Note: A summary is provided on pg 13. It is recommended to go through the summary before reading the full paper

JanusFlow: Harmonizing Autoregression and Rectified Flow for Unified Multimodal Understanding and Generation

Yiyang Ma[1,2] Xingchao Liu[1,†] Xiaokang Chen[1,†] Wen Liu[1,†] Chenyue Wu[1,3] Zhiyu Wu[1,2] Zizheng Pan[1] Zhenda Xie[1] Haowei Zhang[1] Xingkai Yu[1] Liang Zhao[1] Yisong Wang[1,4] Jiaying Liu[2] Chong Ruan[1,‡]

[1]DeepSeek-AI [2]Peking University [3]The University of Hong Kong [4]Tsinghua University
[†]Equal contribution, [‡]Corresponding author

Project Page: https://github.com/deepseek-ai/Janus

arXiv:2411.07975v1 [cs.CV] 12 Nov 2024

Abstract

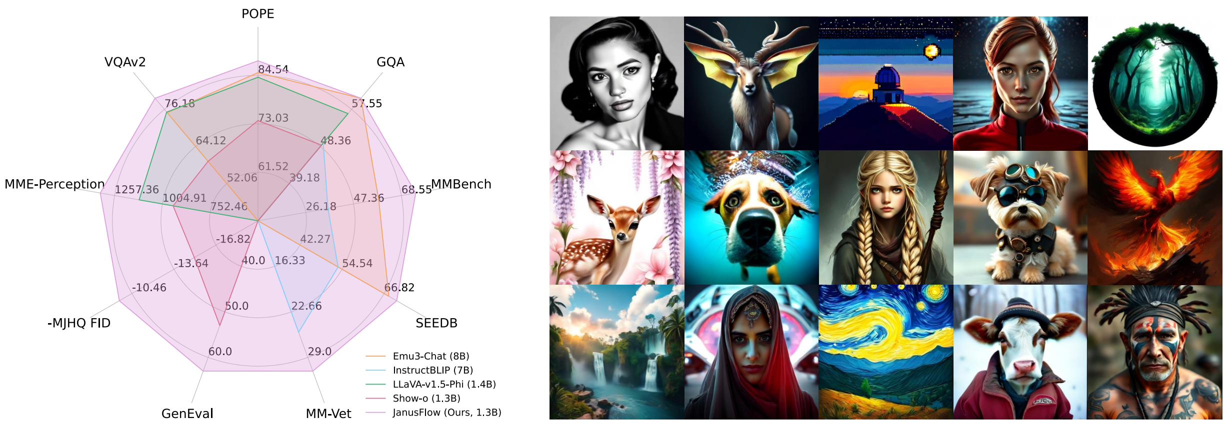
We present **JanusFlow**, a powerful framework that unifies image understanding and generation in a single model. JanusFlow introduces a minimalist architecture that integrates autoregressive language models with rectified flow, a state-of-the-art method in generative modeling. Our key finding demonstrates that rectified flow can be straightforwardly trained within the large language model framework, eliminating the need for complex architectural modifications. To further improve the performance of our unified model, we adopt two key strategies: (i) decoupling the understanding and generation encoders, and (ii) aligning their representations during unified training. Extensive experiments show that JanusFlow achieves comparable or superior performance to specialized models in their respective domains, while significantly outperforming existing unified approaches across standard benchmarks. This work represents a step toward more efficient and versatile vision-language models.

1. Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in learning diverse knowledge and generalizing to new scenarios [1, 7, 8, 68, 88]. Leveraging these capabilities, researchers have developed sophisticated models specialized in image comprehension [2, 15, 47, 49, 56, 57] and text-to-image generation [23, 71, 74, 77].

The field has recently shifted toward creating unified systems capable of handling both tasks simultaneously. One prominent direction involves utilizing pre-trained text-to-image models for high-quality generation while training LLMs to generate conditions for these models [19, 25–27, 84]. However, this approach introduces architectural complexity and potentially constrains the model's capabilities through maintaining separate LLM and generative components. Alternative approaches [85, 93, 95, 96, 103] propose training a single LLM for both tasks, typically incorporating either diffusion models [32, 80] or vector-quantized autoregressive models [22, 83].

Our approach builds upon recent breakthroughs in rectified flow models [3, 23, 55, 60, 61], which provide a simple framework for generative modeling while delivering exceptional empir-



(a) Benchmark Performances.

(b) Visual Generation Results.

Figure 1 | **Multimodal understanding and image generation with JanusFlow**. JanusFlow surpasses the state-of-the-art unified multimodal models and several task-specific understanding models on visual understanding benchmarks. It is also capable of generating high-quality images. The resolution of the images is 384×384.

ical performance [23, 36, 45]. Building on these advances, we propose **JanusFlow**, a powerful unified multimodal model that seamlessly integrates rectified flow with LLM architecture. Following a minimalist design principle, our architecture requires only a lightweight encoder and decoder to adapt the LLM for rectified flow operations. To optimize JanusFlow's performance, we implement two key strategies: First, we maintain separate vision encoders for understanding and generation tasks, preventing task interference and thus enhancing comprehension capabilities. Second, we align the intermediate representations between generation and understanding modules during training, strengthening semantic coherence in the generation process.

JanusFlow shows state-of-the-art performances in both multimodal comprehension and text-to-image generation compared to existing unified approaches, and even outperforms several specialized methods. Specifically, on text-to-image generation benchmarks, MJHQ FID-30k [48], GenEval [28] and DPG-Bench [34], JanusFlow achieves scores of 9.51, 0.63 and 80.09%, surpassing established text-to-image models including SDv1.5 [75] and SDXL [71]. In multimodal comprehension benchmarks, JanusFlow attains scores of 74.9, 70.5 and 60.3 on MMBench [62], SeedBench [46], and GQA [35], respectively, exceeding specialized models such as LLaVA-v1.5 [56] and Qwen-VL-Chat [4]. Notably, these results are achieved with a compact LLM architecture with only 1.3B parameters.

2. Related Work

Visual Generation with Flow-based Generative Models. Recent years have witnessed remarkable progress in visual generation through diffusion models [32, 80], leading to impressive models like [66, 71, 74–77]. Building on these advances, flow-based generative models [3, 55, 60] emerged as a simplified alternative framework. These approaches have recently enabled advanced visual generation models [23, 36] that achieve superior empirical performance with faster sampling. Our work demonstrates that rectified flow [59–61] can be effectively integrated into LLMs, creating unified models that excel in both understanding and generation tasks.

Unified Models For Understanding and Generation. The development of multimodal large language models (MLLMs) has enabled effective integration of text and visual information. Building upon powerful LLMs [7, 88, 89], recent MLLMs [2, 15, 49, 56, 57, 63] have demonstrated exceptional multimodal understanding capabilities. Current research increasingly focuses on architectures that can simultaneously handle visual understanding and generation tasks. One approach extends MLLMs with pre-trained diffusion models [19, 25–27, 84, 97]. However, these systems essentially utilize diffusion models as external tools, where the MLLM generates conditions for image generation without possessing direct generative capabilities. This separation often results in suboptimal performance compared to standalone diffusion models [25, 84]. Another line of work [85, 93, 95, 96, 103] aim to train a single LLM for both tasks. Many of these methods employ vector-quantization [22, 83] to convert images into discrete tokens, enabling unified autoregressive processing [85, 93]. While straightforward to implement, these approaches are inherently limited by their image tokenization quality.

Our work focuses on developing unified models that combine autoregressive capabilities with flow/diffusion models, leveraging their proven effectiveness in visual generation. Compared to similar approaches [96, 103], JanusFlow offers three key advantages: (i) a simple yet effective generation process using rectified flow, (ii) enhanced performance through decoupled vision encoders that resolve inter-task conflicts, and (iii) improved generation quality through representation alignment regularization, enabled by our decoupled encoder design.

3. JanusFlow

In this section, we introduce the architecture of JanusFlow and our training strategies.

3.1. Background

Multimodal LLMs. Given a dataset $\mathcal{D}$ containing discrete token sequences, each of which can be formulated as $x = (x_1, \cdots, x_\ell)$, large language models (LLMs) are trained to model the sequence distribution in an autoregressive manner,

$$\boxed{\log \mathrm{P}_{\theta_{LLM}}(x) = \sum_{i=0}^{\ell-1} \log \mathrm{P}_{\theta_{LLM}}(x_{i+1}|x_1, \dots, x_i),} \tag{1}$$

where θ_{LLM} denotes the parameters of the LLM and ℓ is the sequence length. After being trained on large-scale datasets, LLMs exhibit the ability to generalize across various tasks and follow diverse instructions [1, 8, 68]. To extend these models to handle visual inputs, LLMs are augmented with vision encoders [2, 56, 57]. For instance, LLaVA [57] integrates an LLM with a pre-trained CLIP [73] image encoder via a projection layer, transforming the extracted image features into a joint embedding space that the LLM can process as word embeddings. By leveraging large-scale multimodal datasets and increasingly powerful LLMs, this architecture has facilitated the development of advanced multimodal models capable of addressing a wide range of vision-language tasks [4, 47, 56, 63].

