Probabilistic Image Generation with LLM Priors via Structured Rectified Flow

Mykola Vysotskyi
SoftServe
Sadova St, 2D, Lviv, Lviv Oblast, 79021
mvysot@softserveinc.com

Zahar Kohut
SoftServe
Sadova St, 2D, Lviv, Lviv Oblast, 79021
zkohu@softserveinc.com

Anna-Alina Bondarets
SoftServe
Sadova St, 2D, Lviv, Lviv Oblast, 79021
anbondaret@softserveinc.com

Taras Rumezhak
SoftServe
Sadova St, 2D, Lviv, Lviv Oblast, 79021
trume@softserveinc.com

Volodymyr Karpiv
SoftServe
Sadova St, 2D, Lviv, Lviv Oblast, 79021
vkarpi@softserveinc.com

Abstract

Prior works have investigated the integration of large language models (LLMs) with rectified flow for image synthesis, but systematic studies remain scarce. In this study, we examine how controlling the interaction between stochastic and semantic inputs during encoding, while integrating them during decoding, influences the alignment between noised latents and LLM hidden states. Our investigation shows that architectural refinements, such as dual-stream encoding and single-stream decoding, can accelerate training and improve image quality relative to LLM-adapted rectified flow baselines. We evaluate our approach on standard image benchmarks and observe gains in both training speed and output detail preservation, indicating that structural choices in the integration of LLM features matter for probabilistic inference in generative modeling. Beyond empirical improvements, our findings contribute to understanding how foundation models trained on text can be adapted as structured probabilistic priors in visual domains. These results highlight a promising direction at the intersection of LLMs, rectified flow, and probabilistic image synthesis and motivate further explorations.

1 Introduction

Recent advances in multimodal image generation have demonstrated the potential of Large Language Models (LLMs) to process and synthesize complex visual data. Leveraging them in text-to-image generation enhances image quality and alignment with textual descriptions. The integration of LLMs improves the representation of text, resulting in a more accurate image synthesis. This approach also allows high-quality images to be generated with fewer training data and computational resources. Liu et al. [2024]

The JanusFlow model Ma et al. [2024] combines understanding and generation capabilities within a single framework, using a shared Large Language Model backbone. While effective for unified multimodal tasks, its design allocates capacity to both text and image generation. In this work, we

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Structured Probabilistic Inference & Generative Modeling.

focus exclusively on image synthesis, adapting the LLMs backbone as a high-capacity semantic prior for text–image alignment. By removing the text-generation pathway, we dedicate the model's capacity entirely to improving visual quality and prompt adherence, leveraging the pretrained LLM's broad knowledge and representational strength solely for guiding the image generation process.

Specifically, we introduce a dual-stream encoder that keeps noise and text in distinct pathways but retains inter-stream communication through shared attention layers, limiting interference between stochastic variation and semantic intent.

For the decoder, we adopt a single-stream design that unifies noise and text-conditioned representations, ensuring that semantic guidance and stochastic variation act together to shape the generative process. This integration improves coherence and alignment while preserving expressive diversity.

These adjustments allow for improved alignment between textual input and generated visual output while maintaining high-quality synthesis. By updating the encoder-decoder structure, we streamline the model to better suit the specific demands of visual generation, ultimately enhancing efficiency and output quality. To clarify *why* these architectural changes help, we include a simple representational analysis that inspects internal dynamics during rectified-flow inference while holding comparable components fixed.

In summary, our contributions are:

- We investigate how encoder–decoder design choices, specifically limiting interference between stochastic and semantic inputs at encoding while integrating them at decoding, affect the use of LLMs as structured probabilistic priors for image synthesis.
- We demonstrate that these design adjustments yield faster convergence and higher-quality outputs compared to existing LLM-adapted rectified flow baselines
- We show that these refinements improve the alignment of noised latents with LLM hidden states, leading to faster convergence and higher-quality, semantically coherent outputs compared to LLM-adapted rectified flow baselines.
- We introduce a lightweight *representational analysis* framework that isolates architectural effects during inference on the same 5,000-prompt MJHQ subset used for FID; the observed internal dynamics are consistent with the measured gains in convergence and image quality.

2 Related Works

2.1 Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) Ho et al. [2020] and Latent Diffusion Models (LDMs) Rombach et al. [2022] established the current foundation for image synthesis. Models such as Stable Diffusion and SDXL Podell et al. [2023a] demonstrated how compressed latent spaces enable efficient and high-quality image generation.

2.2 Text-conditioned Generation

Text conditioning plays a crucial role in guiding generative models toward semantically aligned outputs. CLIP Radford et al. [2021] embeddings are frequently used as a conditioning signal, as in Stable Diffusion. Extensions like classifier-free guidance Ho and Salimans [2022] and T2I-Adapter Mou et al. [2023] offer more flexible integration of text and other modalities into the generative process, allowing for fine-grained control and personalization.

Another example is unClip Ramesh et al. [2022], which proposed a two-stage model, composed of a prior that generates a CLIP Radford et al. [2021] image embedding given a text caption, and a decoder that generates an image conditioned on it. Imagen Saharia et al. [2022] experiments conducted a conclusion that a common LLMs pretrained on text-only datasets, such as T5-XXL Raffel et al. [2023], BERT Devlin et al. [2019], and CLIP Radford et al. [2021], are surprisingly effective at encoding text for image synthesis, improving the text alignment and generated images fidelity.

2.3 Cross-Modal Alignment

Several approaches have explored architectural designs that facilitate better alignment between vision and language. BLIP Li et al. [2022] and BLIP-2 Li et al. [2023] propose strategies for pre-training and bridging modalities through lightweight adapters or vision-language transformers. Pix2Seq Chen et al. [2022] and CoCa Yu et al. [2022] demonstrate that unified encoder-decoder frameworks can learn cross-modal mappings effectively. Our work builds on this line by proposing a dedicated dual-stream encoder and single-stream decoder architecture tailored to the LLM-guided image generation task, improving alignment between noised latents and language features while maintaining high image fidelity.

2.4 LLM-Driven Image Generation

Large Language Models (LLMs) have increasingly been adopted in image generation pipelines to enrich semantic understanding and multimodal alignment Koh et al. [2023], Dong et al. [2024], Jin et al. [2024], Ge et al. [2023a,b, 2025].

