational resources to achieve proficiency across all modalities. The computational cost of mastering even a single modality is substantial—training a state-of-the-art text-only large language models (LLMs) like Llama-3 [8] requires training over 15 trillion tokens.

Given these computational demands, we investigate an alternative paradigm that reuses and adapts existing pretrained LLMs [9, 10, 11]. We address a fundamental research question: *How to preserve the text-only performance of pretrained LLMs while equipping them with visual understanding and generation abilities?* Our experiments show that naive finetuning of text-only LLMs on multimodal data leads to significant degradation of their language processing capabilities.

To address this challenge, we introduce **LMFusion**, a framework that enhances a pretrained text-only LLM, Llama-3 [8] with multimodal capabilities by building upon the recipe of Transfusion [4]. Drawing from recent and parallel work on modality separation [12, 13, 14, 15], LMFusion integrates the original Llama modules pretrained for language processing while introducing additional dedicated transformer modules for visual understanding and generation tasks. As shown in Figure 1, we

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

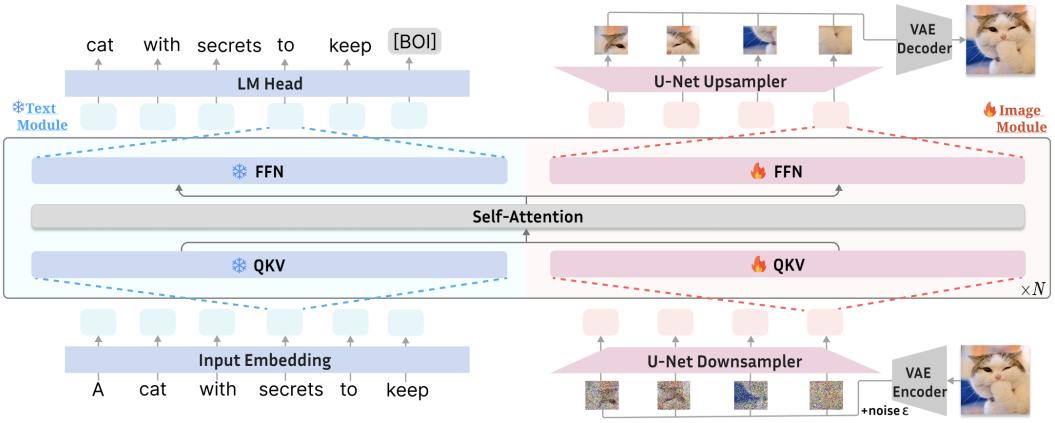

Figure 1: **Overview of LMFusion**. It uses modality-specific FFNs and QKV projections to process text and image data separately: the text "A cat with secrets to keep" goes to the text module, while the image patches of the cat goes to the image module. In the self-attention layer, text and image representations can attend to all previous contexts across the modality boundaries. Both modules are initialized from Llama-3, with the text module frozen to preserve language capabilities while the image module trained on image data. Layer normalization and residual connections are folded into the QKV and FFN modules. A special `BOI` token separates different modalities.

employ modality-specific query-key-value (QKV) projections and feed-forward networks (FFNs) to process text and image data separately while still allowing for cross-modal interactions in the joint self-attention layer. By freezing the text modules while finetuning the image modules, we preserve the language-only capabilities of pretrained LLMs while giving a head start to the learning of visual understanding and generation. Compared to pretraining multimodal generative models from scratch, this approach avoids the need to include text-only data in the training process, significantly reducing the computational demands.

To evaluate the effectiveness of our approach, we conduct comprehensive experiments comparing LMFusion with Transfusion in controlled settings. Specifically, we initialize our LMFusion architecture with a pretrained Llama-3 8B model [8] and continue training on the same image data as in Transfusion [4]. Compared to Transfusion, LMFusion achieves a 20% improvement in image understanding and 3.6% improvement in image generation. It also preserves Llama-3's text-only performance that outperforms Transfusion by 11.6%. Figure 2 presents images generated by LMFusion. Additionally, we further demonstrate that this framework can adapt existing vision-language models (e.g., LLaVA) with multimodal generation ability.

Through ablation studies, we analyze the key architectural decision for LMFusion: separating both self-attention and FFNs for different modality data while freezing weights for the pretrained language modality. We show that naive finetuning of the dense pretrained LLMs on multimodal data (*no separation*) leads to a catastrophic forgetting of their original language capabilities. Furthermore, deep separation proves to be more effective than shallow separation (*using modality-specific FFNs only*), with both approaches outperforming models with no separation.

2 Background: Transfusion

Transfusion [4] is a single unified multimodal model that is capable of text generation, image understanding, and image generation tasks, by jointly predicting next tokens in language and diffusing image representations. Given a multimodal input $(\boldsymbol{x}^{txt}, \boldsymbol{x}^{img})$, Transfusion jointly learns to do *language modeling* (§2.1) on $\boldsymbol{x}^{txt}$ and *image diffusion* (§2.2) on $\boldsymbol{x}^{img}$. Its architecture is same as a standard Transformer [16] with an additional U-Net [17] that projects image representations down and up before and after diffusion.