Rectified Flow. For a dataset $\mathcal{D}$ consisting of continuous d-dimensional data points $x = (x_1, \cdots, x_d)$ drawn from an unknown data distribution π_1, rectified flow [55, 60] models the data distribution by learning an ordinary differential equation (ODE) defined over time $t \in [0, 1]$:

$$\boxed{\frac{\mathrm{d}z_t}{\mathrm{d}t} = v_{\theta_{NN}}(z_t, t), \qquad z_0 \sim \pi_0,} \tag{2}$$

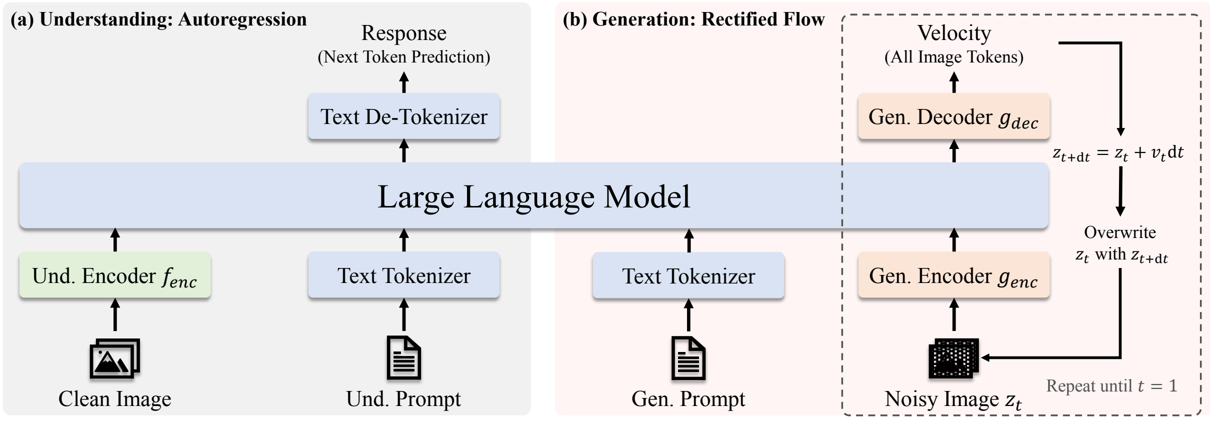


Figure 2 | **Architecture of the proposed JanusFlow.** For visual understanding, the LLM performs autoregressive next-token prediction to generate responses. For image generation, the LLM employs images with rectified flow. Starting from Gaussian noise at $t = 0$, the LLM iteratively updates z_t by predicting velocity vectors until reaching $t = 1$. We omit the VAE encoder, the skip connection leveraged in generation and the linear layer after f_{enc} for simplicity.

where θ_{NN} represents the parameters of the velocity neural network and π_0 is a simple distribution, typically standard Gaussian noise $\mathcal{N}(0, I)$. The network is trained by minimizing the Euclidean distance between the neural velocity and the directions of linear paths connecting random points from π_0 and π_1,

$$\min_{\theta} \mathbb{E}_{t\sim \mathrm{P}(t), z_0\sim\pi_0, x\sim\pi_1} \left[\left\| v_{\theta_{NN}}(z_t, t) - (x - z_0) \right\|^2 \right], \quad \text{where} \quad z_t = tx + (1-t)z_0. \tag{3}$$

Here, $\mathrm{P}(t)$ is a distribution over time $t \in [0, 1]$. When the network has sufficient capacity and the objective is perfectly minimized, the optimal velocity field $v_{\theta^*_{NN}}$ maps the elementary distribution π_0 to the true data distribution π_1. More precisely, the distribution of $z_1 = \int_0^1 v_{\theta^*_{NN}}(z_t, t)\mathrm{d}t$, with $z_0 \sim \pi_0$, follows π_1. Despite its conceptual simplicity, rectified flow has shown superior performance in various generative modeling tasks, including text-to-image generation [23], audio generation [40] and biological structure generation [38].

3.2. A Unified Framework for Multimodal Understanding and Generation

JanusFlow presents a unified framework designed to address both vision understanding and image generation tasks. Next we outline how JanusFlow handles these two tasks within a single LLM architecture.

Multimodal Understanding. In multimodal understanding tasks, the LLM processes an input sequence consisting of interleaved text and image data. The text is tokenized into discrete tokens, each of which is transformed into an embedding of dimension D_{emb}. For the images, an image encoder f_{enc} encodes each image x_{im} into a feature map of shape $H_{im} \times W_{im} \times D_{enc}$. This feature map is flattened and projected through a linear transformation layer into a sequence of embeddings with shape $H_{im}W_{im} \times D_{emb}$. H_{im} and W_{im} are determined by the image encoder. The text and image embeddings are concatenated to form the input sequence to the LLM, which then autoregressively predicts the next tokens based on the input sequence of embeddings. According to common practice [85, 93, 96], we add special token `|BOI|` before the image and `|EOI|` after the image to help the model locate the image embeddings in the sequence.

The fact that tokenization works for images still surprises me. It breaks so many assumptions about the image modality.

Image Generation. For image generation, our LLM takes a text sequence x^{con} as condition and generates a corresponding image using rectified flow. To improve computational efficiency, generation occurs in the latent space using a pre-trained SDXL-VAE [71].

The generation process begins by sampling Gaussian noise z_0 of shape $H_{latent} \times W_{latent} \times D_{latent}$ in the latent space, which is then processed by a generation encoder g_{enc} into a sequence of embeddings $H_{gen}W_{gen} \times D_{emb}$. This sequence is concatenated with a time embedding representing the current time step t ($t = 0$ at the beginning), resulting in a sequence of length $H_{gen}W_{gen} + 1$. Unlike previous approaches that employ various attention masking strategies [96, 103], we found that causal attention suffices, as our preliminary experiments showed no performance benefits from alternative masking schemes. The LLM's output corresponding to z_0 is transformed back into the latent space by a generation decoder g_{dec}, producing a velocity vector of shape $H_{latent} \times W_{latent} \times D_{latent}$. The state is updated by a standard Euler solver,

$$z_{t+\mathrm{d}t} = z_t + v(z_t, t)\mathrm{d}t, \tag{4}$$

where $\mathrm{d}t$ is a user-defined step size. We replace z_0 with $z_{\mathrm{d}t}$ on the input and iterate the process until we get z_1, which is then decoded into the final image by the VAE decoder. To enhance generation quality, we employ classifier-free guidance (CFG) when computing the velocity:

When in doubt, use CFG!

$$\boxed{v(z_t, t) = wv(z_t, t \mid x^{con}) + (1 - w)v(z_t, t \mid \varnothing),} \tag{5}$$

where $v(z_t, t \mid \varnothing)$ denotes the velocity inferred without text conditioning and $w \geqslant 1$ controls the magnitute of CFG. Empirically, increasing w yields higher semantic alignment [23, 61, 71, 75]. Analogous to multimodal understanding, we prepend the special token `|BOI|` to indicate the start of image generation in the sequence.

Decoupling Encoders for the Two Tasks. Previous approaches that unify autoregressive generation and diffusion models within a joint LLM training framework [96, 103] employ identical encoders (f_{enc} and g_{enc}) for both understanding and generation tasks. For instance, Zhou et al. [103] performs both tasks in the same VAE latent space using a shared U-Net or linear encoder, while Xie et al. [96] leverages MAGVIT-v2 [98] to encode image patches into discrete tokens for both tasks.

However, recent work on unified autoregressive models has shown this shared encoder design to be suboptimal [93], particularly in models that generate images through autoregression on vector-quantized tokens. Drawing from these insights, JanusFlow adopts a decoupled encoder design. Specifically, we employ a pre-trained SigLIP-Large-Patch/16 [102] model as f_{enc} to extract semantic continuous features for multimodal understanding, while using separate ConvNeXt blocks [92] initialized from scratch as g_{enc} and g_{dec} for generation, chosen for its effectiveness. Following established practices [5, 14, 90], we incorporate a long skip connection between g_{enc} and g_{dec}. Our controlled experiments in Sec. 4.5 demonstrate that this decoupled encoder design significantly improves the performance of our unified model. The complete architecture of JanusFlow is illustrated in Fig. 2.

3.3. Training Schemes

As illustrated in Fig. 3, we train our model in three sequential stages, detailed below.

Stage 1: Adaptation of Randomly Initialized Components. In the first stage, we focus on training only the randomly initialized components: the linear layers, generation encoder, and

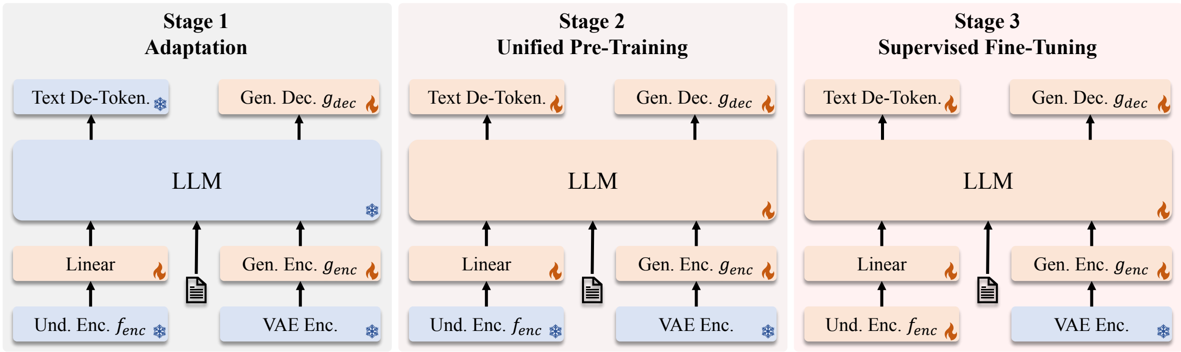


Figure 3 | **Three training stages of JanusFlow.** The trainable modules are marked with flame and the frozen modules are marked with snowflakes.

generation decoder. This stage serves to adapt these new modules to work effectively with the pre-trained LLM and SigLIP encoder, essentially functioning as an initialization phase for the newly introduced components.

Stage 2: Unified Pre-Training. Following the adaptation stage, we train the entire model except for the visual encoder, consistent with previous approaches [57, 63]. The training incorporates three data types: multimodal understanding, image generation, and text-only data. We initially allocate a higher proportion of multimodal understanding data to establish the model's understanding capabilities. Subsequently, we increase the ratio of image generation data to accommodate the convergence requirements of diffusion-based models [18, 70].

Stage 3: Supervised Fine-Tuning (SFT). In the final stage, we fine-tune the pre-trained model using instruction tuning data, which comprises dialogues, task-specific conversations, and high-quality text-conditioned image generation examples. During this stage, we also unfreeze the SigLIP encoder parameters [63, 87, 93]. This fine-tuning process enables the model to effectively respond to user instructions for both multimodal understanding and image generation tasks.