GILL Koh et al. [2023] is one of the first models to propose a fusion of text-only LLM and a pre-trained Stable Diffusion Rombach et al. [2022] by aligning their embedding spaces. DreamLLM Dong et al. [2024] is the first approach to combine multi-modal LLM for free-form interleaved content and image generation in particular. An image tokenizer SEED is introduced in Ge et al. [2023a] to be coupled with LLMs for both image-to-text and text-to-image generation tasks. SEED-LLaMA Ge et al. [2023b] and SEED-X Ge et al. [2025] improvements of SEED are incorporating a SOTA foundation language model LLaMa Touvron et al. [2023] into the pipeline. LaVIT Jin et al. [2024] also combines LLaMa Touvron et al. [2023] in a pipeline to serve as a multi-modal generalist to perform both multi-modal comprehension and generation tasks.

JanusFlow Ma et al. [2024], a recent model from DeepSeek-LLM, integrates Rectified Flow Liu et al. [2022a] with a shared LLM backbone for joint understanding and generation. While effective, its general-purpose design can limit performance when compared with concurrent works in image generation task.

3 Background

3.1 Rectified Flow

Rectified Flow Liu et al. [2022a], Lipman et al. [2023] is a generative modeling approach that learns a continuous transformation from a simple prior distribution p_0, typically a standard Gaussian $\mathcal{N}(0, I)$, to the target data distribution using an ordinary differential equation (ODE).

Rectified Flow models the transformation of continuous d-dimensional data points $\mathbf{x} = (x_1, ..., x_d)$, which follow an unknown distribution p_1, by introducing a parameterized velocity field v_θ that dictates their evolution over time $t \in [0, 1]$:

$$\frac{dz_t}{dt} = v_\theta(z_t, t),$$

where $z_t = tx + (1 - t)z_0$ and $z_0 \sim p_o$

The velocity function v_θ is optimized to minimize the deviation between its predicted velocity and the true displacement direction between samples drawn from p_0 and p_1. The training objective is formulated as follows:

$$\min_\theta \mathbb{E}\left[||v_\theta(z_t, t) - (\mathbf{x} - z_0)||^2\right]$$

Once trained, image generation is performed by integrating the learned velocity field v_θ to transport a sample from p_0 to the target distribution p_1. Given an initial latent variable $z_0 \sim p_0$, corresponding data sample is obtained by solving the ODE:

$$z_1 = z_0 + \int_0^1 v_\theta^{opt}(z_t, t)dt$$

In practice, numerical integration is carried out in an iterative manner using solvers such as Euler's method:

$$z_{t+\Delta t} = z_t + v_\theta^{opt}(z_t, t)\Delta t,$$

e where Δt is a small step size. z_t is progressively updated until z_1, approximating a sample from a desired distribution p_1.Esser et al. [2024]

3.2 Autoregressive Language Modeling

Autoregressive models predict each token in a sequence $w_1, w_2, ..., w_T$ based on previous tokens. The joint probability is factorized as:

$$P(w_1, ..., w_T) = \prod_{t=0}^{T-1} P(w_{t+1} \mid w_1, ..., w_t)$$

Training involves maximizing the log-likelihood over N sequences:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T_i-1} \log P(w_{t+1}^{(i)} \mid w_1^{(i)}, ..., w_t^{(i)})$$

Here, T_i is the length of the i-th sequence. Model parameters θ are optimized using gradient-based methods.

4 Methodology

Our methodology is guided by the hypothesis that large language models, trained purely on text, implicitly learn probabilistic structures that can be transferred to generative inference in other modalities. Prior work such as JanusFlow Ma et al. [2024] demonstrated that LLM hidden states can serve as a joint representational prior for both language and vision. We extend this line of exploration by asking whether alternative architectural designs — specifically, separating noise and text processing at the encoding stage but recombining them at decoding — can more effectively expose and exploit these hidden capacities. In this way, our work should be seen less as proposing a new end-to-end system, and more as investigating how foundational LLM representations can be adapted for probabilistic image generation.

Our approach builds upon advancements in large-scale generative models by integrating a dual-stream encoder and a single-stream decoder, leveraging insights from recent works on scaling vision and rectified flow transformers Dehghani et al. [2023], Esser et al. [2024]. Specifically, we modify traditional architectures to enhance both **Fréchet Inception Distance (FID)** Heusel et al. [2018] and **Contrastive Language-Image Pretraining (CLIP)** Radford et al. [2021] scores, ensuring superior image generation quality and semantic alignment.

Both encoder and decoder operate in a latent space of the pre-trained SDXL-VAE Podell et al. [2023b] to achieve higher computational efficiency.

4.1 Architectural Transition

In contrast to reliance on ConvNeXt-based encoders and decoders Liu et al. [2022b], our revised architecture (Figure 5) employs a more scalable design for high-fidelity generation. Dehghani et al. [2023], Esser et al. [2024]. The key changes include:

- Adopting a dual-stream encoder to improve feature separation and representation quality.Esser et al. [2024]
- Implementing a single-stream decoder that enhances semantic alignment through text-conditioned decoding.Dehghani et al. [2023]
- Transitioning from convolutional architectures to transformer-based layers, ensuring improved scalability and expressive power.

4.2 Dual-Stream Encoding

Dual-stream encoder separates the processing of textual and noise information, improving representation learning and generation fidelity. This consists of two distinct streams:

- **Noise Stream**: Encodes the random latent variables, capturing stochastic variations crucial for image synthesis.
- **Text Stream**: Encodes input textual prompts separately, preserving their structural and semantic information.

The two streams are later merged via a learned cross-attention mechanism and passed into the Large Language Model (LLM), ensuring a controlled and context-aware generative process.

This design follows prior evidence that text and image (or noise) embeddings differ substantially in their representational structure and are better handled by distinct parameterizations. In practice, this corresponds to processing each modality with separate streams, while still allowing them to interact through shared attention layers. Such separation enables each stream to maintain its own representational integrity, reducing interference and yielding more stable mappings in the rectified flow setting. Esser et al. [2024]

The encoder architecture, presented on (Figure 3), realizes the dual-stream design introduced above.

