Imagine Before You Draw: Visual Prompt Engineering for Image Generation

Liyu Jia[1,*] **Fengda Zhang**[1,*,†] **Jiachun Pan**[2,*] **Kesen Zhao**[1] **Saining Zhang**[1]
Wang Lin[3] **Weijia Wu**[2] **Yue Liao**[2] **Aojun Zhou**[4] **Hanwang Zhang**[1]

[1]Nanyang Technological University [2]National University of Singapore
[3]Zhejiang University [4]The Chinese University of Hong Kong
ji0011yu@e.ntu.edu.sg fdzhang328@gmail.com

Abstract

Incorporating visual semantic representations as an intermediate step before image generation can reduce the modeling difficulty between text and images, thereby improving generation quality. Recent works such as X-Omni and BLIP3o-Next have explored this direction, but they typically use a two-stage external pipeline: a separate autoregressive model first generates semantic tokens, which are then fed as conditioning to an independent diffusion decoder. When the decoder is conditioned only on the compressed semantic plan, it cannot directly access the original input, which introduces an information bottleneck that limits detail preservation in downstream tasks such as editing. Internal architectures such as Transfusion, BAGEL, and Show-o2 avoid this bottleneck by enabling cross-modal interaction within a single model, but they still face the difficult text-to-pixel modeling gap without intermediate semantic guidance. We propose Visual Prompt Engineering (VPE), which can be seamlessly integrated into such internal frameworks. Specifically, the model first autoregressively generates visual semantic tokens (*e.g.*, SigLIP 2) as "visual prompts" that capture the semantic layout, then generates the full image tokens conditioned on this plan. We validate VPE across class-conditional generation, text-to-image generation, and image editing, covering various token types and model architectures. Results show that VPE can accelerate convergence, raise quality ceilings, and through internal integration, achieve substantially better editing preservation (PSNR: 26.76 vs. 19.92) than a matched external pipeline with semantic-only decoder conditioning, while maintaining competitive editing responsiveness.

1 Introduction

Recent multimodal models [1, 2, 3, 4, 5] have made rapid progress in image generation. A central challenge in these models is bridging the gap between high-level semantic conditioning (*e.g.*, class labels or text prompts) and low-level pixel-accurate image synthesis. Incorporating visual semantic representations as intermediate signals has emerged as a key trend, as it can reduce the modeling difficulty and improve generation quality.

Several recent works have begun exploring this direction. X-Omni [6] and BLIP3o-Next [7] employ SigLIP 2 [8] tokens as conditioning signals for image generation. However, they adopt an *external* architecture (Figure 1b) where a separate autoregressive model produces SigLIP 2 tokens that are then fed as conditioning to an independent diffusion model. When the diffusion decoder is conditioned only on the compressed semantic plan, it cannot directly access the original conditioning, creating an information bottleneck that is particularly harmful for tasks requiring detail preservation such as image editing. In contrast, *internal* architectures (Figure 1a) such as Transfusion [4], BAGEL [5], and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Show-o2 [2] avoid this bottleneck by enabling cross-modal interaction within a single model, but they have not incorporated visual semantic representations, leaving the difficult text-to-pixel modeling gap unaddressed. In addition, no prior work has systematically investigated whether visual semantic prompts benefit different types of image tokens (discrete tokens such as Emu3 [3] and continuous tokens such as VAE vectors [9]), how they affect convergence and quality ceilings, or what are the trade-offs between these two architectural paradigms.

Drawing an analogy from chain-of-thought reasoning in large language models [10], where decomposing complex reasoning into intermediate steps dramatically improves performance, we propose *Visual Prompt Engineering* (VPE) to address these challenges. VPE inserts a compact set of SigLIP 2 visual tokens that capture the high-level semantic layout of the target image before the model generates the full image tokens, transforming a single difficult generation step into two easier sub-problems: *semantic planning* (what to draw) followed by *detail rendering* (how to draw it). However, learning reliable semantic plans remains non-trivial. If the model is naively conditioned on ground-truth SigLIP 2 visual prompts during training, it can over-rely on these highly informative tokens and pay insufficient attention to other conditioning signals. At inference time, however, the model must condition on its own predicted visual prompts, which are inevitably imperfect; this train–inference discrepancy can lead to severe quality degradation. This issue is further amplified in continuous-token models, where the cross-entropy loss for predicting SigLIP 2 tokens and the flow-matching loss for image tokens operate at different scales, making joint optimization unstable. To address these challenges, we introduce a progressive training schedule that gradually increases the model's reliance on visual prompts (Section 2.2). Crucially, VPE supports both internal integration (shared attention within a single model) and external integration (separate AR + DiT), allowing us to systematically analyze these paradigms.

We validate VPE across class-conditional generation, text-to-image generation, and image editing, showing consistent improvements across these tasks. Our contributions are threefold:

- We propose VPE, a general technique for improving image generation by inserting SigLIP 2 visual tokens into internal architectures. Through experiments, we validate its effectiveness across discrete and continuous token types, various image generation tasks, and multiple model architectures including a controlled comparison of internal and external paradigms.
- We identify the train-inference gap and loss-scale imbalance as key challenges in VPE training, and provide a progressive training schedule that effectively addresses both issues for autoregressive and diffusion objectives.
- Our experimental results show that internal architectures preserve finer details in editing than external ones with semantic-only decoder conditioning, but directly modeling text-to-image within a single model is inherently harder to converge. VPE alleviates this difficulty by providing intermediate semantic guidance, accelerating the convergence of internal architectures while retaining their editing advantage.

2 Method

2.1 Visual Prompt Engineering

Given a conditioning signal c (class label, text prompt, or reference image) and a target image I, standard image generation models learn the mapping $c \rightarrow \mathbf{T}_{\text{img}}$, where $\mathbf{T}_{\text{img}}$ represents the image tokens (discrete codebook indices or continuous latent vectors). VPE introduces an intermediate semantic representation by first generating *visual prompts* $\mathbf{T}_{\text{vp}}$ (SigLIP 2 [8] tokens extracted from the target image) before generating the image tokens:

$$c \rightarrow \underbrace{\mathbf{T}_{\text{vp}}}_{\text{semantic plan}} \rightarrow \underbrace{\mathbf{T}_{\text{img}}}_{\text{detail rendering}} \tag{1}$$

During training, the visual prompts are obtained by encoding the ground-truth image I with a frozen visual semantic encoder, specifically SigLIP 2-Giant-Patch16-384 [8, 6] in this paper. With a patch size of 16, the encoder produces a compact token grid at $16\times$ spatial downsampling: for example, $384 \times 384 \rightarrow 576$ tokens and $224 \times 224 \rightarrow 196$ tokens. The resulting tokens are then discretized using the frozen SigLIP-VQ tokenizer from X-Omni [6], which adopts a codebook of 16,384 entries. At inference time, the model autoregressively generates $\mathbf{T}_{\text{vp}}$ from c, then conditions image token