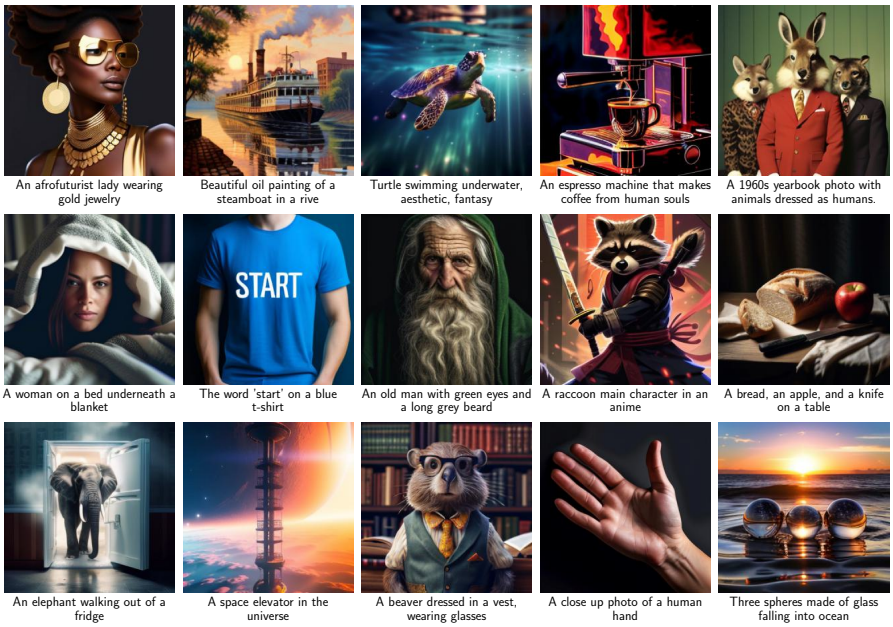

Figure 2: Generated images from LMFusion fine-tuned on aesthetically appealing images for improved quality.

2.1 Language Modeling

Given a sequence of discrete language tokens $\boldsymbol{x}^{txt} = x_1^{txt}, \ldots, x_N^{txt}$, a language model θ represents its joint probability by $P(\boldsymbol{x}^{txt}) = \prod_{i=1}^{N} P_\theta(x_i^{txt} \mid \boldsymbol{x}_{<i}^{txt})$. This formulation sets up an autoregressive task, where each token x_i^{txt} is predicted based on its preceding tokens $\boldsymbol{x}_{<i}^{txt}$. The language model is learned by minimizing the cross-entropy between P_θ and the observed data distribution, which is commonly referred to as the LM loss:

$$\mathcal{L}_{\text{LM}} = \mathbb{E}_{x_i^{txt}}[-\log P_\theta(x_i^{txt} \mid \boldsymbol{x}_{<i}^{txt}, \boldsymbol{x}^{img})] \tag{1}$$

Optionally, if there exists image data preceding the language tokens (e.g., image-caption data), Transfusion adds the representation of $\boldsymbol{x}^{img}$ as additional condition to the objective. More details of representing $\boldsymbol{x}^{img}$ are presented below.

2.2 Image Diffusion

Given a raw image, Transfusion first encodes the image into a sequence of continuous latent representation $\boldsymbol{x}^{img}$ with a pretrained and frozen VAE tokenizer [18]. It then employs Denoising Diffusion Probabilistic Models (i.e., DDPM) to learn to reverse a gradual noise-addition process added in the forward process [19]. In the forward diffusion process, a Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is added to the image representation $\boldsymbol{x}^{img}$ over T steps, creating a sequence of noisy image representations $\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T$. Specifically, at each step t, the noisy image representation is given by:

$$\boldsymbol{x}_t^{img} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}^{img} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon} \tag{2}$$

Here $\bar{\alpha}_t$ follows a common cosine schedule [20]. In the reverse process, the diffusion model $\boldsymbol{\epsilon}_\theta(\cdot)$ with parameters θ learns to predict the added noise $\boldsymbol{\epsilon}$ given the noisy data $\boldsymbol{x}_t^{img}$ at timestep t and a context $\boldsymbol{x}^{txt}$ that can include text prompts such as captions to the image diffusion: [1]

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\boldsymbol{x}^{img}, t, \boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t^{img}, t, \boldsymbol{x}^{txt})\|_2^2] \tag{3}$$

The Transfusion architecture contains U-Net downsampler and upsampler to reduce the dimension of $\boldsymbol{x}^{img}$. The U-Net downsampler transforms the image into fewer patches before the main Transformer modules while the upsampler projects them back to the original dimension of $\boldsymbol{x}^{img}$ after the Transformer.

[1] Similar to $\boldsymbol{x}^{txt}$, this context can also include image representations $\boldsymbol{x}^{img}$ under an image editing setup. We omit it in the notation for simplicity.