4.3 LLM

Our model leverages DeepSeek-LLM DeepSeek-AI et al. [2024] as its backbone, utilizing its knowledge to enhance text-conditioned image synthesis. We specifically adopt the pre-trained variant from the JanusFlow model (1.3B version) Ma et al. [2024], focusing on alignment and integration of our encoder-decoder components with its latent space for even more effective image generation.

4.4 Single-Stream Decoding

In the original setup, the output decoder only processed noise tokens to generate images. However, we introduce a single-stream decoder that also integrates text-informed outputs, reinforcing semantic consistency. The single-stream decoder:

1. Processes a unified representation where noise and text-conditioned outputs are fused.
2. Learns to incorporate text-driven refinements into the image generation process, improving coherence.
3. Retains stochastic expressiveness from the noise stream while ensuring adherence to textual guidance.

This fusion step operationalizes the idea that semantic and stochastic factors should ultimately converge on a single generative path. Whereas dual-stream encoders emphasize disentanglement, a single decoding stream ensures that final outputs reflect both variability and semantic precision without misalignment. In this sense, our methodology investigates how LLM-driven representations can balance randomness and structure in probabilistic inference for image synthesis.

A detailed implementation of the single-stream decoder is provided in (Figure 4).

The motivation for this transition is theoretical as well as empirical. A dual-stream encoder enables probabilistic disentanglement: noise tokens model stochastic variability, while text tokens preserve structured semantic intent. Esser et al. [2024] Combination in a single-stream decoder then sustains coherence by enforcing a shared generative trajectory.

Overall, it is a structural intervention that probes whether LLM latent representations can better guide visual inference when the stochastic and semantic signals are first treated independently.

5 Training

The model learns to transform random noise into images, conditioned on text descriptions, by optimizing both encoder-decoder and LLM components.

5.1 Training objective

The training objective focuses on predicting the velocity of the latent variable transformation during the generation process. At each training step, we sample a timestep t from a logit-normal distribution,

and the model attempts to predict the velocity of the transformation, which guides the latent variable's movement toward the target image. The latent variable z_t is computed as a linear interpolation between the initial latent variable z_0 and the target image $\mathbf{x}$, as follows:

$$z_t = tx + (1-t)z_0$$

where t is drawn at each step and controls the interpolation between the two extremes. This means that as t progresses from 0 to 1, the model gradually refines the latent variable toward a more accurate representation of the target image.

The objective function of the model can be expressed as:

$$\min_{\theta} \mathbb{E}\left[||v_\theta(z_t, t) - (\mathbf{x} - z_0)||^2\right]$$

where $v_\theta(z_t, t)$ represents the model's prediction of the velocity at timestep t and the term $(\mathbf{x} - z_0)$ is the direction from the initial latent image to the target image.

By sampling different values of t throughout training, the model learns to predict the correct velocity at each timestep, gradually transforming the latent variable z_t to closely match the target image. This training process allows the model to learn how to efficiently navigate the latent space, ensuring accurate image generation aligned with the input text descriptions.

5.2 Training stages

We employ a two-stage training process (Figure 6) for our model.

Stage 1. We focus on aligning the encoder-decoder architecture with the latent space of the pretrained Janus LLM DeepSeek-AI et al. [2024]. During this stage, we do not train the LLM itself; instead, we optimize the encoder-decoder components to match the representation space of the LLM. This helps in establishing a strong foundation for the subsequent image generation process.

Stage 2. Then we train the entire model, including the backbone. The LLM is fine-tuned specifically for image generation, allowing the encoder-decoder components to leverage the enhanced representation capabilities of the LLM. This stage enables the full potential of the LLM to improve image quality and alignment with textual descriptions, resulting in a fully integrated model optimized for high-quality image synthesis.

6 Experiment

In this section, we present the experimental setup and results for evaluating the performance of our image generation model. We assess its capabilities in generating high-quality, diverse images across various domains, evaluated using standard metrics.

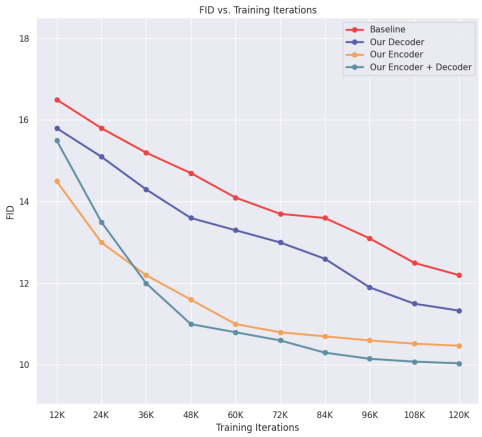
6.1 Experiment setup and training data

We train on **3M** image–text pairs drawn from two sources: *DALL-E 3 High-Quality Captions* Egan et al. [2024] (MIT) and a curated high-aesthetic *LAION–COCO* subset (Apache-2.0), mixed at a **1:2** ratio. Images are center-cropped and resized to **384×384**; captions are used as provided. Dataset provenance, benchmark prompt isolation, and leakage prevention are detailed in Appendix F.

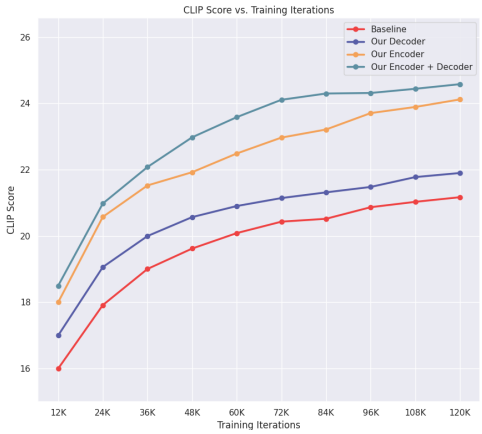
Training uses **2×8 H100** GPUs with PyTorch DDP, following a two-stage schedule; the full configuration (optimizer, learning rates, batch sizes, schedules) and stage overview are in the Appendix (Table 2, Fig. 6). The end-to-end run time is ∼**8.5 days**.