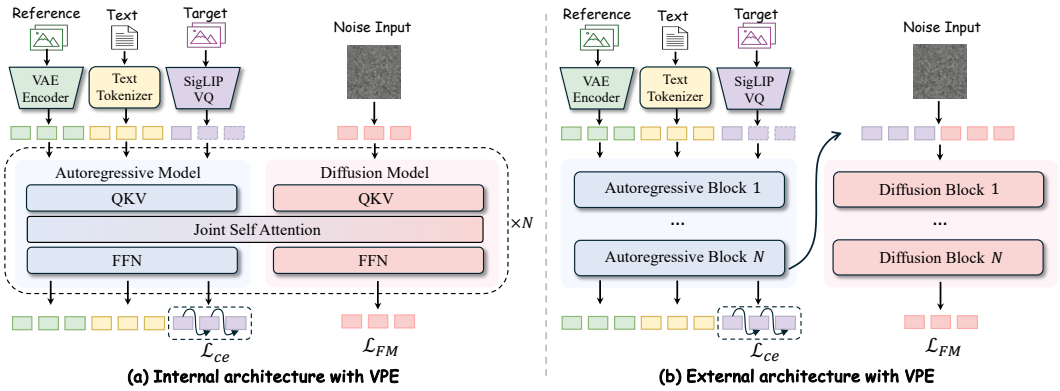

Figure 1: **VPE in two architectural frameworks.** VPE inserts SigLIP 2 visual prompts before image generation as a semantic plan. **(a)** In the internal architecture, all modalities (text, SigLIP 2 tokens, image tokens) share attention within a single model via Eq. 4, enabling direct cross-referencing during generation. **(b)** In the external architecture, an AR model first produces SigLIP 2 features, which are then routed to a separate DiT decoder. While both frameworks benefit from VPE for generation, the external pipeline with semantic-only DiT conditioning introduces an information bottleneck that limits editing fidelity (Section 4.3).

generation on both c and the self-generated $\mathbf{T}_{\text{vp}}$. SigLIP 2 tokens capture high-level semantic content (object categories, spatial layout, color distribution) while abstracting away pixel-level details, making them an effective intermediate representation for semantic planning.

Compatibility with different token types. A key advantage of VPE is its generality. The visual prompt generation is always autoregressive (predicting discrete SigLIP 2 token indices via cross-entropy loss), while the subsequent image generation can use *any* framework. For **discrete** image tokens such as Emu3 [3], the full sequence is $[c, \mathbf{T}_{\text{vp}}, \mathbf{T}_{\text{img}}]$, trained end-to-end with cross-entropy loss. For **continuous** image tokens such as VAE vectors, the image is generated via diffusion or flow matching [11] conditioned on the visual prompts, using cross-entropy loss for $\mathbf{T}_{\text{vp}}$ and the corresponding continuous generation loss for $\mathbf{T}_{\text{img}}$. VPE only adds SigLIP 2 vocabulary embeddings to the model, resulting in $< 4\%$ additional parameters in our experiments.

2.2 Progressive Training Schedule

In our experiments, we observe that if the model is trained with ground-truth visual prompts from the start, it becomes overly dependent on them. At inference time, the model must rely on its own autoregressively generated SigLIP 2 tokens, which are inevitably imperfect in early training. This train-inference discrepancy causes significant quality degradation. To ensure the model can still generate reasonable images even when the visual prompts are noisy or incorrect, we introduce a progressive schedule that gradually increases the model's reliance on visual prompts, preventing it from degenerating into a simple copying task that only works with perfect SigLIP 2 conditioning. Additionally, for continuous token models, the cross-entropy loss for SigLIP 2 tokens and the flow matching loss for image tokens operate at different scales. Without careful balancing, the cross-entropy gradient can dominate and impede the learning of flow matching denoising.

For discrete token models. We define a step-dependent masking probability:

$$p_{\text{mask}}(s) = p_0 \cdot (1 - \sigma(s)) + p_1 \cdot \sigma(s), \quad \sigma(s) = \frac{1}{1 + e^{-k(s/S - 0.5)}} \tag{2}$$

where s is the current step, S is total steps, p_0=0.95, p_1=0.05, and k=10. At each training step, visual prompt tokens are replaced with padding tokens with probability p_{mask}, and no loss is computed on masked positions. This ensures early training focuses on learning image generation from c alone, while later training gradually introduces visual prompts as both targets and conditions.

For continuous token models. Since flow matching and cross-entropy losses operate at different scales, we keep visual prompts unmasked but modulate the cross-entropy loss coefficient:

$$\lambda_{\text{CE}}(s) = \lambda_0 \cdot (1 - \sigma(s)) + \lambda_1 \cdot \sigma(s) \tag{3}$$

with λ_0=0.03, λ_1=0.1, and flow matching loss coefficient fixed at 1. This prevents the cross-entropy gradient from overwhelming the diffusion objective in early training. Both schedules are ablated in Appendix F.3.

2.3 Internal vs. External Architectures

Current image generation models can be broadly categorized into two architectural paradigms (Figure 1). **External** architectures, such as X-Omni [6] and BLIP3o-Next [7], use a two-stage pipeline where an autoregressive model first generates semantic tokens, which are then fed as conditioning to a separate diffusion decoder. The AR and DiT [12] share no parameters; the DiT receives the semantic features as cross-attention conditioning. **Internal** architectures process all modalities within a single model. Transfusion [4] applies causal attention for text tokens and bidirectional attention for image patches, enabling cross-modal interaction where the image generation process can directly access the conditioning information. BAGEL [5] further extends this with a Mixture-of-Transformers (MOT) design that uses parameter-disjoint experts within shared attention. Notably, external architectures have already begun incorporating visual semantic representations (X-Omni and BLIP3o-Next both use SigLIP 2 tokens), while internal architectures have not yet explored this direction. We therefore study how to integrate VPE into both paradigms:

Internal$_{\text{+VPE}}$. Following the MOT design, the AR model handles text and SigLIP 2 tokens (cross-entropy loss), while the DiT model handles image tokens (flow matching loss). At each layer, both models use independent Q/K/V/O projections but compute attention over the *concatenated* key-value sequence:

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}_i \cdot [\mathbf{K}_1; \mathbf{K}_2]^\top}{\sqrt{d}}\right) \cdot [\mathbf{V}_1; \mathbf{V}_2], \quad i \in \{1, 2\} \tag{4}$$

where $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i$ are the query, key, and value projections of model i. This enables the DiT model to directly attend to text, reference images, and SigLIP 2 features simultaneously. The attention mask is causal over the AR tokens, so they never attend to the noisy image tokens. Their keys/values are therefore invariant across denoising steps and can be cached (Algorithm 1, Appendix F.2).

External$_{\text{+VPE}}$. The autoregressive model generates SigLIP 2 tokens from the conditioning signal, then a separate DiT model uses the SigLIP hidden features as cross-attention conditioning to generate image tokens via flow matching. External$_{\text{+VPE}}$ is a controlled reproduction of the external paradigm of X-Omni and BLIP3o-Next with matched parameters, data, and initialization, and performs on par with these systems (Section 4.2).