2.3 Training Objective

During training, Transfusion is optimized to predict both the LM loss on the text input $\boldsymbol{x}^{txt}$ and the diffusion loss on the image input $\boldsymbol{x}^{img}$. These two losses are combined using a hyperparameter λ:

$$\mathcal{L}_{\text{Transfusion}} = \mathcal{L}_{\text{LM}} + \lambda \cdot \mathcal{L}_{\text{DDPM}} \tag{4}$$

3 LMFusion

One notable feature of Transfusion is that it has the same architecture as mainstream LLMs (e.g., Llama [21]) while being capable of text generation, image understanding, and image generation together, through an end-to-end training (Equation 4). [4] trains Transfusion from scratch using language-only and image-caption data. However, such training from scratch requires substantial computational resources, and its performance on language-only tasks still lags behind the pretrained, text-only LLMs. In this work, we aim to effectively adapt pretrained, text-only LLMs to handle image understanding and generation tasks. Specifically, we build on an open-weight LLM, Llama-3 [8], and continue training it with the Transfusion objectives to handle both modalities. Since Transfusion uses shared parameters for its language modeling and image diffusion objectives, the key challenge is to prevent Llama-3's strong text-only performance from dropping while optimizing for its new image capabilities.

3.1 Model Architecture

In response to the challenge above, we propose LMFusion, a framework that combines a pretrained, text-only Llama model with a dedicated image transformer for visual generation and understanding, enabling each modality to be processed through independent weights. By freezing the text modules while finetuning the visual modules, we preserve its language-only capabilities while giving the learning of visual understanding and generation a boost start.

LMFusion is a decoder-only model consisting of N transformer layers. As shown in Figure 1, central to the design are the modality-specific attention layer and Feed-Forward Network (FFN), each handling only data from its corresponding modality. Without loss of generality, we describe LMFusion below in a configuration with a single transformer layer, folding residual connections and layer normalization directly into the self-attention and FFN. The inputs to the model are text tokens $\boldsymbol{x}^{txt}$ and noisy image representations $\boldsymbol{x}_t^{img} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}^{img} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. We use blue for text-specific modules and red for image-specific modules.

Input projection The input text tokens $\boldsymbol{x}^{txt}$ are projected by a linear embedding layer to a sequence of text hidden states $\boldsymbol{h}_{\text{in}}^{txt}$. The noisy image $\boldsymbol{x}_t^{img}$ are projected to a sequence of image representations $\boldsymbol{h}_{\text{in}}^{img}$ via a U-Net downsampler.

$$\boldsymbol{h}_{\text{in}}^{txt} = \text{Proj}_{\text{text}}(\boldsymbol{x}^{txt}) \tag{5}$$

$$\boldsymbol{h}_{\text{in}}^{img} = \text{UNet-Down}_{\text{img}}(\boldsymbol{x}_t^{\text{img}}, t) \tag{6}$$

Then the text hidden states $\boldsymbol{h}_{\text{in}}^{txt}$ or image hidden states $\boldsymbol{h}_{\text{in}}^{img}$ are fed into the following attention layer.

Modality-specific self-attention We create separate attention matrices for each modality. Specifically, the text hidden states $\boldsymbol{h}_{\text{in}}^{txt}$ and image hidden states $\boldsymbol{h}_{\text{in}}^{img}$ are converted into their respective queries, keys, and values via separate Q, K, V matrices. The pre-attention layer normalization is also modality-specific and is folded into the QKV functions.

$$\boldsymbol{h}_{\text{Q}}^{txt}, \boldsymbol{h}_{\text{K}}^{txt}, \boldsymbol{h}_{\text{V}}^{txt} = \text{QKV}_{\text{text}}(\boldsymbol{h}_{\text{in}}^{txt}) \tag{7}$$

$$\boldsymbol{h}_{\text{Q}}^{img}, \boldsymbol{h}_{\text{K}}^{img}, \boldsymbol{h}_{\text{V}}^{img} = \text{QKV}_{\text{img}}(\boldsymbol{h}_{\text{in}}^{img}) \tag{8}$$

We enable cross-modal attention by concatenating the queries, keys, and values from both image and text modalities into unified sequences. The attention-weighted values at text and image tokens are then projected back into the hidden state dimension using separate O weights for each modality.

$$\boldsymbol{h}_{\text{O}}^{txt} = \text{O}_{\text{text}}(\text{softmax}(\frac{\boldsymbol{h}_{\text{Q}}^{txt}[\boldsymbol{h}_{\text{K}}^{img} \circ \boldsymbol{h}_{\text{K}}^{txt}]^T + M}{\sqrt{d}})[\boldsymbol{h}_{\text{V}}^{img} \circ \boldsymbol{h}_{\text{V}}^{txt}]) \tag{9}$$

$$\boldsymbol{h}_{\text{O}}^{img} = \text{O}_{\text{img}}(\text{softmax}(\frac{\boldsymbol{h}_{\text{Q}}^{img}[\boldsymbol{h}_{\text{K}}^{txt} \circ \boldsymbol{h}_{\text{K}}^{img}]^T + M}{\sqrt{d}})[\boldsymbol{h}_{\text{V}}^{txt} \circ \boldsymbol{h}_{\text{V}}^{img}]) \tag{10}$$

where $\circ$ denotes concatenation. M represents a hybrid attention mask same as in Transfusion [4] with a causal mask applied to text tokens and a bi-directional mask applied to image tokens. This design allows for self-attention within and across modalities, encouraging cross-modality integrations.