3.4. Training Objective

Training JanusFlow involves two types of data, multimodal understanding data and image generation data. Both types of data contain two parts: "condition" and "response". "Condition" refers to the prompting of the tasks (*e.g.*, text prompts in the task of generation and images in the task of understanding) while "response" refers to the corresponding responses of the two tasks. The data can be formatted as $x = (x^{con}, x^{res})$, where the superscript *con* denotes "condition" and *res* denotes "response". We denote the length of the whole sequence x as ℓ, the length of x^{con} as ℓ_{con} and the length of x^{res} as ℓ_{res}. We use θ to represent the collection of all the trainable parameters in JanusFlow, including the LLM, f_{enc}, g_{enc}, g_{dec} and the linear transformation layers.

Autoregression Objective. For mutimodal understanding tasks, x^{res} contains only text tokens. JanusFlow is trained using the maximum likelihood principle,

$$\boxed{\mathcal{L}_{AR}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_{und}}\left[\sum_{i=\ell_{con}}^{\ell-1} \log P_\theta\left(x_{i+1}|x_1,\ldots,x_i\right)\right],} \tag{6}$$

Table 1 | **Hyper-parameters of the proposed JanusFlow**. Data ratio denotes the proportion of multimodal understanding data, image generation data and text-only data. In the initial 10,000 steps of Stage 2, we apply a data ratio of 30 : 50 : 20 to boost the understanding ability.

	Stage 1	**Stage 2**	**Stage 3**
Learning Rate	1.0×10^{-4}	1×10^{-4}	2.0×10^{-5}
LR Scheduler	Constant	Constant	Constant
Weight Decay	0.0	0.0	0.0
Gradient Clip	1.0	1.0	1.0
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$)		
Warm-up Steps	2,000	0	1,000
Training Steps	10,000	380,000	26,000
Batch Size	512	512	256
Data Ratio	50 : 50 : 0	14 : 80 : 6	21 : 70 : 9

Marked these because this is ver diff from the settings we see in other MLLMs. I wonder if we replace AdamW with SOAP, how much can we juice this out even more

where the expectation is taken over all (x^{con}, x^{res}) pairs in our multimodal understanding dataset $\mathcal{D}_{und}$, computing loss only over tokens in x^{res}.

Rectified Flow Objective. For image generation tasks, x^{con} consists of text tokens and x^{res} is the corresponding image. JanusFlow is trained with the rectified flow objective,

$$\boxed{\mathcal{L}_{RF}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_{gen}, t \sim \mathrm{P}(t), z_0 \sim \mathcal{N}(0,I)} \left[||v_\theta(z_t, t \mid x^{con}) - (x^{res} - z_0)||^2 \right],} \tag{7}$$

where $z_t = tx^{res} + (1-t)z_0$. Following Stable Diffusion 3 [23], we set the time distribution $\mathrm{P}(t)$ to the logit-normal distribution. To enable CFG inference, we randomly drop 10% of the text prompts in training.

Representation Alignment Regularization. Recent work [99] has shown that aligning intermediate representations between diffusion transformers and semantic vision encoders enhances diffusion model generalization. Our decoupled vision encoder design enables efficient implementation of this alignment as a regularization term. Specifically, for generation tasks, we align features from the understanding encoder f_{enc} with the LLM's intermediate features,

$$\boxed{\mathcal{L}_{REPA}(\theta, \varphi) = -\mathbb{E}_{x \sim \mathcal{D}_{gen}} \left[\mathrm{sim}\left(\texttt{stop\_grad}(f_{enc}(x^{res})), h_\varphi(q_\theta(z_t)) \right) \right],} \tag{8}$$

where $q_\theta(z_t)$ denotes an intermediate LLM representation given input z_t, and h_φ is a small trainable MLP that projects $q_\theta(z_t)$ to dimension D_{enc}. The function $\mathrm{sim}(\cdot,\cdot)$ computes the mean of element-wise cosine similarity between embeddings. Before computing the loss, we reshape $h_\varphi(q_\theta(z_t))$ to $H_{gen} \times W_{gen} \times D_{enc}$. To simplify the implementation, we intentionally adjust the configuration of g_{enc} and g_{dec} to ensure $H_{gen} = H_{im}$ and $W_{gen} = W_{im}$. The gradient of $\mathcal{L}_{REPA}$ is not back-propagated through the understanding encoder. This alignment loss helps the LLM's internal feature space (given noisy input z_t) align with the understanding encoder's semantic feature space, thereby improving generation quality when producing images from new random noise and text conditions during inference.

Summary. All three objectives are applied across all training stages. Multimodal understanding tasks use $\mathcal{L}_{AR}$, while image generation tasks employ the combined loss $\mathcal{L}_{RF} + \mathcal{L}_{REPA}$. Detailed experimental settings are provided in Sec. 4.1.

4. Experiments

We conduct extensive experiments to evaluate the capabilities of JanusFlow in both multimodal understanding and generation tasks. First, we describe our experimental setup and implementation details. Then, we present results on standard benchmarks for multimodal understanding and image generation. Finally, we perform ablation studies to validate our key design choices.

4.1. Experiment Setup and Implementation Details

Our framework builds upon an enhanced version[1] of DeepSeek-LLM (1.3B) [7, 63]. The LLM consists of 24 transformer blocks and supports a sequence length of $4,096$. In our model, both understanding and generation exploits images of resolution 384.

For multimodal understanding, we leverage SigLIP-Large-Patch/16 [102] as f_{enc}. For image generation, we utilize the pre-trained SDXL-VAE [71] for its latent space. The generation encoder g_{enc} comprises a 2×2 patchify layer followed by two ConvNeXt [92] blocks and a linear layer. The generation decoder g_{dec} combines two ConvNeXt blocks, a pixel-shuffle layer to upsample the feature map, and a linear layer. Our SigLIP encoder contains $\sim$ 300M parameters. g_{enc} and g_{dec} are light-weight modules, containing $\sim$ 70M parameters in total. Table 1 details the hyperparameters for each training stage. In the alignment regularization, we use the LLM features after the 6th block as $q_\theta(z_t)$ and a three-layer MLP as h_φ. We employ an exponential moving average (EMA) with a ratio of 0.99 to ensure training stability.

The #params in encoder gives you an insight why I am bullish on small VLMs

For data preprocessing, we deal with understanding and generation data differently. For understanding tasks, we maintain all image information by resizing the long side to the target size and padding the image to squares. For generation tasks, we resize the short side to the target size and apply random square cropping to avoid padding artifacts. During training, multiple sequences are packed to form a single sequence of length $4,096$ for training efficiency. Our implementation is based on the HAI-LLM platform [31] using PyTorch [72]. Training was conducted on NVIDIA A100 GPUs, with each model requiring $\sim 1,600$ A100 GPU days.

4.2. Training Data Settings

We follow Janus [93] to construct the training data. The data configuration for each training stage is listed below.

Data for Stage 1 and Stage 2. The first two stages of our framework uses three types of data: multimodal understanding data, image generation data and text-only data.

1. **Multimodal Understanding Data.** This type of data contains several sub-categories: (a) Image caption data. We incorporate caption datasets from [20, 41, 50, 51, 53, 79] and generate additional captions for images from [16, 43] using open-source multimodal understanding models. The data follows template formats, *e.g.*, "<image>Generate the caption of this picture. <caption>". (b) Charts and tables. We directly adopt the chart and table data from the training data of DeepSeek-VL [63]. (c) Task data. ShareGPT4V [11] data is utilized to facilitate basic question-answering capabilities during pre-training,

[1]This version, trained on an expanded text corpus compared to the one in Janus [93], has been demonstrated to possess better performance on multiple-choice benchmarks (e.g., MMBench [62] and SEED Bench [46]). Our preliminary experiments suggest that it has minimal impact on the quality of visual generation.

Table 2 | **Performances on GenEval benchmark.** "Gen." denotes "generation" and "Unified" denotes unified understanding and generation models. Models using external pre-trained generative models are signed with †.

Type	Method	Params	Single Obj.	Two Obj.	Count.	Colors	Pos.	Color Attri.	Overall↑
Gen. Only	LlamaGen [83]	0.8B	0.71	0.34	0.21	0.58	0.07	0.04	0.32
	LDM [75]	1.4B	0.92	0.29	0.23	0.70	0.02	0.05	0.37
	SDv1.5 [75]	0.9B	0.97	0.38	0.35	0.76	0.04	0.06	0.43
	PixArt-α [9]	0.6B	0.98	0.50	0.44	0.80	0.08	0.07	0.48
	SDv2.1 [75]	0.9B	0.98	0.51	0.44	0.85	0.07	0.17	0.50
	DALL-E 2 [74]	6.5B	0.94	0.66	0.49	0.77	0.10	0.19	0.52
	Emu3-Gen [91]	8B	0.98	0.71	0.34	0.81	0.17	0.21	0.54
	SDXL [71]	2.6B	0.98	0.74	0.39	0.85	0.15	0.23	0.55
	IF-XL [17]	4.3B	0.97	0.74	0.66	0.81	0.13	0.35	0.61
	DALL-E 3 [6]	-	0.96	0.87	0.47	0.83	0.43	0.45	0.67
Unified	Chameleon [85]	34B	-	-	-	-	-	-	0.39
	LWM [58]	7B	0.93	0.41	0.46	0.79	0.09	0.15	0.47
	SEED-X† [27]	17B	0.97	0.58	0.26	0.80	0.19	0.14	0.49
	Show-o [96]	1.3B	0.95	0.52	0.49	0.82	0.11	0.28	0.53
	Janus [93]	1.3B	0.97	0.68	0.30	0.84	0.46	0.42	0.61
	JanusFlow (Ours)	1.3B	0.97	0.59	0.45	0.83	0.53	0.42	0.63

structured as "`<image><question><answer>`". (d) Interleaved text-image data. This sub-category is sourced from [42, 81].

2. **Image Generation Data.** Our image generation dataset combines high-quality images from [16, 21, 41, 43, 67, 69, 79, 82] and 2 million in-house data. We enhance them with machine-generated captions using multimodal understanding models. We filter the images in [16, 79] with aspect ratios and aesthetic scores, retaining approximately 20% of the original datasets. 25% of the data contains single-sentence captions. These kind of data assist the model to be able to process short prompts. All the data points are formatted as "`<prompt><image>`".
3. **Text-Only Data.** We directly use the text corpus of DeepSeek-LLM [7].