6.2 Evaluation

For evaluation, we assess the model's performance using two key metrics: **CLIP Similarity** Radford et al. [2021], **Fréchet Inception Distance (FID)** Heusel et al. [2018]. These metrics are computed every 12,000 training iterations on a **MJHQ FID-30k** Li et al. [2024] that is common for text-to-image models benchmarking. CLIP-ViT-Large-Patch/14 version of CLIP was used. We compare:



(a) FID vs. Training Steps



(b) CLIP Similarity vs. Training Steps

Figure 1: Quantitative evaluations across different model configurations. (a) FID vs. Training Steps. (b) CLIP Similarity vs. Training Steps.

- the original JanusFlow 1.3B model Ma et al. [2024]. We show comparable performance with this, showing that our retraining is fair and correct.
- baseline that leverages pretrained LLM and freshly initialized encoder-decoder proposed in Ma et al. [2024].
- a modified version with a proposed single-stream decoder,
- a modified version with a proposed dual-stream encoder,
- a fully refined model incorporating both encoder and decoder modifications.

This benchmarking provides insight into the effects of architectural modifications on image quality and text alignment.

See Appendices F and G for more details on the datasets and key evaluation metrics accordingly.

In addition to quantitative evaluation, we present qualitative results, showcasing generated images for a set of complex prompts. These examples highlight the model's ability to capture fine-grained details, maintain spatial awareness, and accurately represent multiobject compositions. By comparing outputs from different models, we visually assess improvements in coherence, object interaction, and adherence to textual descriptions.

6.3 Quantitative results

Our quantitative results indicate that the proposed modifications significantly accelerate convergence across all evaluation metrics. As shown in (Figure 1b), our model achieves higher CLIP similarity with fewer training steps compared to the baseline, demonstrating improved text-image alignment. In terms of image quality, (Figure 1a) shows a faster reduction in FID and confirming improvements in realism and diversity.
Upon completion of training, our model achieves CLIP similarity of 24.58, and 10.04 FID, as detailed in (Table 1), further validating the effectiveness of refining both the encoder and decoder.

6.4 Qualitative results

The qualitative results, presented in (Figure 8), illustrate the clear improvements in the performance of our model over the baseline. Our model demonstrates enhanced spatial awareness, ensuring that objects are positioned more accurately and consistently within the scene. In addition, it is demonstrating a refined understanding of object shapes. There is also a notable improvement in prompt adherence, with the generated images aligning more closely with the provided textual descriptions. These advancements lead to higher-quality images that are more realistic and true to the input prompts, showcasing the effectiveness of our approach.

Table 1: Quantitative comparison of different model configurations (after all 120k training steps). Higher CLIP Similarity indicates better text-image alignment and diversity, while lower FID represents improved image quality. Arrows indicate whether lower (↓) or higher (↑) values are better. Values are reported as mean ± standard deviation across 5 random seeds.

Model	FID ↓	CLIP ↑
JanusFlow	9.51 ± 0.12	26.02 ± 0.08
Baseline	11.99 ± 0.21	21.17 ± 0.15
Changed Decoder	11.33 ± 0.18	21.90 ± 0.10
Changed Encoder	10.47 ± 0.14	24.12 ± 0.12
Changed both Encoder and Decoder	10.04 ± 0.11	24.58 ± 0.09

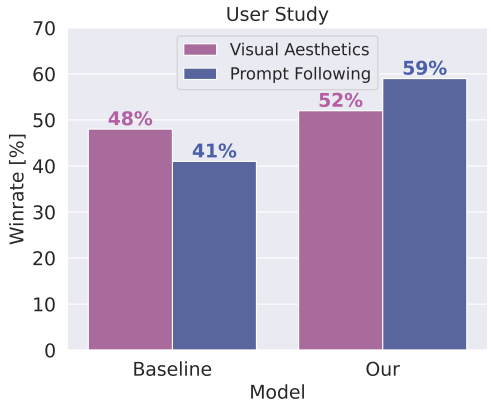


Figure 2: Human evaluation against the baseline JanusFlow model. Our modified architecture significantly increases prompt-following while only marginally affecting visual quality.

We also conduct a user study in (Figure 2) to compare the results generated from the modified model(enhanced both encoder and decoder) with the baseline. We produced a dataset of 40 images, and 52 users were asked to select the best model based on two criteria: visual aesthetics and prompt following. To evaluate human preference, we asked the reviewers to answer questions about each pair of generated images with two models. More details on this user study are given in the Appendix E.

7 Discussion

Reference model and scope. We include *JanusFlow* only to calibrate our implementation and to show that our reproduction falls within the expected performance range. As already noted in the paper, JanusFlow was trained substantially longer on a much larger dataset (approximately 70M pairs) compared to our 3M-pair setup. It is therefore not a compute- or data-matched baseline.

Ablation results Table 1. Across metrics (FID ↓ and CLIP ↑), the structured variant with *dual-stream encoder* and *single-stream decoder* ("Both") consistently outperforms (i) our re-implemented baseline and (ii) single-change ablations ("Encoder-only" and "Decoder-only"). The encoder-side modification contributes more than the decoder-only change, indicating that separating modalities during encoding has a stronger effect on alignment and signal disentanglement; nevertheless, combining both yields the strongest gains, suggesting complementary benefits.

Learning dynamics Figure 1. The training curves show faster convergence for the structured model: it reaches the baseline's final quality in notably fewer steps and exhibits more stable trajectories. Early-phase CLIP improvements correlate with later FID reductions, consistent with the hypothesis that separating stochastic (noise) and semantic (text) pathways during encoding produces cleaner gradients and a more sample-efficient learning signal.

8 Limitations & Societal Impact

Limitations While our approach demonstrates significant improvements in image quality and training efficiency, several limitations remain. One key limitation is the reliance on high-quality pre-trained LLMs, which significantly impacts the performance of our model. The effectiveness of our framework is contingent on the underlying backbone's ability to understand and represent textual data accurately; thus, the model's performance may be constrained by the quality and domain of the pre-trained language model. Additionally, the training data required for fine-tuning can be quite extensive, and the model's generalization capabilities may be limited by the diversity and coverage of the dataset used. Finally, the computational cost training remains a challenge, particularly when fine-tuning large-scale LLMs or working with larger datasets.