In our experiments, both architectures have identical total parameter counts (4.16B for text-to-image, 4.57B for editing with reference image encoder), enabling fair comparison.

3 Related Work

Multimodal generation. Recent image generation models span diverse architectures. Autoregressive models [13, 14, 15, 3] generate images via next-token prediction over discrete tokens [9, 16], recent variants [17, 18, 19] explore scalable alternatives. Diffusion and flow-based models [20, 11, 21] and their transformer variants [12, 22, 23, 24] achieve high quality through denoising. Unified multimodal models [1, 2, 4, 25, 5] combine understanding and generation in a single architecture. Our work builds upon these frameworks and introduces visual prompts as a general enhancement.

Visual semantic tokens as intermediate representations. Vision-language encoders [26, 8, 27] produce semantic features widely used as conditioning signals [28, 29]. X-Omni [6] and BLIP3o-Next [7] generate SigLIP 2 tokens to condition external diffusion models. However, these works treat visual tokens as a fixed architectural choice rather than systematically studying their impact. In contrast, we integrate visual prompts into internal architectures, investigate them across diverse frameworks and tasks, and compare internal and external integration under matched settings.

Table 1: **C2I experimental setup.**

Framework	LlamaGen	Transfusion
Initialization	—	Qwen2.5-0.5B
Resolution	512×512	256×256
Image tokens	1,024	256
SigLIP 2 tokens	196	121
Hidden size	1,280	896
$\text{Model}_{+\text{VPE}}$ params	$1.07\text{B}_{+0.04\text{B}}$	$0.37\text{B}_{+0.02\text{B}}$

Table 2: **C2I on ImageNet.** Without CFG.

Method	Ep/Step	FID↓
Discrete (512×512)		
LlamaGen	100	24.56
$\text{LlamaGen}_{+\text{VPE}}$ (ours)	100	**8.69**
Continuous (256×256)		
Transfusion	266k	15.18
$\text{Transfusion}_{+\text{VPE}}$ (ours)	266k	**11.43**

Image editing. InstructPix2Pix [30] pioneered instruction-based editing using classifier-free guidance over both text and image conditions. Subsequent works have improved editing through curated data and AR formulations [31, 32, 33], attention and inversion techniques [34, 35, 36], and unified frameworks [37, 38, 39]. Under matched settings, our work shows that internal models that jointly attend to reference images, instructions, and semantic features can preserve details during editing, while external models whose decoder is conditioned only on semantic features preserve them poorly, and giving the external decoder direct reference access largely closes this gap (Section 4.3).

4 Experiments

We evaluate VPE across three progressively complex tasks: class-to-image generation (§4.1), text-to-image generation (§4.2), and a controlled analysis of internal vs. external architectures for editing (§4.3).

4.1 Class-to-Image Generation

Setup. We train on ImageNet-1K [40] with two frameworks (Table 1) to validate the effectiveness of VPE under both discrete and continuous schemes. VPE adds only a randomly initialized SigLIP 2 embedding layer ($< 4\%$ parameter increase). All results are evaluated on the ImageNet validation set without classifier-free guidance (CFG=1.0).

- **Discrete.** We adopt LlamaGen [15] with the Emu3 [3] tokenizer to generate 512×512 images (white-padded) as 1,024 tokens. $\text{LlamaGen}_{+\text{VPE}}$ prepends 196 SigLIP 2 tokens (from the 224×224 white-padded target image), increasing per-sample length from 1,025 to 1,221 tokens. Trained from scratch for 100 epochs with batch size 128 for both LlamaGen and $\text{LlamaGen}_{+\text{VPE}}$, where the latter consumes 19.1% more tokens per epoch.
- **Continuous.** We adopt Transfusion [4] to generate 256×256 images (center-cropped) as 256 VAE tokens, with fixed-length sequence packing [5] at 10,240 tokens per step. $\text{Transfusion}_{+\text{VPE}}$ prepends 121 SigLIP 2 tokens (from the 176×176 center-cropped target image) but sees approximately 32% fewer images per step due to longer per-sample sequences. The model loads non-embedding weights from Qwen2.5-0.5B and randomly initializes 1,000 class token embeddings, resulting in 0.37B total parameters. Trains 130k shared steps without VPE, then forks into Transfusion and $\text{Transfusion}_{+\text{VPE}}$ branches for 136k additional steps.

Results. VPE consistently improves final generation quality and convergence speed across LlamaGen and Transfusion, validating its effectiveness under both discrete and continuous token paradigms.

- **Quality ceiling (Table 2, Figure 2).** At comparable training budgets, VPE substantially raises the quality ceiling. $\text{LlamaGen}_{+\text{VPE}}$ achieves FID 8.69 vs. 24.56 at epoch 100. $\text{Transfusion}_{+\text{VPE}}$ achieves FID 11.43 vs. 15.18 at step 266k. The training loss curves (Figure 2b,d) confirm that VPE converges to a lower loss in both frameworks.
- **Convergence acceleration (Figure 2).** VPE dramatically accelerates convergence. As shown in Figure 2a, $\text{LlamaGen}_{+\text{VPE}}$ at epoch 50 already surpasses LlamaGen at epoch 100 (FID 22.07 vs. 24.56). Since VPE adds 19.1% more tokens per epoch, $\text{LlamaGen}_{+\text{VPE}}$ at epoch 50 consumes approximately 60% of LlamaGen's total token budget at epoch 100. For Transfusion, where both branches consume the same tokens per step, $\text{Transfusion}_{+\text{VPE}}$ at step 184k surpasses Transfusion at step 266k (FID 14.02 vs. 15.18, Figure 2c). This illustrates that VPE effectively reduces the difficulty of directly modeling images from text.

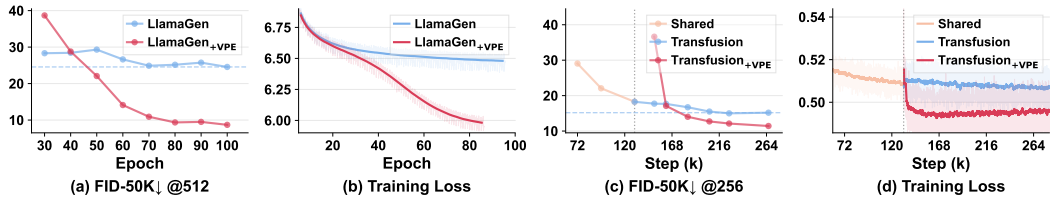

Figure 2: **Convergence on ImageNet. (a)** LlamaGen FID-50K (512×512): VPE starts higher due to progressive masking ($p_{\text{mask}}=0.95$) but surpasses the baseline from epoch 50, with the gap widening to 15.87 by epoch 100. **(b)** LlamaGen training loss converges to a lower value with VPE. **(c)** Transfusion FID-50K (256×256): both branches fork from the shared checkpoint (dashed line). The initial high FID of VPE at 150k reflects the train-inference gap on self-generated visual prompts, which resolves quickly. **(d)** Transfusion flow loss curves for the shared base and both branches.