Data for Stage 3. The SFT stage also uses three types of data:

1. **Multimodal Instruction Data.** We leverage the instruction tuning datasets from [29, 33, 35, 47, 64, 78].
2. **Image Generation Data.** We reformat the high-quality text-image pairs from [16, 79, 82] into an instruction format: "`User:<user prompt>\n\n Assistant:<image>`".
3. **Text-Only Data.** We directly incorporate the text-only data from [47].

4.3. Evaluation Settings

Image Generation. We evaluate the generated images using both visual quality and semantic accuracy metrics. For visual quality assessment, we employ the Fréchet Inception Distance [30] (FID) metric and compute FID between 30,000 generated images and their corresponding reference images from the MJHQ dataset [48]. The FID computation follows the implementation from GigaGAN [39]. To evaluate semantic accuracy, we utilize two specialized frameworks: GenEval [28] and DPG-Bench [34]. These frameworks are designed to assess whether the generated images accurately contain the objects and relationships specified in the input prompts, providing a broad evaluation of the generation capabilities.

Arghh.. the FID again....

Table 3 | **Performances on DPG-Bench.** The methods in this table are all generation-specific models except our method.

Method	Global	Entity	Attribute	Relation	Other	Overall↑
SDv1.5 [75]	74.63	74.23	75.39	73.49	67.81	63.18
PixArt-α [9]	74.97	79.32	78.60	82.57	76.96	71.11
Lumina-Next [105]	82.82	88.65	86.44	80.53	81.82	74.63
SDXL [71]	83.27	82.43	80.91	86.76	80.41	74.65
Playground v2.5 [48]	83.06	82.59	81.20	84.08	83.50	75.47
Hunyuan-DiT [54]	84.59	80.59	88.01	74.36	86.41	78.87
PixArt-Σ [10]	86.89	82.89	88.94	86.59	87.68	80.54
Emu3-Gen [91]	85.21	86.68	86.84	90.22	83.15	80.60
JanusFlow (Ours)	87.03	87.31	87.39	89.79	88.10	80.09

Multimodal Understanding. We evaluate JanusFlow's multimodal understanding abilities across a diverse set of vision-language benchmarks for general understanding capabilities, including POPE [52], MME [24], MMBench [62], SEEDBench [46], VQAv2 [29], GQA [35], MM-Vet [100], and MMMU [101].

4.4. Quantitative Results

Image Generation Performances. We report the performances on GenEval, DPG-Bench and MJHQ FID-30k. In Tab. 2, we give comparisons on GenEval including the scores of all the sub-tasks and the overall score. JanusFlow achieves an overall score of 0.63, surpassing the previous unified framework and several generation specific models including SDXL [71] and DALL-E 2 [74]. In Tab. 3, We show results on DPG-Bench and the corresponding comparisons. It is noted that all the methods in Tab. 3 are generation-specific models except our model. The results on GenEval and DPG-Bench demonstrate the ability of instruction following of our model. We give the comparisons on MJHQ FID-30k in Tab. 4. The images which are sampled to calculate FID are generated with a CFG factor $w = 2$ and a number of sampling steps 30. We sweep the CFG factor and the sampling steps and provide the results in the appendix. Our method achieves the best performance among all the models with 1.3B LLM. The results prove that the rectified flow is able to improve the quality of generated images over autoregressive models such as Janus [93].

Table 4 | **Results of MJHQ FID-30k.** The models which have similar scales to our model are marked with blue background. JanusFlow achieves the best FID among 1.3B models.

Method	Params	FID↓
LWM [58]	7B	17.77
VILA-U 256 [95]	7B	12.81
VILA-U 384 [95]	7B	7.69
Show-o [96]	1.3B	15.18
Janus [93]	1.3B	10.10
JanusFlow (Ours)	1.3B	9.51

Multimodal Understanding Performances. We show comparisons of our method and other methods including understanding-specific models and unified understanding and generation models in Tab. 5. Our model reaches the best performances among all the models with similar number of parameters and even surpasses multiple understanding-specific methods with larger scales. Our results demonstrate that our method harmonizes autoregressive LLM and rectified flow, achieving satisfying performance in both understanding and generation.

Table 5 | **Comparison with other methods on multimodal understanding benchmarks**. "Und." denotes "understanding" and "Unified" denotes unified understanding and generation models. The models employing external pre-trained generative models are marked with †. The models with LLMs which have similar number of parameters to us are marked with blue background under the line of dashes.

Type	Model	LLM Params	POPE↑	MME-P↑	MMB_{dev} ↑	SEED↑	VQAv2_{test}↑	GQA↑	MMMU↑	MM-Vet↑
Und. Only	MobileVLM [12]	2.7B	84.9	1288.9	59.6	-	-	59.0	-	-
	MobileVLM-V2 [13]	2.7B	84.7	1440.5	63.2	-	-	61.1	-	-
	LLaVA-Phi [104]	2.7B	85.0	1335.1	59.8	-	71.4	-	-	28.9
	LLaVA [57]	7B	76.3	809.6	38.7	33.5	-	-	-	25.5
	LLaVA-v1.5 [56]	7B	85.9	1510.7	64.3	58.6	78.5	62.0	35.4	31.1
	InstructBLIP [15]	7B	-	-	36.0	53.4	-	49.2	-	26.2
	Qwen-VL-Chat [4]	7B	-	1487.5	60.6	58.2	78.2	57.5	-	-
	IDEFICS-9B [44]	8B	-	-	48.2	-	50.9	38.4	-	-
	Emu3-Chat [91]	8B	85.2	-	58.5	68.2	75.1	60.3	31.6	-
	InstructBLIP [15]	13B	78.9	1212.8	-	-	-	49.5	-	25.6
	LLaVA-v1.5-Phi-1.5 [96]	1.3B	84.1	1128.0	-	-	75.3	56.5	30.7	-
	MobileVLM [12]	1.4B	84.5	1196.2	53.2	-	-	56.1	-	-
	MobileVLM-V2 [13]	1.4B	84.3	1302.8	57.7	-	-	59.3	-	-
Unified	Gemini-Nano-1 [86]	1.8B	-	-	-	-	62.7	-	26.3	-
	LWM [58]	7B	75.2	-	-	-	55.8	44.8	-	9.6
	VILA-U [95]	7B	85.8	1401.8	-	59.0	79.4	60.8	-	33.5
	Chameleon [85]	7B	-	-	-	-	-	-	22.4	8.3
	DreamLLM† [19]	7B	-	-	-	-	72.9	-	-	36.6
	LaVIT† [37]	7B	-	-	-	-	66.0	46.8	-	-
	Emu† [84]	13B	-	-	-	-	52.0	-	-	-
	NExT-GPT† [94]	13B	-	-	-	-	66.7	-	-	-
	Show-o [96]	1.3B	73.8	948.4	-	-	59.3	48.7	25.1	-
	Janus [93]	1.3B	87.0	1338.0	69.4	63.7	77.3	59.1	30.5	34.3
	JanusFlow (Ours)	1.3B	88.0	1333.1	74.9	70.5	79.8	60.3	29.3	30.9

4.5. Ablation Studies

We conduct comprehensive ablation studies to validate the effectiveness of our key design choices. For computational efficiency, all ablation experiments are performed on 256×256 resolution images[2]. All models are trained on our unified pre-training dataset for 50,000 iterations, except for the understanding-only and generation-only variants, which are trained for proportionally fewer iterations based on their respective data ratios in the pre-training phase. The quantitative results of these ablation studies are presented in Tab. 6.

Impact of Representation Alignment. The comparison between Exp. A and F demonstrates the significant benefits of incorporating representation alignment regularization [99] during training. Specifically, models trained with representation alignment show notably lower FID scores on MJHQ dataset and higher CLIP scores, indicating simultaneous improvements in both image quality and semantic alignment. Importantly, our architecture differs from previous studies [65, 70] examined in [99] due to our incorporation of LLM and an additional skip connection between g_{enc} and g_{dec}. The effectiveness of representation alignment in our modified architecture suggests its broad applicability and generalization capability across different network structures.

Impact of Decoupling Visual Encoders. e efficacy of using powerful pre-trained visual encoders

[2]The understanding encoders in the 256×256-based ablation studies is also SigLIP-Large-Patch/16 which is pre-trained on 256×256 images.

Table 6 | **Ablation studies.** The weights of the modules with † are frozen during training. “Exp.” denotes “experiment”. “FID” in this table is MJHQ FID-10k with CFG factor $w = 7.5$ and 30 steps. “CLIP” denotes CLIP similarity with the backbone of CLIP-ViT-Large-Patch/14. Exp. F is the final configuration for training JanusFlow.

Exp. ID	Model Setting				Train. Iter.	Evaluation Benchmarks				
	REPA	Und. Modules	Gen. Modules	Type		POPE↑	VQAv2$_{val}$ ↑	GQA↑	FID↓	CLIP ↑
A	×	SigLIP	VAE†+ConvNeXt	Unified	50,000	82.40	69.62	54.43	19.84	24.94
B	✓	Shared VAE†+ConvNeXt		Unified	50,000	78.13	53.94	44.04	18.05	26.38
C	✓	VAE+ConvNeXt	VAE†+ConvNeXt	Unified	50,000	75.30	55.41	44.44	17.53	26.32
D	✓	SigLIP	-	Und. Only	13,000	85.03	69.10	54.23	-	-
E	✓	-	VAE†+ConvNeXt	Gen. Only	37,000	-	-	-	16.69	26.89
F	✓	SigLIP	VAE†+ConvNeXt	Unified	50,000	84.73	69.20	54.83	17.61	26.40

A corgi’s head depicted as an explosion of a nebula, with vibrant cosmic colors like deep purples, blues, and pinks swirling around. The corgi’s fur blends seamlessly into the nebula, with stars and galaxies forming the texture of its fur. Bright bursts of light emanate from its eyes, and faint constellations can be seen in the background, giving the image a surreal, otherworldly feel.