Societal Impact While our focus is on technical improvements in image quality and training efficiency, it is vital to acknowledge broader societal implications. Enhanced prompt adherence may increase misuse risks, such as deepfakes and misinformation, threatening privacy and public trust. The model's efficiency and portability democratize access, but also heighten potential for abuse. Additionally, reliance on large datasets and pre-trained models risks reinforcing existing biases. We urge the community to develop safeguards and ethical frameworks to ensure responsible deployment of generative technologies.

Safeguards. To mitigate potential risks, we adopt the following safeguards:

- **No model release:** We do not plan to release trained checkpoints, preventing uncontrolled distribution and misuse.
- **Bias and fairness:** Preliminary evaluation indicates that training data biases (e.g., stereotypes in gender/occupation) may propagate into outputs; we explicitly acknowledge this risk even without public release.

9 Conclusion

In this work, we have introduced a refined approach to image generation by enhancing the encoder-decoder architecture of the JanusFlow model. Our proposed changes lead to improvements in image quality and alignment with textual prompts, as demonstrated through qualitative and quantitative evaluations. The model shows faster convergence and higher performance across key metrics, compared to the baseline. These results highlight the potential of our approach for advancing text-to-image generation, particularly in the context of LLM-backed image generation frameworks.

References

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection, 2022. URL https://arxiv.org/abs/2109.10852. arXiv preprint arXiv:2109.10852.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism, 2024. URL https://arxiv.org/abs/2401.02954. arXiv preprint arXiv:2401.02954.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL https://arxiv.org/abs/2302.05442. arXiv preprint arXiv:2302.05442.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805. arXiv preprint arXiv:1810.04805.

Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Dreamllm: Synergistic multimodal comprehension and creation, 2024. URL https://arxiv.org/abs/2309.11499. arXiv preprint arXiv:2309.11499.

Ben Egan, Alex Redden, XWAVE, and SilentAntagonist. Dalle3 1 Million+ High Quality Captions, May 2024. URL https://huggingface.co/datasets/ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206. arXiv preprint arXiv:2403.03206.

Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model, 2023a. URL https://arxiv.org/abs/2307.08041. arXiv preprint arXiv:2307.08041.

Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer, 2023b. URL https://arxiv.org/abs/2310.01218. arXiv preprint arXiv:2310.01218.

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, 2025. URL https://arxiv.org/abs/2404.14396. arXiv preprint arXiv:2404.14396.

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, 2018. arXiv preprint arXiv:1706.08500.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. arXiv preprint arXiv:2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. arXiv preprint arXiv:2006.11239.

Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, Di Zhang, Wenwu Ou, Kun Gai, and Yadong Mu. Unified language-vision pretraining in llm with dynamic discrete visual tokenization, 2024. URL https://arxiv.org/abs/2309.04669. arXiv preprint arXiv:2309.04669.

Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models, 2023. URL https://arxiv.org/abs/2305.17216. arXiv preprint arXiv:2305.17216.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024. URL https://arxiv.org/abs/2402.17245.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086. arXiv preprint arXiv:2201.12086.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597. arXiv preprint arXiv:2301.12597.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747. arXiv preprint arXiv:2210.02747.

Guangyi Liu. "laion-coco-aesthetic" dataset. https://huggingface.co/datasets/guangyil/laion-coco-aesthetic.

Mushui Liu, Yuhang Ma, Yang Zhen, Jun Dan, Yunlong Yu, Zeng Zhao, Zhipeng Hu, Bai Liu, and Changjie Fan. Llm4gen: Leveraging semantic representation of llms for text-to-image generation, 2024. URL https://arxiv.org/abs/2407.00737. arXiv preprint arXiv:2407.00737.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022a. URL https://arxiv.org/abs/2209.03003. arXiv preprint arXiv:2209.03003.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022b. URL https://arxiv.org/abs/2201.03545. arXiv preprint arXiv:2201.03545.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2411.07975. arXiv preprint arXiv:2411.07975.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2302.08453. arXiv preprint arXiv:2302.08453.

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

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023b. URL https://arxiv.org/abs/2307.01952. arXiv preprint arXiv:2307.01952.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020. arXiv preprint arXiv:2103.00020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683. arXiv preprint arXiv:1910.10683.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125. arXiv preprint arXiv:2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. arXiv preprint arXiv:2112.10752.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487. arXiv preprint arXiv:2205.11487.

Christoph Schuhmann. Aesthetic score predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL https://arxiv.org/abs/2210.08402. arXiv preprint arXiv:2210.08402.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971. arXiv preprint arXiv:2302.13971.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. URL https://arxiv.org/abs/2205.01917. arXiv preprint arXiv:2205.01917.