Modality-specific feed-forward network After the attention layer, we employ modality-specific FFNs to process text and image data separately. The pre-FFN layer normalization is also modality-specific and is folded in the FFN functions.

$$\boldsymbol{h}^{txt}_{\text{FFN}} = \text{FFN}_{\text{text}}(\boldsymbol{h}^{txt}_{\text{O}}) \tag{11}$$

$$\boldsymbol{h}^{img}_{\text{FFN}} = \text{FFN}_{\text{img}}(\boldsymbol{h}^{img}_{\text{O}}) \tag{12}$$

Output projection Finally, after N layers of self-attention and FFNs, the resulting hidden states are projected either to logits in text via language model's output layer, or to predicted noise in image via a U-Net upsampler.

$$\boldsymbol{p}_{\text{logits}} = \text{LM-Head}_{\text{text}}(\boldsymbol{h}^{txt}_{\text{FFN}}) \tag{13}$$

$$\boldsymbol{\epsilon}_{\text{pred}} = \text{UNet-Up}_{\text{img}}(\boldsymbol{h}^{img}_{\text{FFN}}, t, \boldsymbol{h}^{img}_{\text{in}}) \tag{14}$$

Same as Transfusion, the output $\boldsymbol{p}_{\text{logits}}$ and $\boldsymbol{\epsilon}_{\text{pred}}$ are passed through the language modeling loss (Equation 1) and DDPM loss (Equation 3) respectively. All parameters in the text modules along with self-attention and FFN parameters in the image modules are initialized from the pretrained Llama model. During optimization, we ***decouple the learning rates*** for the text and image parameter groups: a text learning rate, η_{text}, is used for $\{\text{Proj}_{\text{text}}, \text{QKV}_{\text{text}}, \text{O}_{\text{text}}, \text{FFN}_{\text{text}}, \text{LM-Head}_{\text{text}}\}$, and an image learning rate, η_{img}, for $\{\text{UNet-Down}_{\text{img}}, \text{QKV}_{\text{img}}, \text{O}_{\text{img}}, \text{FFN}_{\text{img}}, \text{UNet-Up}_{\text{img}}\}$. To preserve the model's performance on text-only benchmarks, we use $\eta_{\text{text}} = 0$ (freezing text modules) for our main experiments and explore different configurations in §5.

4 Experiments

4.1 Training Setup

Data Following Transfusion [4], we use the same collection of 380M Shutterstock image-caption data, where each image is center-cropped and resized to 256×256 pixels. We order the captions before images (i.e., emphasizing image generation conditioned on texts) 80% of the time, and order the images before captions for the rest.

Model Details For image tokenization, we use the same VAE encoder[2] as Transfusion to compress an image of 256×256 pixels into a $32 \times 32 \times 8$ tensor. These tensors are then passed into a 2-block U-Net downsampler [17] to further reduce dimensions, resulting in a sequence of 256 patches (tokens). Both text-specific and image-specific Transformer modules are initialized from the pretrained Llama-3 8B model [8]. The U-Net downsampler and a corresponding U-Net upsampler, totaling 0.27 billion parameters [4], are trained from scratch. Like Transfusion, LMFusion uses a maximum context length of 4096 tokens.

Optimization In our main experiments, to preserve the language-only performance, we freeze the text modules ($\eta_{\text{text}} = 0$) while training only the image modules using an AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1 \times 10^{-8}$) with a learning rate $\eta_{\text{image}} = 1 \times 10^{-4}$. The learning rate follows a cosine decay schedule with a 4000-step warmup period before gradually decreasing to 1.5×10^{-5}. The model is trained using 128 H100 GPUs over 4 days.

4.2 Evaluation Setup

We compare our model with both the original Transfusion 7B model trained from scratch [4] and the Transfusion model initialized from the same LLaMA model. [3] The original Transfusion was trained for 250K steps on 0.25T language-only tokens (text data) and 0.25T image-captions tokens (image data). Since we freeze and reuse the text module from existing text-only models during training, we can exclude text data from our training process while maintaining language capabilities. In the first

[2] `https://huggingface.co/stabilityai/sd-vae-ft-mse`

[3] Transfusion 7B and Llama-3 8B have the same Transformer sizes. The size difference is due to the different vocabularies, which affects input and output embedding layers.