Beautiful surreal symbolism the mesmerizing vision of a Cleopatra Queen of Egypt, mesmerizing brown eyes, black hair and ethereal features, radiating celestial aura, super high definition, true lifelike color, perfect exposure, razor sharp focus, golden ratio, soft reflections, bokeh effect, fine art photography, cinematic compositing, authentic, professional.

A lone figure in dark robes ascends worn stone steps toward a glowing light in an ancient temple entrance. Ornate arches, lush greenery, and intricate carvings adorn the scene, evoking a mystical, high-fantasy atmosphere reminiscent of works by artists like Randy Vargas, with cinematic lighting and epic storytelling.

Figure 4 | **Image generation results of JanusFlow.** Our model can generate high-quality images that are semantically consistent with text prompts.

in multimodal understanding. The comparison among Exp. B, C, and F demonstrates the advantages of using separate visual encoders for understanding and generation tasks. In Exp. B, following a design similar to Transfusion [103], we implement shared ConvNeXt blocks in the SDXL-VAE latent space for both understanding and generation encoders. Exp. C employs separate encoders with identical architectures and initialization parameters, but trained independently. The performance differences between these configurations validate the necessity of decoupled visual encoders in improving our unified model’s capabilities. Moreover, the superior results in Exp. C and F highlight the benefits of leveraging pre-trained semantic visual encoders for multimodal understanding tasks.

Fair Comparison with Understanding / Generation-Only Models. To establish meaningful benchmarks, we evaluate task-specific models trained under identical conditions - using the same pre-training dataset, infrastructure, and hyperparameters. Exp. D and E represent these specialized models, trained with data volumes matching the unified models in Tab. 6. The minimal performance gap between Exp. F and these task-specific baselines demonstrates that our unified framework successfully integrates understanding and generation capabilities

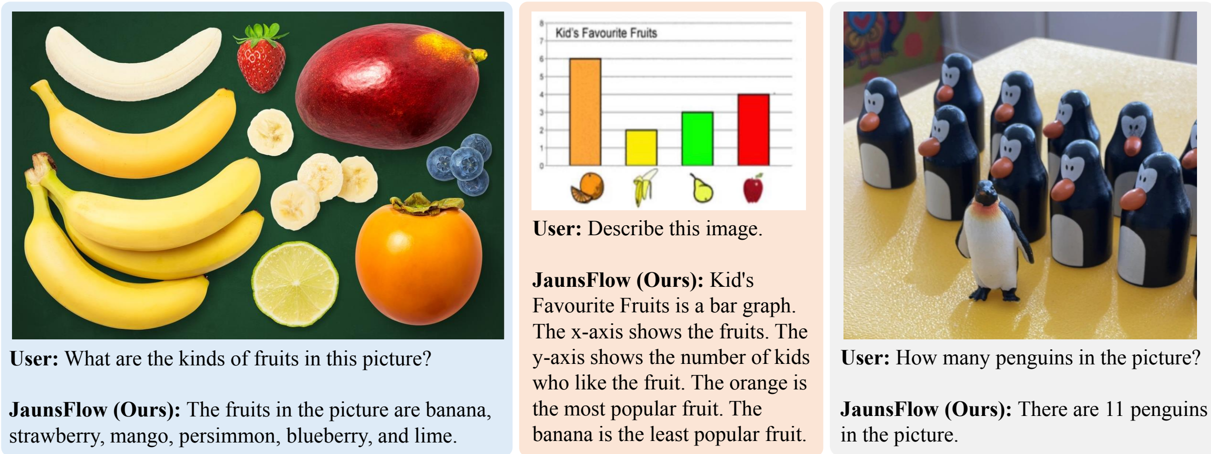


Figure 5 | **Visual Understanding with JanusFlow.** Our model effectively handles various visual understanding tasks, such as question answering, plot interpretation and object counting.

without significant compromise in either task's performance.

4.6. Qualitative Results

We present qualitative evaluations of our method for both image generation and understanding tasks. Fig. 1(b) and Fig. 4 showcases the image generation capabilities of JanusFlow. These results demonstrate both the high visual quality of our generated images and our framework's ability to faithfully follow diverse instructions. For multimodal understanding, Fig. 5 presents example conversations that show our model's understanding capabilities across various scenarios. These interactions demonstrate the model's ability to understand and reason about visual content in natural language dialogues. Additional qualitative examples showcasing the versatility and effectiveness of JanusFlow are provided in the appendix.

5. Conclusion

We present JanusFlow, a unified framework that successfully harmonizes autoregressive and rectified flow models for multimodal understanding and generation tasks. Our extensive experiments demonstrate that this unification achieves comparable performance to task-specific models. The successful integration of these fundamentally different model architectures not only addresses current challenges in multimodal learning but also opens new possibilities for future research in training unified models.

Summary

MultiModal Understanding

In multimodal understanding tasks, the LLM processes an input sequence consisting of interleaved text and image data.

- Text is tokenized into discrete tokens. Each token is transformed into an embedding of dimension $Demb$.
- An image encoder f_enc encodes images into feature maps of shape $Him \times Wim \times Denc$. The authors use a pre-trained SigLIP-Large-Patch/16 model as the image encoder (f_enc).
- The image feature map is flattened and projected through a linear transformation layer into a sequence of embeddings with shape $(H\_im\ W\_im) \times Demb$.
- The text and image embeddings are concatenated to form the input sequence to the LLM. Special tokens |BOI| and |EOI| are inserted before and after the image tokens to help the model locate the image embeddings in the sequence.
- Based on the above sequence of input embeddings, the LLM predicts the next token autoregressively.

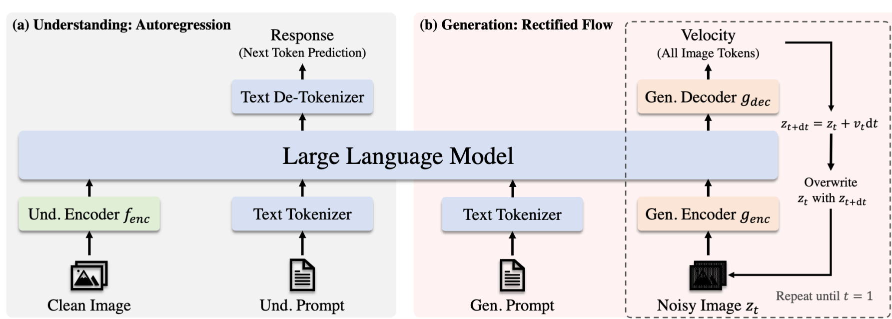


Figure 2 | **Architecture of the proposed JanusFlow.** For visual understanding, the LLM performs autoregressive next-token prediction to generate responses. For image generation, the LLM employs images with rectified flow. Starting from Gaussian noise at $t = 0$, the LLM iteratively updates z_t by predicting velocity vectors until reaching $t = 1$. We omit the VAE encoder, the skip connection leveraged in generation and the linear layer after f_{enc} for simplicity.

Image Generation

The same LLM used for multimodal understanding employs rectified flow for image generation.

- Generation occurs in the latent space using a pre-trained SDXL-VAE.
- The LLM takes a text sequence x_con as a condition and generates a corresponding image using rectified flow.
- Start by sampling Gaussian noise z0 of shape $H\_latent \times W\_latent \times D\_latent$ in the latent space.
- A generation encoder g_enc transforms the above image(noise in the beginning) into a sequence of embeddings $(H\_gen\ W\_gen) \times Demb$. The authors use ConvNext blocks initialized from scratch in the generation encoder here.
- The above sequence is then concatenated with a time embedding representing the current time step t ($t = 0$ at the beginning), resulting in a sequence of length $(H\_gen \cdot W\_gen + 1)$. Special token |BOI| is prepended to indicate the start of image generation in the sequence.
- The LLM predicts the next token autoregressively. The output is then transformed to a velocity vector of shape $H\_latent \times W\_latent \times D\_latent$ by a generation decoder g_dec. The authors use ConvNext blocks initialized from scratch in the generation decoder.
- The transformed output state is then updated by a standard Euler solver: $z(t+dt) = z\_t + v(z\_t, t)dt$. The step size dt is defined by the user.

- Use z_dt as the input, replacing z0 in the above steps iteratively till we obtain z1.
- To improve the image generation quality, the authors use the good old classifier-free guidance (CFG). $w \geqslant 1$ controls the magnitude of CFG, and increasing w yields higher semantic alignment.

Pay attention to the details here. Not only do the authors employ separate image encoders for understanding and generation, but they also use different kinds of models, SigLIP for understanding and ConvNext for generation.

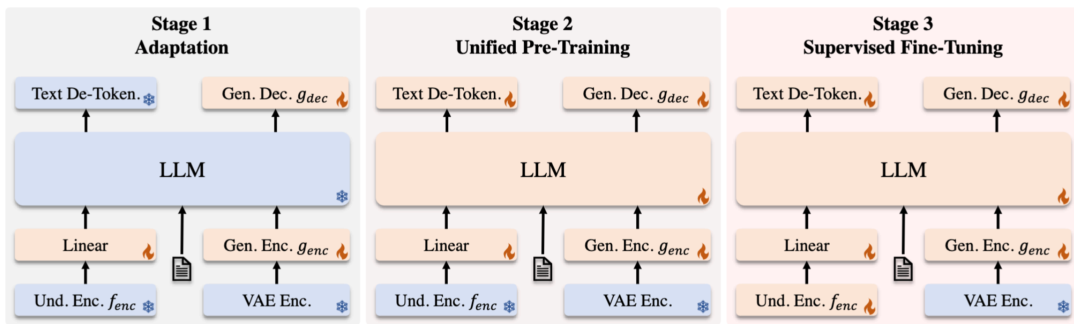


Figure 3 | **Three training stages of JanusFlow.** The trainable modules are marked with flame and the frozen modules are marked with snowflakes.