A Architectures

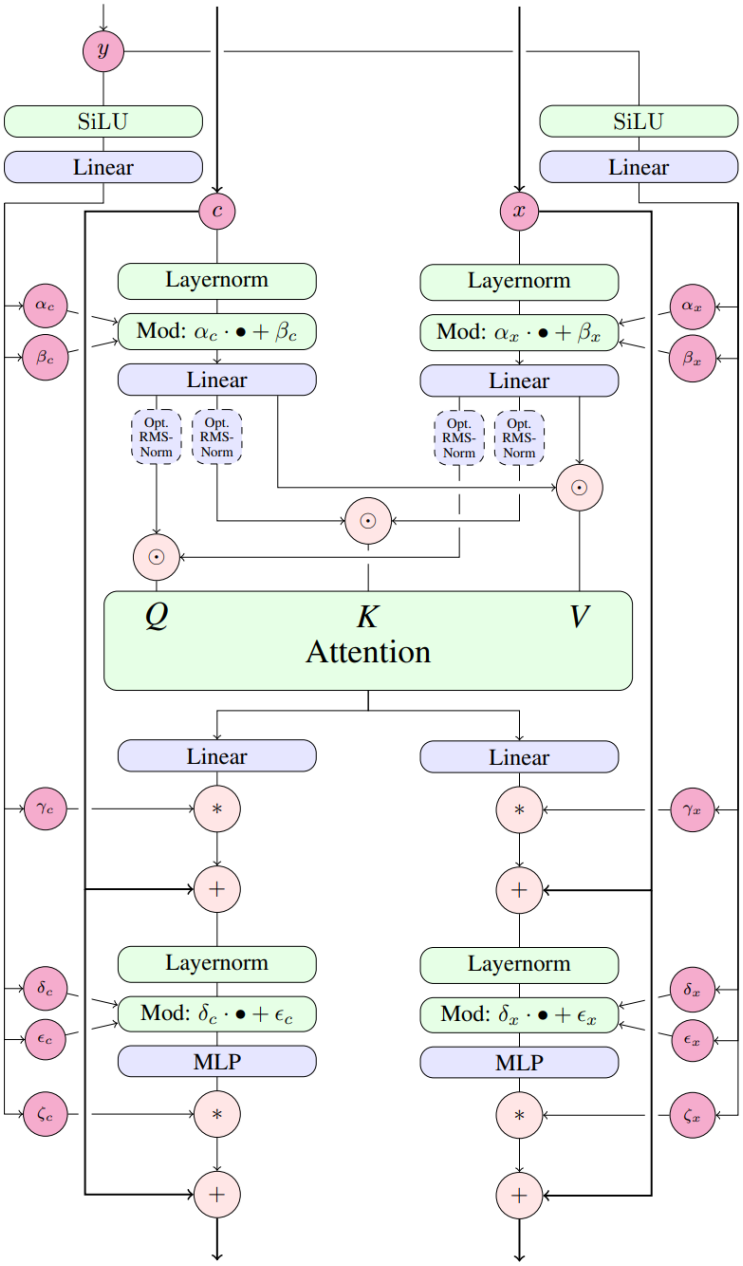


Figure 3: Dual-Stream Encoder architecture. Originally, MM-DiT block from Esser et al. [2024].

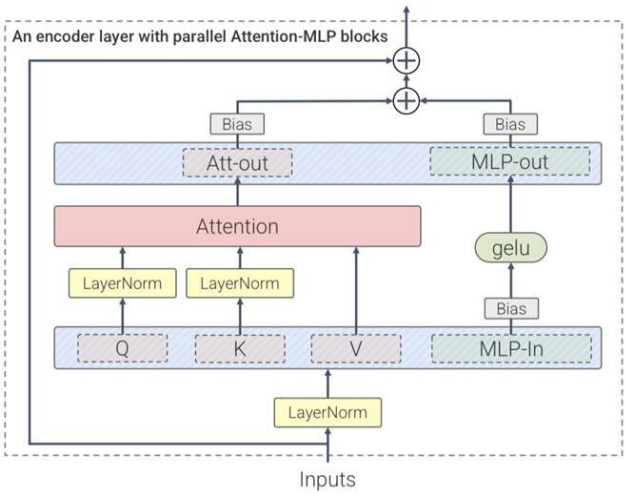


Figure 4: Single-Stream Decoder architecture. Originally, parallel ViT-22B layer Dehghani et al. [2023].

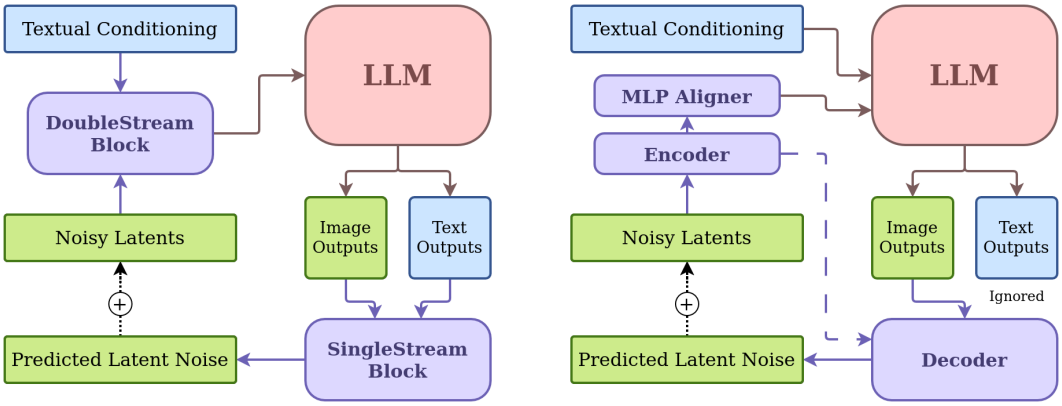


Figure 5: Comparison of our proposed (left) and the original Janus architectures (right). Our model employs a Double-Stream Block Encoder for joint text and image encoding, while the original method utilizes an MLP Aligner and Encoder to map latents before passing them to an LLM. The single-stream decoder in our approach processes both noise and text LLM outputs, while the original decoder handles only output noise.

B Training Details

Hardware & framework. 2 nodes $\times$ 8 NVIDIA H100 GPUs; PyTorch Distributed Data Parallel (DDP).
Runtime. End-to-end training $\sim$8.5 days.
Resolution. All training images are 384$\times$384.
Schedule & hyperparameters. Two-stage recipe; full settings in Table 2, overview in Fig. 6.

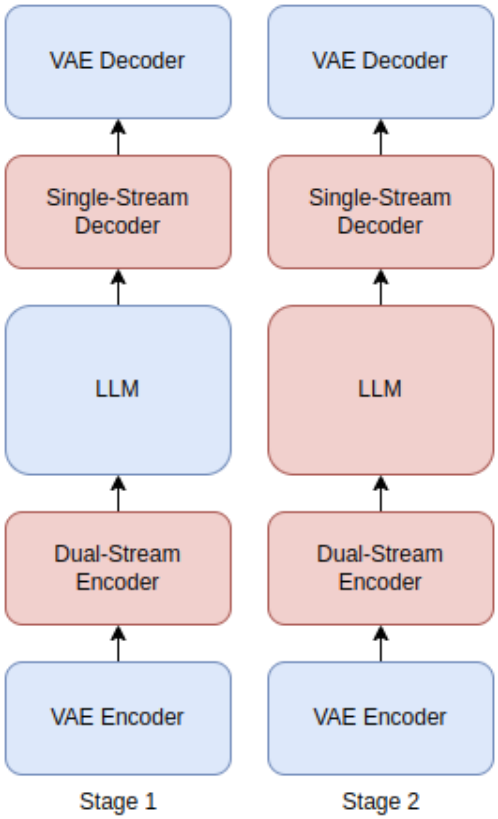


Figure 6: Stage 1 (left): Align the encoder-decoder with the pretrained Janus LLM's latent space, without training the LLM. Stage 2 (right): Fine-tune the entire model, including the LLM, for image generation. (blue - weights are frozen, red - component is being trained)

Table 2: Hyperparameters used for Stage 1 and Stage 2 of training. Shared values are centered, while stage-specific values are listed separately.