Model	Language-only Evaluation			Image Understanding	Image Generation (without | with CFG)	
	HellaSwag↑	SIQA↑	WinoGrande↑	CIDEr ↑	FID ↓	CLIP ↑
Llama 3	60.0	48.1	72.8	–	–	–
Transfusion	51.0	42.3	64.3	32.0	14.4 | 8.70	22.1 | 24.4
LMFusion (0.5x FLOPs)	60.0	48.1	72.8	38.3	13.9 | 8.75	22.0 | 24.4
LMFusion (1.0x FLOPs)	60.0	48.1	72.8	38.4	14.0 | 8.61	22.1 | 24.4

Table 1: **Results across text-only benchmarks, image understanding and image generation.** LMFusion preserves Llama-3's text performance while adding strong image understanding and generation capabilities. Image generation results are obtained without classifier-free guidance or with a CFG factor of 1.55. **Additionally, we show detailed analyses of our full LMFusion recipe vs. the vanilla Transfusion recipe initialized from Llama-3 in Figure 4 and Figure 5.**

configuration, we use the amount of 0.25T image data in Transfusion while leaving out the text data (approximately half the total FLOPs of Transfusion), while in the second configuration, we match Transfusion's total FLOPs. Following Transfusion, we evaluate LMFusion on language-only, image understanding, and image generation tasks.

Language-only: We evaluate the model's language abilities using four tasks from the standard Llama evaluation suite [8], including Hellaswag [22], PIQA [23], SIQA [24], and WinoGrande [25]. We report accuracy on these benchmarks. **Image Generation**: For evaluating image generation, we use the MS-COCO benchmark [26]. We generate images for 30K randomly selected prompts from the validation set and measure the Frechet Inception Distance (FID) [27] and CLIP scores [28]. Our image generation results include versions obtained without classifier-free guidance and with a CFG coefficient of 1.55 or 1.6. **Image Understanding**: We evaluate the models' ability to generate image descriptions using the test split of MS-COCO [26], reporting CIDEr scores [29].

4.3 Results

Table 1 compares two variants of LMFusion against Transfusion. On language-only benchmarks, LMFusion keeps the strong performance of Llama-3 since we freeze all text modules. For image understanding, LMFusion substantially surpasses Transfusion, with a 20% improvement. In image generation tasks, LMFusion also shows superior results in both FID and CLIP scores. Furthermore, in §5, we show from Figure 4 that LMFusion outperforms Transfusion initialized from Llama-3 (i.e., dense model with no separation) during the training process. In Figure 3, we benchmark the performance of LMFusion and Transfusion throughout the training.[4] We observe a consistent advantage of LMFusion over Transfusion during the entire training, for image captioning and generation. These results suggest that LMFusion effectively leverages the pretrained language modules from Llama while developing strong image abilities. Although LMFusion has twice as many parameters as Transfusion, it uses same training FLOPs since only half of the parameters are activated for each input token from an arbitrary modality.

5 Analysis

Central to LMFusion is our modality separation techniques, which employs the design of modality-specific modules and decoupled learning rates for language and image modules. Our architectural ablation (§5.1) demonstrates the importance of the design for maintaining model performance across both modalities. We further showcase that this recipe could be used for adapting pretrained vision-language models.

5.1 Architecture Ablations

5.1.1 Experimental Design

To evaluate different design choices, we conduct ablation studies using small-scale variants of LMFusion. Our analysis focuses on the impact of modality separation by comparing three designs:

[4]For the image generation results plotted throughout the training, we use a smaller subset of 5K prompts and without classifier-free guidance.

Model	Base LLM	Image Und.				Image Gen.
		MMMU ↑	ChartQA ↑	RealWorldQA ↑	MME-P ↑	FID ↓
EMU-3	–	31.6	51.8	57.4	–	12.8
Show-O	Phil-1.5 1.3B	27.4	–	–	1435.7	9.2
Janus	DeepSeek 1.3B	30.5	–	–	1338.0	8.5
Chameleon	–	28.4	0.0	19.6	–	26.7
MetaMorph	LLaMA-3.1 8B	**41.8**	37.1	58.3	–	11.8
Transfusion	–	–	–	–	–	8.7
LLaVAFusion	LLaVA-Next 8B	41.7	**69.5**	**60.0**	**1603.7**	**8.2**

Table 2: **Comparison of multimodal models across image understanding and generation capabilities.** Models are evaluated on various image understanding benchmarks and image generation quality (FID). The models without base LLM are pretrained from scratch.