Training Schemes

The authors adopt a three-stage sequential training:

1. Adaptation: Randomly initialized components such as the linear layers, generation encoder, and generation decoder are trained in this stage.
2. Unified pre-training: Once the adaptation phase is complete, the authors train the entire model except for the visual encoder. The training incorporates three data types; multimodal understanding, image generation, and text-only data. At the start, the data mixture contains more multimodal understanding data, and the authors then increase the ratio of image generation data gradually.
3. Supervised Fine-tuning (SFT): At this stage, the SigLIP image encoder is also unfrozen. The model is fine-tuned using instruction-tuning data, which comprises dialogues, task-specific conversations, and high-quality text-conditioned image generation examples.

This training scheme reminds me of the old transfer learning followed by the layer-wise fine-tuning paradigm widely used in the peak CNN era of 2014-18.

The first two stages use three types of data: multimodal understanding data, image generation data, and text-only data. The multimodal understanding data contains image-caption pairs, charts and tables, and task data (ShareGPT4V). The image generation data comes from different sources like LAION-Aesthetics, SAM, etc., and 2 million in-house samples. All samples go through a filtration process to ensure only high quality. For text-only data, the authors use the text corpus of DeepSeek-LLM.

Training Objective

Training JanusFlow involves two types of data, multimodal understanding data and image generation data. Both types of data contain two parts: "condition" and "response". The data is formatted in pairs as $x = (xcon, xres)$, where the superscript con denotes "condition" and res denotes "response". The sequence length of x is denoted by l, while the sequence lengths of xcon and xres are denoted by lcon and lres respectively. θ denotes all the trainable parameters.

1. Autoregression Objective: For multimodal understanding tasks, $xres$ contains only text tokens, and the model is trained using the maximum likelihood principle where the loss is calculated over tokens in xres.
2. Rectified Flow Objective: For image generation tasks, $xcon$ consists of text tokens and $xres$ is the corresponding image, and the model is trained with the rectified flow objective. 10% of the text prompt is dropped during training to enable CFG.
3. Representation Alignment Regularization: For generation tasks, features from the understanding encoder $fenc$ are aligned with the intermediate features of the LLM as shown below. The function sim(...) computes the mean of element-wise cosine similarity between embeddings. This alignment loss helps the LLM's internal feature space (given noisy input zt) align with the understanding encoder's semantic feature space, thereby improving generation quality when producing images from new random noise and text conditions during inference.

All three objectives are applied across all training stages. Multimodal understanding tasks use L(AR), while image generation tasks employ the combined loss L(RF) + L($REPA$).

$$\mathcal{L}_{AR}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_{und}}\left[\sum_{i=\ell_{con}}^{\ell-1} \log \mathrm{P}_{\theta}\left(x_{i+1}|x_1,\ldots,x_i\right)\right].$$

$$\mathcal{L}_{RF}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{gen},t\sim\mathrm{P}(t),z_0\sim\mathcal{N}(0,I)}\left[||v_\theta(z_t,t \mid x^{con}) - (x^{res} - z_0)||^2\right]$$

$$\mathcal{L}_{REPA}(\theta,\varphi) = -\mathbb{E}_{x\sim\mathcal{D}_{gen}}\left[\mathrm{sim}\left(\texttt{stop\_grad}(f_{enc}(x^{res})), h_\varphi(q_\theta(z_t))\right)\right]$$

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2022.

[3] M. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In Proc. Int'l Conf. Learning Representations, 2023.

[4] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966, 2023.

[5] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu. All are worth words: A ViT backbone for diffusion models. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2023.

[6] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, et al. Improving image generation with better captions. Computer Science, 2023.

[7] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. DeepSeek LLM: Scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954, 2024.

[8] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv preprint arXiv:2303.12712, 2023.

[9] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. PixArt-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv preprint arXiv:2310.00426, 2023.

[10] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. PixArt-Sigma: Weak-to-strong training of diffusion transformer for 4K text-to-image generation. arXiv preprint arXiv:2403.04692, 2024.

[11] L. Chen, J. Li, X. Dong, P. Zhang, C. He, J. Wang, F. Zhao, and D. Lin. ShareGPT4V: Improving large multi-modal models with better captions. arXiv preprint arXiv:2311.12793, 2023.

[12] X. Chu, L. Qiao, X. Lin, S. Xu, Y. Yang, Y. Hu, F. Wei, X. Zhang, B. Zhang, X. Wei, et al. MobileVLM: A fast, reproducible and strong vision language assistant for mobile devices. arXiv preprint arXiv:2312.16886, 2023.

[13] X. Chu, L. Qiao, X. Zhang, S. Xu, F. Wei, Y. Yang, X. Sun, Y. Hu, X. Lin, B. Zhang, et al. MobileVLM V2: Faster and stronger baseline for vision language model. arXiv preprint arXiv:2402.03766, 2024.

[14] K. Crowson, S. A. Baumann, A. Birch, T. M. Abraham, D. Z. Kaplan, and E. Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In Proc. Int'l Conf. Machine Learning, 2024.

[15] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2023.

[16] dclure. LAION-Aesthetics-UMAP, 2022. URL https://huggingface.co/datasets/dclure/laion-aesthetics-12m-umap.

[17] DeepFloyd. DeepFloyd IF, 2023. URL https://huggingface.co/DeepFloyd/IF-I-XL-v1.0.

[18] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2021.

[19] R. Dong, C. Han, Y. Peng, Z. Qi, Z. Ge, J. Yang, L. Zhao, J. Sun, H. Zhou, H. Wei, et al. DreamLLM: Synergistic multimodal comprehension and creation. In Proc. Int'l Conf. Learning Representations, 2024.

[20] echo840. Detailed caption, 2023. URL https://huggingface.co/datasets/echo840/Detailed_Caption.

[21] B. Egan, A. Redden, XWAVE, and SilentAntagonist. DALLE-3 1 million+ high quality captions, 2024. URL https://huggingface.co/datasets/ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions.

[22] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2021.

[23] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In Proc. Int'l Conf. Machine Learning, 2024.

[24] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, Y. Wu, and R. Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394, 2024.

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

[26] Y. Ge, S. Zhao, Z. Zeng, Y. Ge, C. Li, X. Wang, and Y. Shan. Making LLaMA SEE and draw with SEED tokenizer. arXiv preprint arXiv:2310.01218, 2023.

[27] Y. Ge, S. Zhao, J. Zhu, Y. Ge, K. Yi, L. Song, C. Li, X. Ding, and Y. Shan. SEED-X: Multimodal models with unified multi-granularity comprehension and generation. arXiv preprint arXiv:2404.14396, 2024.

[28] D. Ghosh, H. Hajishirzi, and L. Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2024.

[29] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in VQA matter: Elevating the role of image understanding in visual question answering. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2017.

[30] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. Proc. Annu. Conf. Neural Inf. Process. Systems, 2017.

[31] High-flyer. HAI-LLM: Efficient and lightweight training tool for large models, 2023. URL https://www.high-flyer.cn/en/blog/hai-llm.

[32] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2020.

[33] Y.-C. Hsiao, F. Zubach, G. Baechler, V. Carbune, J. Lin, M. Wang, S. Sunkara, Y. Zhu, and J. Chen. ScreenQA: Large-scale question-answer pairs over mobile app screenshots. arXiv preprint arXiv:2209.08199, 2022.

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

[35] D. A. Hudson and C. D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2019.

[36] Y. Jin, Z. Sun, N. Li, K. Xu, H. Jiang, N. Zhuang, Q. Huang, Y. Song, Y. Mu, and Z. Lin. Pyramidal flow matching for efficient video generative modeling. arXiv preprint arXiv:2410.05954, 2024.

[37] Y. Jin, K. Xu, L. Chen, C. Liao, J. Tan, Q. Huang, C. Bin, C. Song, D. ZHANG, W. Ou, et al. Unified language-vision pretraining in llm with dynamic discrete visual tokenization. In Proc. Int'l Conf. Learning Representations, 2024.

[38] B. Jing, B. Berger, and T. Jaakkola. AlphaFold meets flow matching for generating protein ensembles. In Proc. Int'l Conf. Machine Learning, 2024.

[39] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park. Scaling up GANs for text-to-image synthesis. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2023.

[40] S. Kim, K. Shih, J. F. Santos, E. Bakhturina, M. Desta, R. Valle, S. Yoon, B. Catanzaro, et al. P-Flow: a fast and data-efficient zero-shot tts through speech prompting. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2024.

[41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In Proc. IEEE Int. Conf. Comput. Vision, 2023.

[42] M. Koupaee and W. Y. Wang. WikiHow: A large scale text summarization dataset. arXiv preprint arXiv:1810.09305, 2018.

[43] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. Int'l Journal of Computer Vision, 2020.

[44] H. Laurençon, D. van Strien, S. Bekman, L. Tronchon, L. Saulnier, T. Wang, S. Karamcheti, A. Singh, G. Pistilli, Y. Jernite, et al. Introducing IDEFICS: An open reproduction of state-of-the-art visual language model, 2023, 2023. URL https://huggingface.co/blog/idefics.

[45] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, et al. VoiceBox: Text-guided multilingual universal speech generation at scale. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2024.

[46] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan. SEED-Bench: Benchmarking multimodal llms with generative comprehension. arXiv preprint arXiv:2307.16125, 2023.

[47] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, Y. Li, Z. Liu, and C. Li. LLaVA-OneVision: Easy visual task transfer. arXiv preprint arXiv:2408.03326, 2024.

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

[49] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proc. Int'l Conf. Machine Learning, 2023.

[50] L. Li, Y. Wang, R. Xu, P. Wang, X. Feng, L. Kong, and Q. Liu. Multimodal arXiv: A dataset for improving scientific comprehension of large vision-language models. In Annual Meeting of the Association for Computational Linguistics, 2024.

[51] X. Li, F. Zhang, H. Diao, Y. Wang, X. Wang, and L.-Y. Duan. DenseFusion-1M: Merging vision experts for comprehensive multimodal perception. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2024.

[52] Y. Li, Y. Du, K. Zhou, J. Wang, X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models. In Proc. Conf. on Empirical Methods in Natural Language Process., 2023.

[53] Z. Li, X. Yang, K. Choi, W. Zhu, R. Hsieh, H. Kim, J. H. Lim, S. Ji, B. Lee, X. Yan, et al. MMSci: A multimodal multi-discipline dataset for phd-level scientific comprehension. In AI for Accelerated Materials Design, 2024.