Hyperparameter	**Stage 1**	**Stage 2**
Learning Rate	1×10^{-4}	
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.999$)	
Scheduler	Constant	Constant
Training Steps	10k	100k
Warmup Steps	2000	
Effective Batch Size	512	
Weight decay	0.0	

C Sampling Protocol

For all reported quantitative and qualitative experiments, we used the following sampling configuration unless otherwise noted:

- **Number of steps:** 30
- **Sampler:** Euler solver
- **Classifier-Free Guidance (CFG) scale:** 2.0

C.1 Background on the Euler Sampler

Euler sampling is one of the simplest numerical solvers for ordinary differential equations (ODEs), commonly used in rectified flow and diffusion-based generative models. It approximates the continuous trajectory of the rectified flow by discretizing time into steps and updating the latent representation using local gradient information. While simple, Euler's method is computationally efficient and provides stable results when combined with rectified flow guidance.

C.2 Background on Classifier-Free Guidance (CFG)

Classifier-Free Guidance (CFG) is a technique that improves conditional generation by interpolating between unconditional and conditional model outputs. Specifically, the model is run once with the conditioning signal (e.g., text prompt) and once without it; the two predictions are linearly combined using a guidance scale γ. A higher γ encourages stronger alignment with the condition at the cost of sample diversity. In our experiments, $\gamma = 2.0$ provided a good balance between faithfulness to text and image quality.

C.3 Algorithm: Sampling Procedure

Algorithm 1 Rectified Flow Sampling with Euler and CFG

Require: Initial noise latent x_0, text embedding c, stepsse $T = 30$, CFG scale $\gamma = 2.0$
1: **for** $t = 0$ to $T - 1$ **do**
2: Compute conditional velocity $v_{\text{cond}} = f(x_t, c, t)$
3: Compute unconditional velocity $v_{\text{uncond}} = f(x_t, \varnothing, t)$
4: Combine: $v = v_{\text{uncond}} + \gamma \cdot (v_{\text{cond}} - v_{\text{uncond}})$
5: Euler update: $x_{t+1} = x_t + \Delta t \cdot v$
6: **end for**
7: **Return** x_T (decode with VAE)

D Qualitative results

Figure 7: Sample images generated by our model. These images demonstrate the enhanced image generation capabilities achieved using our LLM-backed Rectified Flow with transformer-based encoder and decoder.

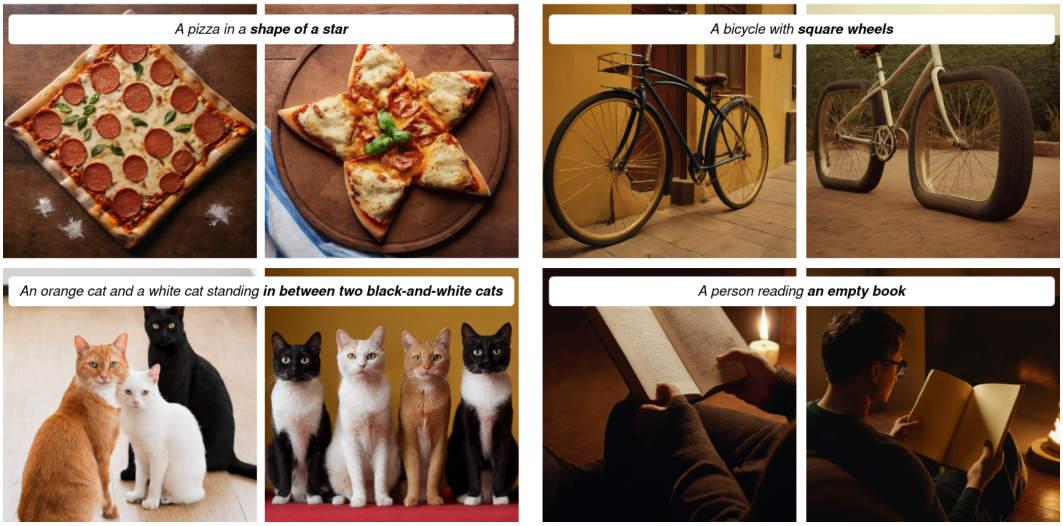


Figure 8: Qualitative comparison of generated images. Each pair shows an image generated by the baseline (left) and our improved model (right) for the same input prompt. Our model demonstrates enhanced spatial coherence, finer detail preservation, and improved text–image alignment.

E Human Study Design and Details

We conducted a human evaluation study to assess prompt alignment and perceptual quality of images generated by our model compared to baselines.

E.1 Participants

The study involved 52 participants, all of whom were employees of our company. Participation was strictly voluntary, and no monetary or material compensation was provided. Distributed using corporate email.

E.2 Procedure

Each participant was shown a set of 40 randomly selected prompts with two corresponding images (ours vs. baseline). The order of images (left/right) was randomized per comparison to mitigate positional bias. Similarly, the order of prompts was randomized for each participant to reduce ordering effects. Participants were not informed about which model generated each image, ensuring that comparisons were conducted in a blind setting.

E.3 Instructions

For each prompt and pair of images, participants were asked to answer two multiple-choice questions:

1. **Prompt alignment:** "Which image looks more representative of the text shown above and faithfully follows it? (Left / Right)"
2. **Aesthetic quality:** "Given the prompt, which image is of higher quality and aesthetically more pleasing? (Left / Right)"

Responses were analyzed using a two-sided binomial test with null hypothesis $p = 0.5$, and 95% Wilson confidence intervals were computed for the preference rate of our model.