Table 3: **Text rendering on TextAtlas benchmarks.** Metrics include CLIP Score (CS) ↑, OCR Accuracy (Acc.) ↑, F1 Score ↑, and Character Error Rate (CER) ↓. All results use CFG=5.5. Best open-source values in **bold**. Gray : closed-source models. $*$: continued training on 6.35M text rendering data.

	TextScenesHQ				TextVisionBlend				StyleTextSynth			
Method	CS↑	Acc↑	F1↑	CER↓	CS↑	Acc↑	F1↑	CER↓	CS↑	Acc↑	F1↑	CER↓
Grok 3	0.32	35.07	37.94	0.57	0.17	41.54	44.22	0.57	0.29	15.82	21.40	0.73
DALL-E 3 [41]	0.34	69.26	51.63	0.67	0.19	8.38	7.94	0.93	0.29	30.58	38.25	0.78
AnyText [42]	0.22	0.42	0.81	0.95	–	–	–	–	0.25	0.35	0.66	0.98
TextDiffuser-2 [43]	0.23	0.66	1.25	0.96	–	–	–	–	0.25	0.76	1.46	0.99
PixArt-Σ [24]	0.23	0.34	0.53	0.91	0.19	2.40	1.57	0.83	0.28	0.42	0.62	0.90
Infinity-2B [19]	0.23	1.06	1.74	0.88	**0.20**	2.98	3.44	0.83	0.27	0.80	1.42	0.93
SD3.5 Large [21]	0.24	**19.03**	24.45	**0.73**	0.18	14.55	16.25	0.88	0.28	27.21	33.86	0.73
Show-o2* (1.5B) [2]	**0.29**	2.62	4.36	0.88	0.16	76.45	77.40	0.22	**0.29**	21.83	27.72	**0.59**
Show-o2$_{+\text{VPE}}$ (ours, 1.5B)	0.26	16.29	**26.07**	0.77	0.16	**80.07**	**82.55**	**0.20**	0.26	**32.91**	**43.78**	0.60

- **Early-stage behavior (Figure 2).** LlamaGen$_{+\text{VPE}}$'s early-epoch FID increase (epochs 30–40) is expected: progressive masking ($p_{\text{mask}}=0.95$) limits the visual prompt signal initially. Once the model learns to leverage visual prompts (epoch 50+), improvement is decisive and monotonic. For Transfusion$_{+\text{VPE}}$, the initial high FID at step 150k reflects the train-inference gap: training uses ground-truth SigLIP 2 tokens while inference relies on self-generated ones. This resolves quickly as the model learns to produce reliable visual prompts.
- **Further analysis.** Ablations of the progressive schedule, a comparison between ground-truth and predicted visual prompts, and training cost are given in Appendices F.3, F.4, and F.1.

4.2 Text-to-Image Generation

Text rendering setup. We fine-tune the pre-trained Show-o2-1.5B model [2] on 6.35M English text rendering images collected from TextAtlas [47] and AnyWord-3M [42], excluding Chinese text images and academic paper images whose text becomes illegible after white-padding preprocessing. Since the official Show-o2-1.5B was not trained on dense text data [2], we also continue-train the baseline on the same 6.35M data (denoted Show-o2*) to ensure improvements stem from VPE rather than additional data. Show-o2$_{+\text{VPE}}$ uses batch size 8 with max sequence length 1,856, while Show-o2* uses batch size 12 with max sequence length 1,280, resulting in closely matched per-step token throughput ($< 3.5\%$ difference). Both models are trained for 3 epochs on 24 H200 GPUs. At this matched token throughput, the two runs have nearly identical cost (2,186 vs. 2,224 GPU-hours, Appendix F.1). The VPE model's CE loss follows Eq. 3 with total training steps S=100k.

Text rendering results. Show-o2$_{+\text{VPE}}$ achieves notable improvements over existing open-source models and Show-o2*, demonstrating that visual prompts effectively reduce the difficulty of directly modeling text-to-image within internal architectures such as Show-o2.



Figure 3: **Text rendering comparison.** Show-o2$_{+\text{VPE}}$ generates more accurate text compared to Show-o2*, consistent with the quantitative improvements in Table 3.

Table 4: **GenEval benchmark.** External$_{+\text{VPE}}$ at 4.16B surpasses many larger models, demonstrating the effectiveness of VPE for T2I generation quality.

Method	#Params	Single	Two	Count	Color	Pos.	C.Attr.	Overall↑
ILLUME [44]	7B	0.99	0.86	0.45	0.71	0.39	0.28	0.61
Transfusion [4]	7B	–	–	–	–	–	–	0.63
D-DiT [45]	2B	0.97	0.80	0.54	0.76	0.32	0.50	0.65
Emu3 [3]	8B	–	–	–	–	–	–	0.66
SD3-Medium [21]	–	0.99	0.94	0.72	0.89	0.33	0.60	0.74
Show-o2 [2]	7B	**1.00**	0.87	0.58	**0.92**	0.52	0.62	0.76
BLIP3o-NEXT-SFT [7]	3B	0.98	0.88	0.52	0.86	0.74	0.61	0.77
MetaQuery-XL [46]	7B	–	–	–	–	–	–	0.80
Janus-Pro [25]	7B	0.99	0.89	0.59	0.90	**0.79**	0.66	0.80
X-Omni [6]	7B	0.98	**0.95**	**0.75**	0.91	0.71	**0.68**	**0.83**
Internal$_{+\text{VPE}}$ (ours)	4.16B	0.98	0.91	0.68	0.86	0.77	0.55	0.79
External$_{+\text{VPE}}$ (ours)	4.16B	0.98	0.90	0.69	0.86	0.76	**0.68**	0.81

- **Comparison with open-source models (Table 3).** Show-o2$_{+\text{VPE}}$ at 1.5B parameters outperforms all open-source baselines. It only slightly trails SD3.5 Large on TextScenesHQ, but given that SD3.5 Large has 8.1B parameters (5.4× larger), this already demonstrates strong competitiveness. On TextVisionBlend, it achieves the highest accuracy (80.07%) and F1 (82.55) among all methods including closed-source ones.
- **Controlled comparison with Show-o2* (Table 3).** Compared to Show-o2* trained on identical data without VPE, Show-o2$_{+\text{VPE}}$ improves OCR accuracy from 2.62% to 16.29% (6.2×) on TextScenesHQ, from 76.45% to 80.07% on TextVisionBlend, and from 21.83% to 32.91% on StyleTextSynth. These consistent improvements confirm that visual prompts help the model "plan" text layout before rendering. By first generating SigLIP 2 tokens that encode the semantic content, the model produces more accurate character structures during the subsequent generation phase.
- **Qualitative results (Figure 3).** Show-o2$_{+\text{VPE}}$ generates more accurate and readable text compared to Show-o2*, consistent with the quantitative improvements.
- **Image quality.** VPE also improves the LAION aesthetic score (Appendix F.6).