[54] Z. Li, J. Zhang, Q. Lin, J. Xiong, Y. Long, X. Deng, Y. Zhang, X. Liu, M. Huang, Z. Xiao, et al. Hunyuan-DiT: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. arXiv preprint arXiv:2405.08748, 2024.

[55] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In Proc. Int'l Conf. Learning Representations, 2023.

[56] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2024.

[57] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2024.

[58] H. Liu, W. Yan, M. Zaharia, and P. Abbeel. World model on million-length video and language with ringattention. arXiv preprint arXiv:2402.08268, 2024.

[59] Q. Liu. Rectified flow: A marginal preserving approach to optimal transport. arXiv preprint arXiv:2209.14577, 2022.

[60] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In Proc. Int'l Conf. Learning Representations, 2023.

[61] X. Liu, X. Zhang, J. Ma, J. Peng, et al. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In Proc. Int'l Conf. Learning Representations, 2024.

[62] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. MMBench: Is your multi-modal model an all-around player? In Proc. European Conf. Computer Vision, 2024.

[63] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang, et al. DeepSeek-VL: towards real-world vision-language understanding. arXiv preprint arXiv:2403.05525, 2024.

[64] P. Lu, L. Qiu, J. Chen, T. Xia, Y. Zhao, W. Zhang, Z. Yu, X. Liang, and S.-C. Zhu. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2021.

[65] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. arXiv preprint arXiv:2401.08740, 2024.

[66] Y. Ma, H. Yang, W. Wang, J. Fu, and J. Liu. Unified multi-modal latent diffusion for joint subject and text conditional image generation. arXiv preprint arXiv:2303.09319, 2023.

[67] madebyollin. Megelith-10M, 2024. URL https://huggingface.co/datasets/madebyollin/megalith-10m.

[68] B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.

[69] mehdidc. YFCC-15M, 2024. URL https://huggingface.co/datasets/mehdidc/yfcc15m.

[70] W. Peebles and S. Xie. Scalable diffusion models with transformers. In Proc. IEEE Int. Conf. Comput. Vision, 2023.

[71] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In Proc. Int'l Conf. Learning Representations, 2024.

[72] PyTorch-Contributors. PyTorch, 2024. URL https://pytorch.org.

[73] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In Proc. Int'l Conf. Machine Learning, 2021.

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

[75] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2022.

[76] L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo. MM-Diffusion: Learning multi-modal diffusion models for joint audio and video generation. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2022.

[77] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2022.

[78] S. Shah, A. Mishra, N. Yadati, and P. P. Talukdar. KVQA: Knowledge-aware visual question answering. In Proc. AAAI Conf. on Artificial Intelligence, 2019.

[79] V. Singla, K. Yue, S. Paul, R. Shirkavand, M. Jayawardhana, A. Ganjdanesh, H. Huang, A. Bhatele, G. Somepalli, and T. Goldstein. From pixels to prose: A large dataset of dense image captions. arXiv preprint arXiv:2406.10328, 2024.

[80] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In Proc. Int'l Conf. Learning Representations, 2021.

[81] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, and M. Najork. WIT: Wikipedia-based image text dataset for multimodal multilingual machine learning. In Proc. ACM SIGIR Conf. Research and Develop. in Info. Retrieval, 2021.

[82] K. Sun, J. Pan, Y. Ge, H. Li, H. Duan, X. Wu, R. Zhang, A. Zhou, Z. Qin, Y. Wang, et al. JourneyDB: A benchmark for generative image understanding. In Proc. Annu. Conf. Neural Inf. Process. Systems, 2024.

[83] P. Sun, Y. Jiang, S. Chen, S. Zhang, B. Peng, P. Luo, and Z. Yuan. Autoregressive model beats diffusion: LLaMA for scalable image generation. arXiv preprint arXiv:2406.06525, 2024.

[84] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang. Generative pretraining in multimodality. In Proc. Int'l Conf. Learning Representations, 2024.

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

[86] G. Team. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

[87] S. Tong, E. Brown, P. Wu, S. Woo, M. Middepogu, S. C. Akula, J. Yang, S. Yang, A. Iyer, X. Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. arXiv preprint arXiv:2406.16860, 2024.

[88] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

[89] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. LLaMA 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

[90] C. N. Vasconcelos, A. Rashwan, A. Waters, T. Walker, K. Xu, J. Yan, R. Qian, Y. Li, S. LUO, Y. Onoe, et al. Greedy growing enables high-resolution pixel-based diffusion models. Transactions on Machine Learning Research, 2024.

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

[92] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie. ConvNeXt v2: Co-designing and scaling ConvNets with masked autoencoders. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2023.

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

[94] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua. NExT-GPT: Any-to-any multimodal LLM. In Proc. Int'l Conf. Machine Learning, 2024.

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

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

[97] H. Ye, D.-A. Huang, Y. Lu, Z. Yu, W. Ping, A. Tao, J. Kautz, S. Han, D. Xu, P. Molchanov, et al. X-VILA: Cross-modality alignment for large language model. arXiv preprint arXiv:2405.19335, 2024.

[98] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, A. G. Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In Proc. Int'l Conf. Learning Representations, 2024.

[99] S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940, 2024.

[100] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In Proc. Int'l Conf. Machine Learning, 2024.

[101] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2024.

[102] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In Proc. IEEE Int. Conf. Comput. Vision, 2023.

[103] C. Zhou, L. Yu, A. Babu, K. Tirumala, M. Yasunaga, L. Shamis, J. Kahn, X. Ma, L. Zettlemoyer, and O. Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. arXiv preprint arXiv:2408.11039, 2024.

[104] Y. Zhu, M. Zhu, N. Liu, Z. Ou, X. Mou, and J. Tang. LLaVA-Phi: Efficient multi-modal assistant with small language model. arXiv preprint arXiv:2401.02330, 2024.

[105] L. Zhuo, R. Du, H. Xiao, Y. Li, D. Liu, R. Huang, W. Liu, L. Zhao, F.-Y. Wang, Z. Ma, et al. Lumina-Next: Making Lumina-T2X stronger and faster with Next-DiT. arXiv preprint arXiv:2406.18583, 2024.

Appendix

A. Performance Analysis of 256 Resolution Model

We trained our model at two resolutions: 256×256 and 384×384. The main paper presents results from the 384×384 model as our primary results. Here, we provide a comprehensive evaluation of the 256×256 model's performance. The visual understanding performances are presented in Tab. 1. The generation capabilities are evaluated using GenEval [28], DPG-Benchmark [34], and MJHQ FID-30k [48], with results shown in Tab. 2 and 3.

Table 1 | Results on visual understanding tasks.

Model	LLM Params	POPE↑	MME-P↑	MMB_{dev} ↑	SEED↑	VQAv2_{test}↑	GQA↑	MM-Vet↑
JanusFlow 256	1.3B	85.3	1203.0	71.9	67.6	76.3	58.4	27.4
JanusFlow 384	1.3B	88.0	1333.1	74.9	70.5	79.8	60.3	30.9

Table 2 | Results on GenEval [28].

Method	LLM Params	Single Obj.	Two Obj.	Count.	Colors	Pos.	Color Attri.	Overall↑
JanusFlow 256	1.3B	0.98	0.73	0.54	0.83	0.63	0.53	0.70
JanusFlow 384	1.3B	0.97	0.59	0.45	0.83	0.53	0.42	0.63

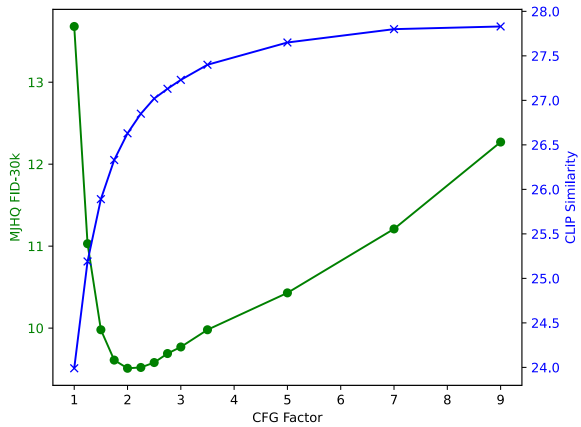
Table 3 | Results on DPG-Bench [34] and MJHQ FID-30k [48].

Method	DPG-Bench↑						MJHQ FID-30k↓
	Global	Entity	Attribute	Relation	Other	Overall	
JanusFlow 256	91.20	88.83	88.00	87.60	89.53	81.23	12.70
JanusFlow 384	87.03	87.31	87.39	89.79	88.10	80.09	9.51

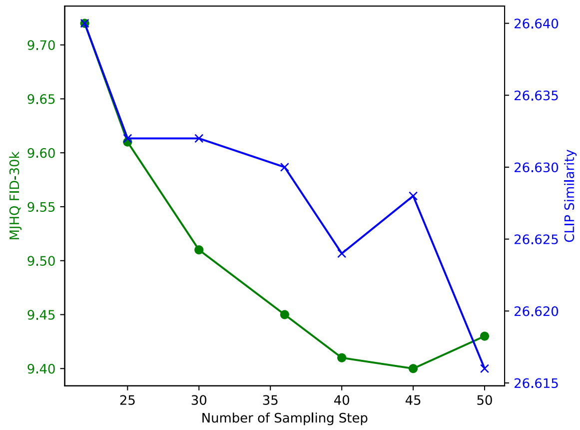
As expected, the 256×256 model shows slightly lower performance compared to the 384×384 model on visual understanding metrics due to its reduced resolution. Interestingly, however, the 256×256 model outperforms its higher-resolution counterpart on GenEval and DPG-Bench - benchmarks specifically designed to evaluate instruction following capabilities and semantic accuracy. This superior performance on semantic tasks can be attributed to the model's better control over lower-resolution images, where reduced visual complexity allows for more precise semantic manipulation.