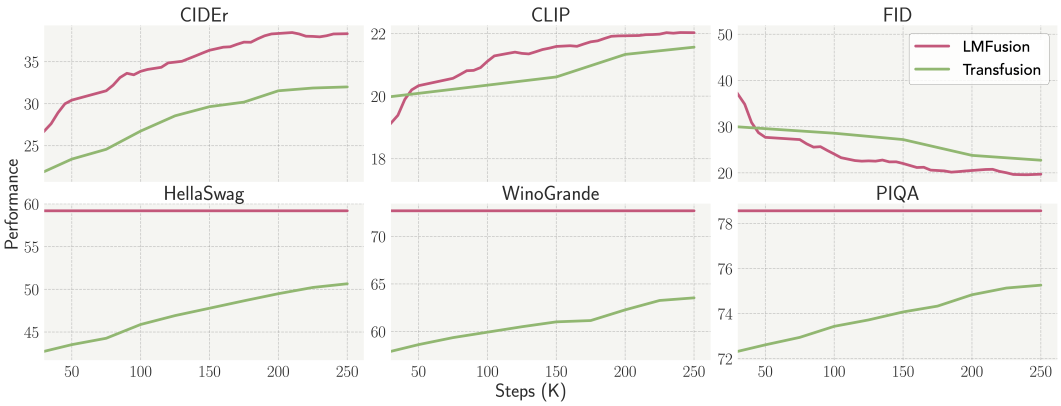

Figure 3: **Evaluation of LMFusion and Transfusion during training.** LMFusion keeps the text performance of Llama throughout training, while achieving better image understanding ability (CIDEr) and image generation quality (CLIP, FID).

(1) *no separation* (a single dense model), (2) *shallow separation* (using modality-specific FFNs only), and (3) *deep separation* (using both modality-specific FFNs and attention mechanisms, our final LMFusion).

No separation (dense model) We begin our experiments with the dense Llama-3 8B model trained using the Transfusion recipe. This dense model maintains a unified structure where most components are shared across modalities (a single set of QKV, O and FFN process both texts and images), with the exception of U-Net upsampler and downsampler. For training, we use a text learning rate (η_{text}) for the components initialized from the text-only LLM $\{\text{Proj}_{\text{text}}, \text{QKV}, \text{O}, \text{FFN}, \text{LM-Head}_{\text{text}}\}$, and an image learning rate η_{img} for $\{\text{UNet-Down}_{\text{img}}, \text{UNet-Up}_{\text{img}}\}$. To investigate the impact of learning rate decoupling, we experiment with various learning rate ratios $\frac{\eta_{\text{text}}}{\eta_{\text{image}}} \in \{0, 0.1, 1\}$, with a constant image learning rate $\eta_{\text{image}} = 1 \times 10^{-4}$, the same as the main experiments. A ratio of 1 represents standard continual pretraining where all components share the same learning rate, while a ratio of 0 indicates a complete freezing of text-related components.

Shallow separation (modality-specific FFNs only) We explore a simplified variant of LMFusion that separates only FFNs into text-specific and image-specific modules—a common approach in mixture-of-experts architectures [3, 30]. In this setup, we use a single shared attention mechanism (QKV , O) for processing both image and text data. For training, we employ separate learning rates: η_{text} for text-related components $\{\text{Proj}_{\text{text}}, \text{QKV}, \text{O}, \text{FFN}_{\text{text}}, \text{LM-Head}_{\text{text}}\}$ and η_{img} for image-related components $\{\text{Unet-Down}_{\text{img}}, \text{FFN}_{\text{img}}, \text{Unet-Up}_{\text{img}}\}$. We experiment with various learning rate ratios $\frac{\eta_{\text{text}}}{\eta_{\text{image}}} \in \{0, 0.1, 1\}$.

Deep separation (modality-specific FFNs and attention) Our LMFusion, as described in section 3, represents a deep separation design where both FFNs and attention mechanisms are modality-specific.

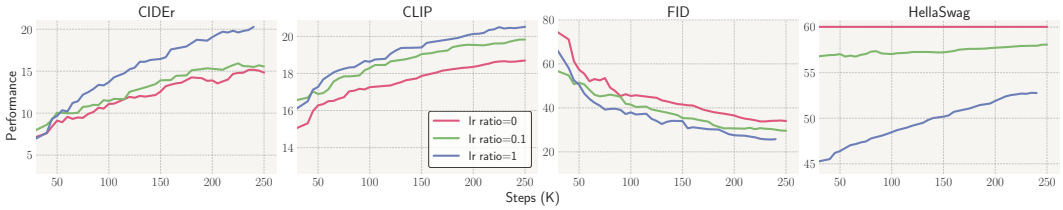

Figure 4: **Performance of naive Llama-3 finetuning (no separation) with varying lr ratio $\frac{\eta_{\text{text}}}{\eta_{\text{image}}}$.** When directly finetuning the Llama-3 model for multimodal generation, using the same learning rate for both text and image components (lr ratio $= 1$) substantially reduces its text-only performance. Lowering the learning rate for the text component relative to the image component (lr ratio < 1) helps preserve language performance but slows down the acquisition of multimodal abilities.

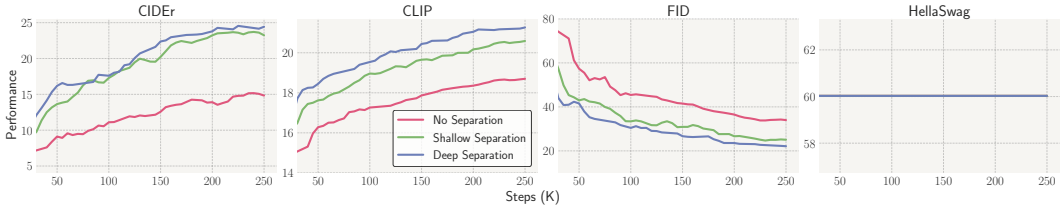

Figure 5: **Performance of *no separation* (dense model), *shallow separation* (modality-specific FFNs only), and *deep separation* (modality-specific FFNs and attention) when text modules are frozen.** Deep modality separation outperforms shallow separation and no separation.