General T2I setup. Beyond text rendering, we evaluate whether VPE improves general T2I quality. We adopt two mainstream architectures illustrated in Figure 1: Internal$_{+\text{VPE}}$ (mixture-of-transformers) and External$_{+\text{VPE}}$ (AR + DiT). Both models load all parameters from Show-o2-1.5B [2] except for the lightweight SigLIP 2 embedding layer, which is randomly initialized. Since the T2I task has no image input, we remove the VAE encoder, resulting in 4.16B parameters. We collect 32M T2I images from the publicly available BLIP3o [7] data, combined with the 6.35M text rendering data described above. Training proceeds in two stages:

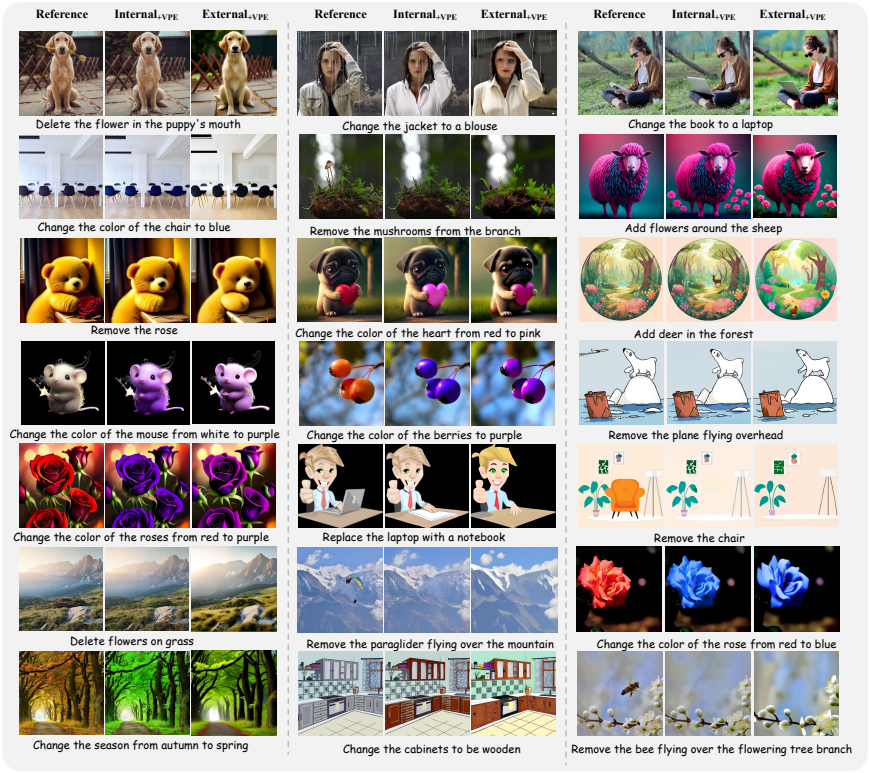

Figure 4: **Editing comparison.** Each triplet shows Reference / Internal$_{+\text{VPE}}$ / External$_{+\text{VPE}}$. Internal$_{+\text{VPE}}$ preserves unedited regions while External$_{+\text{VPE}}$ regenerates inconsistent backgrounds.

- **Stage-1** (text-to-SigLIP 2 alignment): trains only the AR model for 4 epochs on the 39M mixed data. The resulting AR checkpoint is shared by both architectures in Stage-2, ensuring identical text-to-SigLIP 2 alignment capability.
- **Stage-2** (image generation): trains on the same 39M mixed data, following each framework's standard training paradigm. Internal$_{+\text{VPE}}$ performs joint training of both the AR and DiT models. External$_{+\text{VPE}}$ freezes the pretrained AR model and only trains the DiT.

This setup also serves as the foundation for the editing experiments in Section 4.3.

General T2I results. Table 4 shows GenEval [48] results. At only 4.16B parameters, External$_{+\text{VPE}}$ surpasses most compared 7B+ models on overall score, performs on par with X-Omni (0.83, 7B, with RL post-training), and outperforms BLIP3o-NEXT-SFT (0.77, 3B), while Internal$_{+\text{VPE}}$ remains competitive with 7B-scale models. This also confirms that our controlled External$_{+\text{VPE}}$ is a faithful, competitive instance of the external paradigm. This illustrates that VPE reduces the modeling difficulty and improves generation quality, enabling smaller models to exceed much larger ones. On GenAI-Bench [49], DSG-1k [50], and GenEval with 3 seeds (Table 15), both models match T2I-R1 [51] and perform on par with each other.

4.3 Analysis: Internal vs. External architecture

Setup. Both Internal$_{+\text{VPE}}$ and External$_{+\text{VPE}}$ from Section 4.2 achieve comparable T2I quality (0.79 and 0.81 on GenEval), providing a relatively fair baseline for comparing internal and external paradigms on editing. Full training details are provided in Appendix C. Both models add a reference image encoder (initialized from Show-o2-1.5B [2] weights), bringing total parameters to 4.57B with the VAE encoder included. Starting from the Stage-2 T2I checkpoints, we fine-tune on 5.4M editing pairs from NHR-Edit [52] ($\times$3 upsampling, 2.16M), UnicEdit [37] (2.03M), ShareGPT4o [53] ($\times$5, 233k), and Pico-Banana [38] ($\times$4, 1.03M), with batch size 5 per GPU on 24 H200 GPUs for approximately 1.5 epochs (70k steps). We adopt InstructPix2Pix-style [30] conditional dropout for text and reference image conditions, where both models drop text and SigLIP 2 tokens *jointly*, and

Table 5: **Internal vs. External architecture comparison.** Both architectures achieve comparable T2I quality ((a), Table 4, and Table 15), but Internal$_{+\text{VPE}}$ dramatically outperforms External$_{+\text{VPE}}$ in editing preservation (b). In (b), the "+" rows are ablations of External$_{+\text{VPE}}$. They start from the same Stage-2 checkpoint, fully unfrozen, with the same data and 70k steps. "+ ref access" also feeds the reference image, text, and SigLIP 2 features to the DiT decoder. BLIP3o-NEXT uses its released editing weights.