B. Analysis of CFG Factor and Sampling Steps

We investigate the impact of two key generation parameters: the Classifier-Free Guidance (CFG) factor and the number of sampling steps. While our main results use $w = 2$ for CFG and 30 sampling steps to calculate FID, here we present a comprehensive analysis of these hyperparameters. Fig. 1(a) shows the effect of varying CFG factors while maintaining 30 sampling steps. The results reveal an optimal CFG value for FID scores, while CLIP [73] similarity continues to improve with increasing CFG values, consistent with findings from previous work [71]. Fig. 1(b) demonstrates the impact of different sampling steps while maintaining a CFG factor of 2. The number of sampling steps shows relatively minor influence on performance. Our choice of 30 steps in the main paper represents a balance between generation quality and computational efficiency.



(a) Results of varying CFG Factors



(b) Results of Varying Numbers of Sampling Steps

Figure 1 | **Results of varying CFG factors and numbers of sampling steps**. In Fig. (a), the number of sampling steps is set to 30. In Fig. (b), the CFG factor is set to 2.

C. Additional Qualitative Results

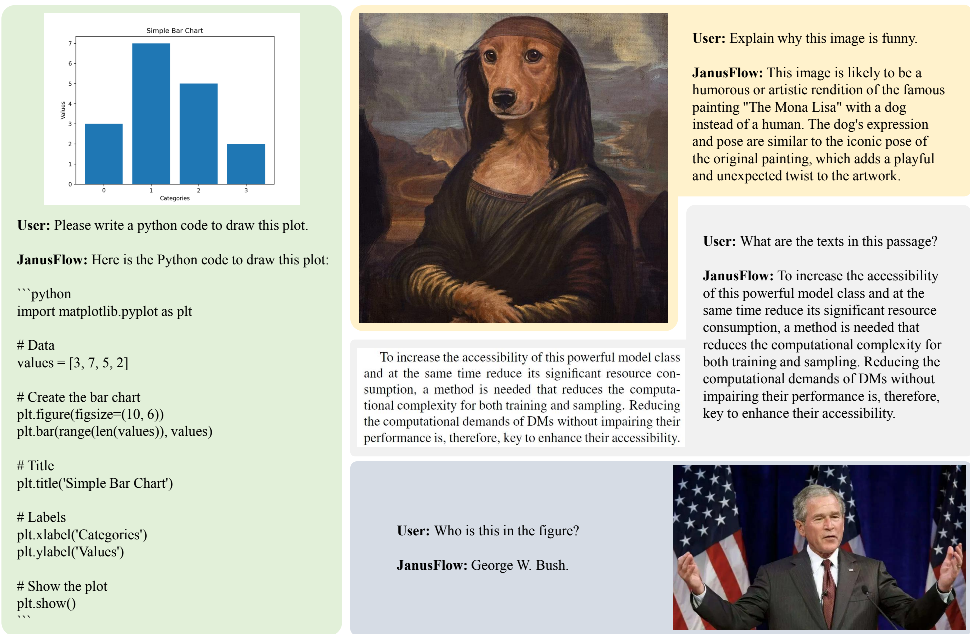
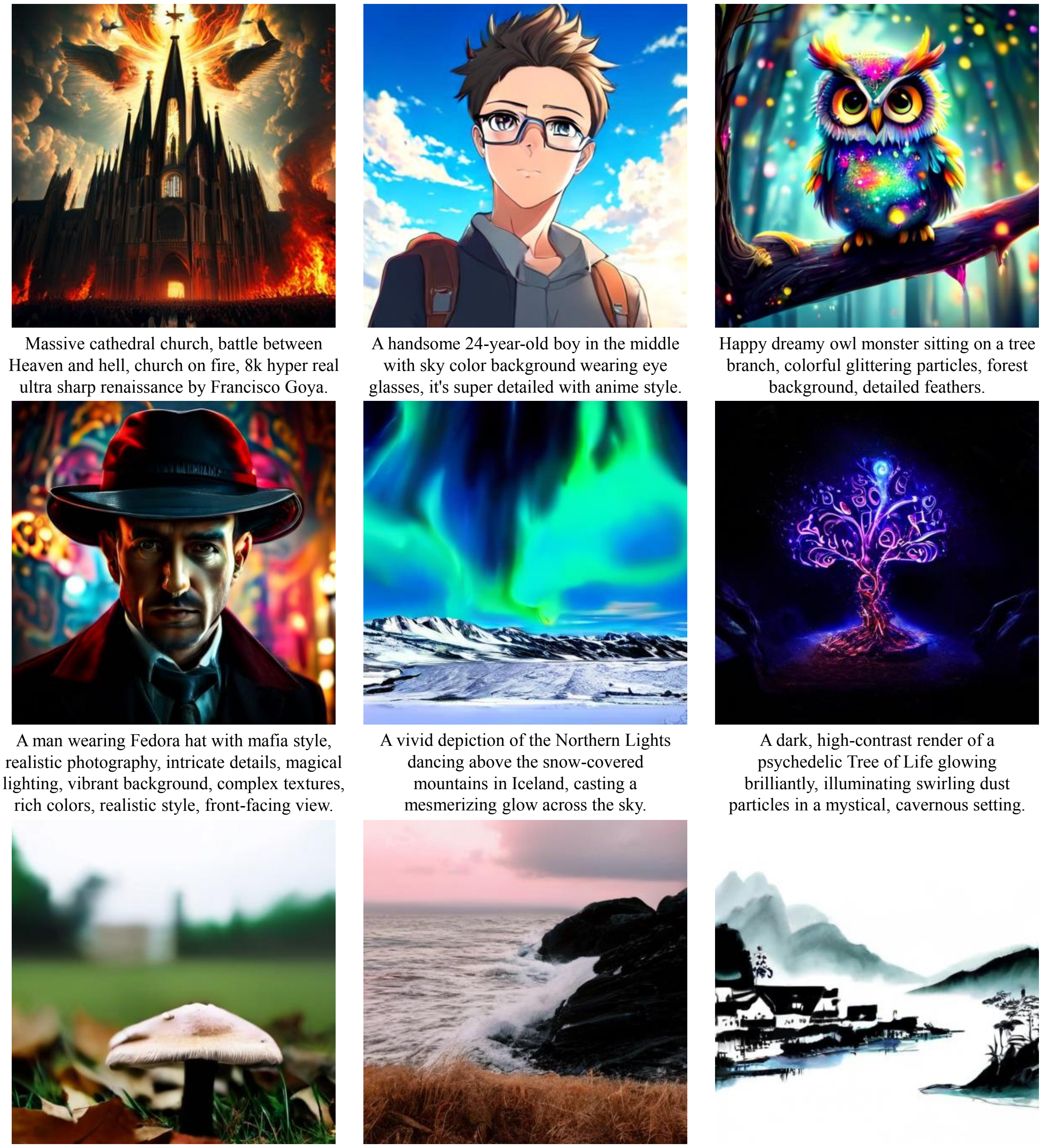


Figure 2 | More multimodal understanding cases.

Additional qualitative examples for both understanding and generation tasks are presented in Fig. 2 and Fig. 3, respectively. The understanding examples demonstrate JanusFlow's diverse capabilities, including code generation, person identification, character recognition, and visual reasoning. For image generation, our model exhibits strong performance in both visual quality and semantic alignment with input prompts.



Massive cathedral church, battle between Heaven and hell, church on fire, 8k hyper real ultra sharp renaissance by Francisco Goya.

A handsome 24-year-old boy in the middle with sky color background wearing eye glasses, it's super detailed with anime style.

Happy dreamy owl monster sitting on a tree branch, colorful glittering particles, forest background, detailed feathers.

A man wearing Fedora hat with mafia style, realistic photography, intricate details, magical lighting, vibrant background, complex textures, rich colors, realistic style, front-facing view.

A vivid depiction of the Northern Lights dancing above the snow-covered mountains in Iceland, casting a mesmerizing glow across the sky.

A dark, high-contrast render of a psychedelic Tree of Life glowing brilliantly, illuminating swirling dust particles in a mystical, cavernous setting.

The image features a mushroom growing on grassy ground amidst fallen leaves. Their caps are light brownish-white with visible gills underneath; the stems appear dark and sturdy. In the background, there's an out-of-focus scene that includes greenery and possibly some structures or trees shrouded by mist or fog, giving it a serene yet slightly eerie atmosphere. This photograph employs shallow depth of field to emphasize the mushrooms while blurring the surroundings for artistic effect.

The image captures a vast ocean view at either sunrise or sunset, with soft pink hues near the horizon blending into darker clouds above. Waves crash against rugged black rocks on the right, where water flows down onto smaller stones below. In the foreground, dry grass contrasts with the smooth sea surface. The scene feels tranquil but also reveals the raw power of nature through the interaction between the dynamic waves and the solid land.

A serene Chinese ink painting depicts a tranquil mountain village. Simple homes nestle at the foot of misty peaks, while a gentle river winds through the village. Bamboo and pine trees dot the landscape. The minimalist brushstrokes reflect a harmonious relationship between nature and human life, capturing the peaceful essence of the scene with elegant simplicity.

Figure 3 | More text-to-image generation results.

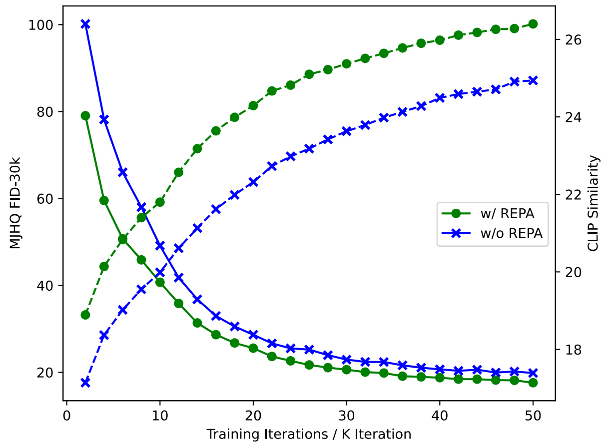


Figure 4 | The FID and CLIP similarity during the first 50,000 iterations.

D. Details of REPA Ablation

We provide the FID and CLIP similarity of the first 50,000 training iterations of the pre-train stage in Fig. 4 with and without representation alignment regularization. The gap between the two models demonstrates the benefits of using representation alignment regularization.