While our primary configuration freezes text modules during training, we also analyze the impact of different learning dynamics by varying the learning rate ratio $\frac{\eta_{\text{text}}}{\eta_{\text{image}}}$ across $\{0, 0.1, 1\}$.

In the ablation study, all models are trained for 250K training steps with a sequence length of 4,096 tokens and a batch size of 250K tokens. The training data comprised 0.03T text-only tokens and 0.03T image-caption tokens. All other hyperparameters remained consistent with those employed in our main experiments.

5.1.2 Results

Naive finetuning of dense pretrained LLMs for multimodal generation compromises their original language capabilities. When directly finetuning Llama-8B (no separation) using the Transfusion recipe, we observe significant performance trade-offs between image and text capabilities (Figure 4). With equal learning rates for text and image components ($\frac{\eta_{\text{text}}}{\eta_{\text{image}}} = 1$), the model shows continuous improvement in image understanding and generation. However, this comes at a substantial cost to language capabilities, with performance on HellaSwag dropping by 15% initially. While language performance improves during training, it never recovers to the original Llama-3 model's level, maintaining a persistent 7% gap.

To mitigate this issue, we explore setting $\frac{\eta_{\text{text}}}{\eta_{\text{image}}} < 1$, which allows us to train image-specific modules (U-Nets) with a regular learning rate while preserving text abilities using a smaller learning rate for the general Transformer components. Figure 4 shows this improves language-only benchmark performance, reducing the gap from 7% to 2% when the ratio is 0.1. However, for dense models, this improvement comes at the cost of consistently reduced image capabilities. Overall, while learning rate decoupling offers some mitigation to the text performance drop, training dense pretrained LLMs without modality separation remains suboptimal.

Deep Modality Separation Outperforms Shallow Separation. In Figure 5, we compare three architectures: no separation (dense), shallow separation (modality-specific FFNs only), and deep separation (modality-specific FFNs and attention). We set $\frac{\eta_{\text{text}}}{\eta_{\text{image}}} = 0$ (freezing the text module) across all models to maintain Llama-3's text performance. Both separation approaches significantly outperform the dense model on all image benchmarks. While shallow separation performs slightly worse on image understanding, the performance gap widens notably in image generation tasks.

Additionally, deep separation with $\frac{\eta_{\text{text}}}{\eta_{\text{image}}} = 0$ has the same amount of *tunable* parameters as no separation with $\frac{\eta_{\text{text}}}{\eta_{\text{image}}} = 1$. Despite the intrinsic advantage of modality separation for text-only tasks,

for image understanding and generation, we still observe that deep separation (blue curve in Figure 5) are better than no separation (blue curve in Figure 4). These results show that modality separation is crucial for adapting pretrained language-only LLMs for multimodal generation.

5.2 LLaVAFusion: extending LMFusion to vision-language models

LMFusion continues training the language-only pretrained LLM Llama with the Transfusion recipe. Can this recipe be extended to on vision-language models (VLMs) such as LLaVA [31, 32] and Qwen-VL [33] as well? In this section, we extend the recipe of LMFusion to VLMs, preserving their multimodal understanding capabilities while introducing image generation abilities. Specifically, we build on LLaVA-NeXT [32], freezing its transformer parameters and integrating a dedicated, image-specific transformer module trained in parallel. We use the same data and model settings as LMFusion. We refer to this new model as LLaVAFusion and demonstrate its image understanding performance on MMMU [34], MME-Perception [35], ChartQA [36], and RealWorldQA [37], as well as its image generation results. For baselines, we compare LLaVAFusion against EMU-3 [38], Show-O [39], Janus [40], Chameleon [41], MetaMorph [42], and Transfusion [4]. As shown in Table 2, LLaVAFusion LLaVAFusion demonstrates strong performance in both image understanding and generation when compared to other unified multimodal LMs. This demonstrates that LMFusion is promising as an extension not only to language-only LLMs but also to VLMs, enhancing the multimodal generation capabilities in both cases.