(a) TextAtlas (text rendering, CFG=5.5)

	TextScenesHQ				TextVisionBlend				StyleTextSynth			
Method	CS	Acc↑	F1↑	CER↓	CS	Acc↑	F1↑	CER↓	CS	Acc↑	F1↑	CER↓
Internal$_{+\text{VPE}}$	0.27	**22.70**	**34.37**	**0.72**	0.16	77.59	**80.25**	0.23	0.27	**38.63**	**49.62**	0.54
External$_{+\text{VPE}}$	0.27	19.07	29.69	0.75	0.16	77.59	80.15	0.23	0.27	38.24	49.06	0.54

(b) PIE-Bench (Editing, Image CFG=1.0, Text CFG=2.5)

Method	Struct. Dist.↓	PSNR↑	LPIPS↓	MSE↓	SSIM↑	CLIP Whole↑	CLIP Edit↑
Internal$_{+\text{VPE}}$	**24.60**	**26.76**	**58.61**	**46.76**	**86.66**	23.00	20.50
BLIP3o-NEXT [7]	98.87	15.94	198.84	333.50	62.92	25.09	**22.28**
External$_{+\text{VPE}}$	61.66	19.92	158.09	150.88	70.02	23.09	20.69
+ unfrozen	62.32	18.56	168.66	203.68	68.90	**25.34**	21.89
+ unfrozen + ref access	26.67	25.76	68.66	53.86	81.98	23.56	20.63

the no-text, no-reference, and null cases are mutually exclusive (5% each) (details in Appendix D). Inference uses classifier-free guidance with text CFG 2.5 and image CFG 1.0. We evaluate on PIE-Bench [36].

Results. Table 5 shows that while both architectures achieve comparable T2I quality and editing *responsiveness* (similar CLIP similarity scores), Internal$_{+\text{VPE}}$ dramatically outperforms External$_{+\text{VPE}}$ in detail *preservation*, with $2.5\times$ better Structure Distance (24.60 vs. 61.66), $+6.84$ dB PSNR (26.76 vs. 19.92), $2.7\times$ better LPIPS (58.61 vs. 158.09), and $+16.6$ SSIM (86.66 vs. 70.02). This gap arises despite identical parameter counts, training data, and training duration. The ablations in Table 5b identify its source: unfreezing the external model does not close the gap (PSNR 18.56), while additionally routing the reference image, text, and SigLIP 2 features into the DiT decoder reduces it to 1.0 dB (PSNR 25.76 vs. 26.76). The gap therefore stems from the semantic-only DiT conditioning adopted by existing external systems such as X-Omni and BLIP3o-Next. On I2EBench [54] (Appendix F.7), Internal$_{+\text{VPE}}$ wins all low-level dimensions and is comparable on high-level ones. Figure 4 shows representative examples: Internal$_{+\text{VPE}}$ faithfully preserves unedited regions while External$_{+\text{VPE}}$ regenerates the entire scene.

We provide an explanation for this gap from an information perspective, for an external decoder conditioned only on semantic features. Let $\mathbf{x}$ denote the reference image, t the edit instruction, and $\mathcal{P} \subseteq \mathbf{x}$ the region that should remain unchanged. An ideal edit modifies only the attributes specified by t while keeping $\mathcal{P}$ intact.

In the external pipeline, the AR model takes the reference image (via VAE encoding) and the edit instruction, producing $\mathbf{s}' = f_{\text{AR}}(\mathbf{x}, t)$, the SigLIP 2 representation of the target image after editing. Since SigLIP 2 is a semantic encoder that discards fine-grained spatial details [8], $\mathbf{s}'$ inevitably loses the pixel-level information (textures, edges, background details) that must be preserved. The DiT decoder generates $\hat{\mathbf{x}} = g_{\text{DiT}}(\mathbf{s}')$, and by the data processing inequality:

$$I(\hat{\mathbf{x}}; \mathcal{P}) \leq I(\mathbf{s}'; \mathcal{P}) < H(\mathcal{P}) \tag{5}$$

The first inequality holds because $\hat{\mathbf{x}}$ is a function of $\mathbf{s}'$. The strict inequality holds because $\mathbf{s}'$ is a SigLIP 2 representation that has irreversibly discarded the fine-grained details of $\mathcal{P}$.

In contrast, the internal design allows the DiT to attend jointly to the full sequence $[\mathbf{x}, t, \mathbf{s}']$ at every layer through shared attention (Eq. 4):

$$I(\hat{\mathbf{x}}; \mathcal{P}) \leq I(\mathbf{x}, t, \mathbf{s}'; \mathcal{P}) = H(\mathcal{P}) \tag{6}$$

since $\mathcal{P} \subseteq \mathbf{x}$. The model can cross-reference t (what to change) with $\mathbf{x}$ (original details) at every decoding step, using $\mathbf{s}'$ purely for semantic guidance. No information bottleneck exists because the

decoder has direct access to $\mathbf{x}$ throughout the generation process. An external decoder given direct reference access likewise escapes the first bound, matching the ablation above.

5 Conclusion

We have presented Visual Prompt Engineering (VPE), which integrates visual semantic planning into unified architectures, together with the progressive training schedule this requires. Our systematic study demonstrates that VPE accelerates convergence and raises quality ceilings across discrete and continuous token types, significantly enhances text rendering, and through internal architecture integration enables effective image editing with strong detail preservation. Under the evaluated configurations, internal architectures with joint self-attention preserve editing details far better than external ones with semantic-only DiT conditioning: internal joint attention provides reference access natively, while external pipelines must inject it explicitly.

Acknowledgments and Disclosure of Funding

This research is supported by National Research Foundation, Singapore, NRF-NRFI10-2024-0004. The authors declare no competing interests.

References

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

[2] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

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

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

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

[6] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

[7] Jiuhai Chen, Le Xue, Zhiyang Xu, Xichen Pan, Shusheng Yang, Can Qin, An Yan, Honglu Zhou, Zeyuan Chen, Lifu Huang, et al. Blip3o-next: Next frontier of native image generation. *arXiv preprint arXiv:2510.15857*, 2025.

[8] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

[10] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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

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

[13] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems*, 34:19822–19835, 2021.

[14] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

[15] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

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

[17] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[18] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[19] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15733–15744, 2025.

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

[21] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[22] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[23] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[24] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.

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

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

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

[28] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[29] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

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

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

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

[33] Jiteng Mu, Nuno Vasconcelos, and Xiaolong Wang. Editar: Unified conditional generation with autoregressive models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7899–7909, 2025.

[34] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[35] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023.

[36] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023.

[37] Keming Ye, Zhipeng Huang, Canmiao Fu, Qingyang Liu, Jiani Cai, Zheqi Lv, Chen Li, Jing Lyu, Zhou Zhao, and Shengyu Zhang. Unicedit-10m: A dataset and benchmark breaking the scale-quality barrier via unified verification for reasoning-enriched edits. *arXiv preprint arXiv:2512.02790*, 2025.

[38] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025.

[39] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[41] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[42] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023.

[43] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser-2: Unleashing the power of language models for text rendering. In *European Conference on Computer Vision*, pages 386–402. Springer, 2024.

[44] Chunwei Wang, Guansong Lu, Junwei Yang, Runhui Huang, Jianhua Han, Lu Hou, Wei Zhang, and Hang Xu. Illume: Illuminating your llms to see, draw, and self-enhance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21612–21622, 2025.

[45] Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2779–2790, 2025.

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

[47] Alex Jinpeng Wang, Dongxing Mao, Jiawei Zhang, Weiming Han, Zhuobai Dong, Linjie Li, Yiqi Lin, Zhengyuan Yang, Libo Qin, Fuwei Zhang, et al. Textatlas5m: A large-scale dataset for dense text image generation. *arXiv preprint arXiv:2502.07870*, 2025.