E.4 Question 1: Prompt Faithfulness

> *"Which image looks more representative of the text shown above and faithfully follows it? (Left / Right)"*

- Votes for our model: 1227
- Votes for baseline: 853
- Win-rate for our model: 59%
- Binomial test ($p = 0.5$): p-value $= 1.198 \times 10^{-16}$
- Wilson confidence interval (95%): $(0.5686, 0.6109)$

E.5 Question 2: Aesthetic Quality

> *"Given the prompt, which image is of higher quality and aesthetically more pleasing? (Left / Right)"*

- Votes for our model: 1082
- Votes for baseline: 998
- Win-rate for our model: 52%
- Binomial test ($p = 0.5$): p-value $= 0.0344$
- Wilson confidence interval (95%): $(0.4987, 0.5416)$

E.6 Ethics and Consent

All participants gave informed consent to voluntarily take part in the study. The study was conducted internally and did not involve external subjects or sensitive data. No personally identifiable information was collected, and results were aggregated anonymously. Since the evaluation was conducted on a voluntary basis with company employees and did not involve vulnerable populations or sensitive topics, no formal IRB approval was required.

E.7 Analysis

For each question, we aggregated the votes across participants and computed the percentage of preferences for our model versus the baseline. Statistical significance was assessed using a binomial test at the $p < 0.05$ level.

F Dataset

F.1 Sources, composition, and preprocessing

Sources & licenses. DALL-E 3 High-Quality Captions (MIT) and a "laion-coco-aesthetic" subset Liu derived from LAION-5B (Apache-2.0) Schuhmann et al. [2022].
Composition. We mix datasets at a 1:2 ratio (DALL-E:LAION–COCO) to form **3M** image–text pairs. Typical source resolutions are 1024^2 and 1792×1024.
Preprocessing. Center-crop and resize all images to **384**$\times$**384**; captions are used as provided by the datasets.

F.2 Additional characteristics of the sources

DALL-E 3 High-Quality Captions. This corpus consists primarily of AI-generated images (largely DALL-E 3, with contributions from MidJourney and Stable Diffusion) paired with detailed, model-written captions. It spans a wide range of subject matter (objects, scenes, portraits), artistic styles (photorealism, digital art, watercolor, anime), and compositional patterns (single-object, multi-entity, relational prompts). The release is filtered for duplicates, non-AI content, and inappropriate material. The captions are typically longer and more attribute-rich than web captions, which we find beneficial for conditioning.

LAION–COCO aesthetic subset. We draw from the "laion-coco-aesthetic" split Liu, a high-aesthetic slice of LAION-5B Schuhmann et al. [2022] scored via CLIP-based aesthetics Schuhmann. Images are real-world and higher-variance in camera viewpoint, lighting, and texture statistics compared to synthetic sources. Captions are synthetic but concise. We select a higher-aesthetic tranche to complement DALL-E imagery, balancing stylized variety with natural image statistics.

F.3 Benchmark prompt isolation.

All evaluation prompts were drawn from the MJHQ FID-30k benchmark. Prior to training, we hashed all evaluation prompts and removed any entries from our training corpus that contained exact or near-duplicate text spans (using case-insensitive matching and cosine similarity thresholding at 0.95 over sentence embeddings).

F.4 Prompt Leakage Prevention

A critical concern in evaluating text-to-image generative models is the potential for *prompt leakage*, where benchmark prompts inadvertently overlap with training data. Such overlap may lead to inflated performance metrics and undermine fair comparison.

To mitigate this risk, we adopted the following protocol:

1. **Dataset provenance and licensing.** Our training data consisted of the DALL-E 3 High-Quality Captions dataset (MIT license) and a curated subset of the LAION-COCO dataset (Apache 2.0 license).
2. **Randomized verification.** A random sample of 1,000 training prompts was manually inspected against the evaluation set to confirm absence of near-overlap in phrasing and semantic structure.

F.5 Human Study Prompts

For the human preference study, evaluation prompts were not taken from any existing dataset but were instead **generated with GPT-4o**. This approach allowed us to create diverse and natural instructions while avoiding contamination from commonly used benchmarks.

To ensure reproducibility and fairness:

1. **Prompt generation.** We instructed GPT-4o with prompt designed to elicit varied, semantically rich descriptions spanning objects, styles, and compositional requirements:

   "Generate 40 diverse text prompts suitable for evaluating text-to-image models. Prompts should vary in style (photorealistic, artistic, abstract), complexity (single object vs. multiple entities with relations), and domains (nature, urban, science fiction, cultural). Avoid clichés and keep prompts between 10–25 words."

2. **Leakage avoidance.** After prompt generation, we ran the same hashing and semantic similarity filtering procedure as with the benchmark set to confirm that none of these GPT-4o-generated prompts overlapped with our training data corpus.

G Key Evaluation Metrics

G.1 Fréchet Inception Distance (FID) Computation

For distribution-level image quality, we report the Fréchet Inception Distance (FID). Specifically:

- **Backbone:** Inception-V3, pool3 activations (2048-D).
- **Computation:** Let μ_r, Σ_r be the mean and covariance of reference features, and μ_g, Σ_g those of generated features. FID is

$$\mathrm{FID}(\mathcal{R}, \mathcal{G}) \;=\; \|\mu_r - \mu_g\|_2^2 \;+\; \mathrm{Tr}\big(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}\big).$$

  For numerical stability we add a small diagonal term ($10^{-6} I$) to covariances prior to the matrix square root.

G.2 CLIP Score Computation

For image–text alignment evaluation, we report the CLIP score as a cosine similarity between image and text embeddings, following standard practice. Specifically:

- **Model:** We use the OpenAI CLIP ViT-L/14@336px encoder.
- **Computation:** Given a generated image I and its corresponding text prompt T, we obtain the normalized image embedding $f(I)$ and text embedding $g(T)$. The CLIP score is then

$$\mathrm{CLIP}(I, T) = \frac{f(I) \cdot g(T)}{\|f(I)\| \, \|g(T)\|} \in [0, 1].$$

  Note: All CLIP scores reported in this paper are multiplied by 100 for readability.
- **Aggregation:** For each model, we report the mean CLIP score over the entire evaluation set (30k prompts from MJHQ).
- **Variance reporting:** We additionally compute the standard deviation across random seeds.