6 Related Work

Unified Models for Multimodal Generation: Recent work has extensively explored unified frameworks for multimodal generation, including text generation, image understanding, and image generation. While texts are commonly represented as discrete tokens across models, approaches to representing images—especially for image generation—vary significantly. For instance, methods in [6, 43, 7, 5, 44, 11], represents images using vector-quantized discrete tokens [45, 46, 47]. An alternative method, adopted by [48, 49], employs continuous embeddings that require a separate diffusion model for decoding. In this work, we build upon Transfusion [4], which integrates autoregressive generation for texts with diffusion for images within a single model. **Model Sparsity**: Model sparsity through Mixture of Experts (MoE) [50, 30, 51, 52] has proven highly effective in improving LLM training efficiency. This approach has recently been extended to multimodal models [12, 53, 54, 55], particularly to address potential conflicts between different modalities. For example, recent efforts [13, 3, 56, 57] replace standard Transformer FFNs with modality-specific experts, enabling separate processing paths for different modalities. Our work takes this concept further by using modality-specific attention mechanisms. Concurrent work [15, 14] demonstrates the effectiveness of this deeper separation in multimodal pretraining and image generation. **Reuse of LLMs in Multimodal Training**: Based on the strong language capabilities of LLMs, some recent models on multimodal generation initializes their models from pretrained, language-only LLMs. For example, [9, 10, 1, 44, 11, 58] continued training upon the weights of language-only LLMs [21] or vision LLMs without visual generation capabilities [33]. The main focus of our work is to effectively reuse pretrained LLMs for multimodal generation, particularly with the Transfusion recipe, without any compromise on the LLMs' existing text-only capabilities.[5]

7 Conclusion

We present LMFusion, a framework designed to equip LLMs with multimodal generative capabilities. By using Llama-3 for text generation and integrating parallel transformer modules for image diffusion, LMFusion efficiently reuses compute of pretrained LLMs. LMFusion's modular design enables independent developments of language and vision modules, de-risking the complexities associated with a large-scale, joint-modality pretraining. In this work, we reuse only the pretrained language components, which still requires substantial compute to train the image generation module from scratch. Future work could explore reusing pretrained image generation components as well.

[5]Concurrent to our work, [15] tackles multimodal generation via a joint attention mechanism between a DiT structure [59] for images and a frozen Llama-3 [8] for texts.

References

[1] 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.

[2] Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Xi Victoria Lin, Akshat Shrivastava, Liang Luo, Srinivasan Iyer, Mike Lewis, Gargi Ghosh, Luke Zettlemoyer, and Armen Aghajanyan. Moma: Efficient early-fusion pre-training with mixture of modality-aware experts, 2024.

[4] 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.

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

[6] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2022.

[7] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024.

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[9] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023.

[10] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

[11] 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.

[12] Sheng Shen, Zhewei Yao, Chunyuan Li, Trevor Darrell, Kurt Keutzer, and Yuxiong He. Scaling vision-language models with sparse mixture of experts. *arXiv preprint arXiv:2303.07226*, 2023.

[13] Junyi Chen, Longteng Guo, Jianxiang Sun, Shuai Shao, Zehuan Yuan, Liang Lin, and Dongyu Zhang. Eve: Efficient vision-language pre-training with masked prediction and modality-aware moe. *ArXiv*, abs/2308.11971, 2023.

[14] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models, 2024.

[15] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models, 2024.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[18] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[20] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[22] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[23] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[24] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, 2019.

[25] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[27] 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.

[28] 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.

[29] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[30] Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

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

[32] 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.

[33] 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. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

[34] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

[35] 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.

[36] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[37] xAI. Realworldqa, January 2024.

[38] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024.

[39] 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, 2024.

[40] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation, 2024.

[41] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024.

[42] 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, 2024.

[43] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023.

[44] 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.

[45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[46] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[47] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[48] 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.

[49] 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.

[50] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[51] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[52] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[53] Clare Lyle and Razvan Pascanu. Switching between tasks can cause ai to lose the ability to learn. *Nature*, 632(8026):745–747, 2024.

[54] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *ArXiv*, abs/2401.15947, 2024.

[55] Wanggui He, Siming Fu, Mushui Liu, Xierui Wang, Wenyi Xiao, Fangxun Shu, Yi Wang, Lei Zhang, Zhelun Yu, Haoyuan Li, Ziwei Huang, LeiLei Gan, and Hao Jiang. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis, 2024.

[56] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *ArXiv*, abs/2111.02358, 2021.

[57] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[58] Wanggui He, Siming Fu, Mushui Liu, Xierui Wang, Wenyi Xiao, Fangxun Shu, Yi Wang, Lei Zhang, Zhelun Yu, Haoyuan Li, et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis. *arXiv preprint arXiv:2407.07614*, 2024.

[59] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

NeurIPS Paper Checklist

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 accurately describe our core contributions.

   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: Please refer to the Conclusion section (§7).

   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: No formal 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: The code will be open-sourced and we describe all model experiments 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: Data and code will be open-sourced with instructions

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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See hyperparameter in §4.1

   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: NA

   Guidelines: We show the model performance with multiple checkpoints in Figure 3.

   - 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: Yes, we report computer resources in detail in §4.1.

   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: The research conforms with the 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 discuss these in §7.

    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: Yes, we describe safeguards in §4.1

    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: Yes, we describe licenses in §4.1.

    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: Yes, documentation is provided throughout the text.

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: There are no crowdsourcing experiments 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: There are no crowdsourcing experiments 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: The core method development in this research does not involve LLMs in non-standard components.

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.