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

[49] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, et al. Genai-bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024.

[50] Jaemin Cho, Yushi Hu, Jason Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *International conference on learning representations*, volume 2024, pages 15625–15645, 2024.

[51] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *Advances in Neural Information Processing Systems*, 38:39856–39890, 2026.

[52] Maksim Kuprashevich, Grigorii Alekseenko, Irina Tolstykh, Georgii Fedorov, Bulat Suleimanov, Vladimir Dokholyan, and Aleksandr Gordeev. Nohumansrequired: Autonomous high-quality image editing triplet mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6059–6068, 2026.

[53] Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025.

[54] Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. *Advances in Neural Information Processing Systems*, 37:41494–41516, 2024.

[55] 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. In *International Conference on Learning Representations*, volume 2024, pages 1862–1874, 2024.

[56] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

A Full Class-to-Image Results

Tables 6 and 7 report the complete evaluation metrics at more checkpoints for the class-to-image experiments.

Table 6: **LlamaGen results on ImageNet 512×512.**

	LlamaGen$_{+\text{VPE}}$				LlamaGen			
Epoch	FID↓	IS↑	Prec↑	Rec↑	FID↓	IS↑	Prec↑	Rec↑
30	38.68	34.1	0.453	0.611	28.32	40.8	0.480	0.647
40	28.79	45.7	0.502	0.623	28.43	40.8	0.479	0.655
50	22.07	53.7	0.520	0.632	29.31	39.7	0.472	0.663
60	14.12	74.5	0.584	0.621	26.63	42.5	0.484	0.660
70	10.91	82.4	0.598	0.626	24.89	44.2	0.497	0.660
80	9.35	88.8	0.614	0.629	25.16	44.4	0.492	0.667
90	9.50	92.1	0.611	0.625	25.74	43.9	0.497	0.658
100	8.69	95.7	0.616	0.624	24.56	45.1	0.498	0.666

Table 7: **Transfusion results on ImageNet 256×256.** Steps 72k–130k are shared training before the fork point. "—" indicates the model variant does not exist at that step.

	Transfusion$_{+\text{VPE}}$				Transfusion			
Step (k)	FID↓	IS↑	Prec↑	Rec↑	FID↓	IS↑	Prec↑	Rec↑
72	—	—	—	—	29.05	45.9	0.421	0.614
96	—	—	—	—	22.07	59.3	0.468	0.621
130	—	—	—	—	18.26	69.6	0.488	0.629
140	198.92	4.1	0.084	0.376	17.91	69.8	0.492	0.629
150	36.62	57.5	0.518	0.354	17.73	70.1	0.495	0.626
162	17.10	147.8	0.676	0.271	17.66	70.2	0.496	0.628
172	15.32	166.5	0.689	0.280	17.00	72.4	0.501	0.629
184	14.02	172.0	0.691	0.307	16.72	73.4	0.497	0.632
206	12.72	166.1	0.679	0.343	15.50	77.5	0.511	0.633
226	12.12	159.6	0.672	0.369	15.00	80.2	0.513	0.628
246	11.87	152.9	0.664	0.388	15.26	79.0	0.509	0.629
266	11.43	148.2	0.660	0.404	15.18	78.8	0.508	0.637
330	10.92	136.8	0.641	0.435	—	—	—	—
530	12.10	107.7	0.593	0.495	—	—	—	—

B Text Rendering at Matched Steps

To address potential concerns about different numbers of gradient updates at evaluation, we also compare Show-o2$_{+\text{VPE}}$ and Show-o2* at the same checkpoint step (72k). Because the SigLIP 2 tokens increase the per-sample sequence length, Show-o2$_{+\text{VPE}}$ uses a smaller micro batch size (8 vs. 12) to maintain comparable per-step token throughput ($8 \times 1{,}856 = 14{,}848$ vs. $12 \times 1{,}280 = 15{,}360$ tokens/step, $< 3.5\%$ difference). This means that at the same step count, Show-o2$_{+\text{VPE}}$ has seen fewer total images and fewer image tokens. Despite this data disadvantage, it still achieves substantially higher OCR accuracy (Table 8), confirming that the improvement stems from VPE.

Table 8: **Text rendering at matched steps (72k).** All results at CFG=5.5. $*$: continued training on 6.35M text rendering data.

	TextScenesHQ				TextVisionBlend				StyleTextSynth		
Method	CS↑	Acc↑	F1↑	CER↓	CS↑	Acc↑	F1↑	CER↓	Acc↑	F1↑	CER↓
Show-o2* (72k)	**0.29**	2.62	4.36	0.88	0.16	76.45	77.40	0.22	21.83	27.72	**0.59**
Show-o2$_{+\text{VPE}}$ (ours, 72k)	0.26	**12.10**	**20.66**	**0.82**	0.16	**79.83**	**82.34**	**0.20**	**26.73**	**37.05**	0.67

C Internal vs. External Training Details

SigLIP 2 encoder. We use the SigLIP 2-Giant-Patch16-384 model [8] as the visual semantic encoder. Target images are resized to 384×384 with white padding to extract $24 \times 24 = 576$ SigLIP 2 tokens. The resulting tokens are then discretized using the frozen SigLIP-VQ tokenizer from X-Omni [6], which adopts a codebook of 16,384 entries.

Table 9: **Training hyperparameters for each stage.** Stage 1 is shared, Stages 2 and 3 differ between internal and external architectures. "CE only" indicates that only cross-entropy loss on SigLIP 2 tokens is used (no flow matching loss). "Flow only" indicates that only flow matching loss is used (no cross-entropy loss).

		Stage 2 (T2I)		Stage 3 (Editing)	
	Stage 1	Internal	External	Internal	External
GPUs	16 H200	24 H200	24 H200	24 H200	24 H200
Data	39M	39M	39M	5.4M	5.4M
Epochs	4	2	2	$\sim$1.5	$\sim$1.5
BS / GPU	30	8	8	5	5
Grad Accum	1	1	1	2	2
LR	10^{-4}	10^{-4}	10^{-4}	5×10^{-5}	5×10^{-5}
LR Schedule	const+warm	const+warm	const+warm	const+warm	const+warm
EMA	—	0.9995	0.9995	0.9995	0.9995
$\lambda_{\text{CE}} : \lambda_{\text{flow}}$	CE only	0.03$\rightarrow$0.1:1.0	Flow only	0.1:1.0	CE only
Frozen	DiT	—	AR	—	DiT

Stage 1: SigLIP 2 alignment. Both architectures share the same Stage-1 checkpoint. The model learns to autoregressively predict SigLIP 2 tokens given text prompts, using only cross-entropy loss. The DiT and all image-related modules are frozen, only the AR (LLM) is trained. We train for 4 epochs on 39M mixed data (32M T2I + 6.35M text rendering) with constant LR 10^{-4} and 2,000 warmup steps.

Stage 2: T2I training. Both Internal$_{+\text{VPE}}$ and External$_{+\text{VPE}}$ load the Stage-1 checkpoint and are trained on 39M mixed data for 2 epochs. The two training strategies follow the standard pipelines of their respective architectures. Internal$_{+\text{VPE}}$: all parameters are unfrozen and jointly trained following the Transfusion-style training paradigm. For Internal$_{+\text{VPE}}$, the CE loss coefficient λ_{CE} is progressively increased from 0.03 to 0.1 during training to balance the CE and flow matching objectives. External$_{+\text{VPE}}$: the AR is frozen throughout and only the DiT is trained for the full 2 epochs, following the standard external paradigm [6, 7] where the AR serves as a fixed semantic encoder, so no loss balancing is needed.

Stage 3: Editing. Both models load their respective Stage-2 T2I checkpoints and add a reference image encoder initialized from Show-o2-1.5B [2] weights, increasing parameters from 4.16B to 4.57B. Both are trained on 5.4M editing pairs for approximately 1.5 epochs (70k steps) with the same data, batch size, and training duration.

Internal$_{+VPE}$: all parameters are unfrozen. The model receives the reference image, edit instruction, and SigLIP 2 tokens within a single sequence, and is trained with both cross-entropy loss ($\lambda_{\text{CE}} = 0.1$) and flow matching loss ($\lambda_{\text{flow}} = 1.0$). We adopt InstructPix2Pix-style [30] conditional dropout: with 5% probability drop the reference image, with 5% probability drop text and SigLIP 2 tokens jointly, and with 5% probability drop all conditions. Text and SigLIP 2 tokens are coupled because both serve as semantic prompts for the generation process (details in Appendix D).

External$_{+VPE}$: the DiT is frozen and only the AR model is trained. In the external architecture, the DiT always receives SigLIP 2 features as conditioning regardless of the task. For editing, the new capability the model needs to learn is how to incorporate the reference image information into the SigLIP 2 representation, which is handled entirely by the AR component. Therefore, only the AR is trained with cross-entropy loss. Conditional dropout uses the same three mutually exclusive 5% cases as the internal model (no reference, no text and SigLIP 2, or null) (details in Appendix D).

D Editing CFG Formulation

Internal model (Internal$_{+\text{VPE}}$). We denote the reference image tokens as $\mathbf{x}_{\text{ref}}$, the edit instruction as t, the SigLIP 2 visual prompt tokens as $\mathbf{s}$, and the generated image tokens as $\mathbf{x}_{\text{gen}}$. Special tokens include [BOS]/[EOS] (sequence boundaries), [BOI]/[EOI] (image boundaries), and [BOT]/[EOT] (visual prompt boundaries). Same notations apply to the external model. The internal model uses four input configurations during training, with the last three at 5% probability each:

$$\begin{aligned}
&\text{Full:} \quad [\text{BOS}][\text{BOI}]\ \mathbf{x}_{\text{ref}}\ [\text{EOI}]\ t\ [\text{BOT}]\ \mathbf{s}\ [\text{EOT}]\ [\text{BOI}]\ \mathbf{x}_{\text{gen}}\ [\text{EOI}][\text{EOS}]\\
&\text{No-text:} \quad [\text{BOS}][\text{BOI}]\ \mathbf{x}_{\text{ref}}\ [\text{EOI}]\ [\text{BOI}]\ \mathbf{x}_{\text{gen}}\ [\text{EOI}][\text{EOS}]\\
&\text{No-ref:} \quad [\text{BOS}]\ t\ [\text{BOT}]\ \mathbf{s}\ [\text{EOT}]\ [\text{BOI}]\ \mathbf{x}_{\text{gen}}\ [\text{EOI}][\text{EOS}]\\
&\text{Null:} \quad [\text{BOS}][\text{BOI}]\ \mathbf{x}_{\text{gen}}\ [\text{EOI}][\text{EOS}]
\end{aligned}$$

Note that text and SigLIP 2 tokens are always dropped *jointly* in the internal model, since they are semantically coupled through the AR model.

External model (External$_{+\text{VPE}}$). The external model's AR component uses four input configurations during editing training, with the last three at 5% probability each:

$$\begin{aligned}
&\text{Full:} \quad [\text{BOS}][\text{BOI}]\ \mathbf{x}_{\text{ref}}\ [\text{EOI}]\ t\ [\text{BOT}]\ \mathbf{s}\ [\text{EOT}][\text{EOS}]\\
&\text{No-text:} \quad [\text{BOS}][\text{BOI}]\ \mathbf{x}_{\text{ref}}\ [\text{EOI}][\text{EOS}]\\
&\text{No-ref:} \quad [\text{BOS}]\ t\ [\text{BOT}]\ \mathbf{s}\ [\text{EOT}][\text{EOS}]\\
&\text{Null:} \quad [\text{BOS}][\text{EOS}]
\end{aligned}$$

As in the internal model, the four cases are mutually exclusive (a single random draw selects one of them), and text and SigLIP 2 tokens are always dropped jointly. The DiT is conditioned on the AR hidden states at the SigLIP 2 positions. When these tokens are absent (No-text and Null), it receives an all-zero SigLIP 2 feature. During editing, only the AR component is trained (DiT remains frozen), since the AR effectively serves as a visual semantic encoder that translates the reference image and edit instruction into SigLIP 2 tokens for the DiT, following the standard external paradigm [6, 7].

Inference CFG. Both models generate the visual prompt $\mathbf{s}$ greedily, without guidance, and then combine three velocity predictions at each denoising step following InstructPix2Pix [30]:

$$\hat{\mathbf{v}} = \mathbf{v}_{\varnothing} + s_I\,(\mathbf{v}_{\text{ref}} - \mathbf{v}_{\varnothing}) + s_T\,(\mathbf{v}_{\text{full}} - \mathbf{v}_{\text{ref}}),$$

where $\mathbf{v}_{\text{full}}$, $\mathbf{v}_{\text{ref}}$, and $\mathbf{v}_{\varnothing}$ are predicted under the Full, No-text, and Null configurations, with $s_I{=}1.0$ and $s_T{=}2.5$. For the external model, the No-text and Null configurations both give the DiT an all-zero SigLIP 2 feature, so $\mathbf{v}_{\text{ref}}{=}\mathbf{v}_{\varnothing}$ and the rule reduces to standard two-way guidance $\mathbf{v}_{\varnothing} + s_T(\mathbf{v}_{\text{full}} - \mathbf{v}_{\varnothing})$.

E Training Loss Curves

We provide training loss curves for all experiments in Figure 5. Smooth curves (moving average) are overlaid on raw data (low opacity) for clarity.

F Additional Results

F.1 Training and inference efficiency

Table 10 reports training cost and Table 11 inference cost. C2I experiments run on H800 GPUs and text rendering on H200 GPUs. LlamaGen$_{+\text{VPE}}$ incurs an 18.9% per-step overhead from its longer sequences, but reaches the baseline's final FID (24.56) by epoch 50 ($\sim$904 H800 GPU-hours vs. 1,520), a 40% reduction in total compute to match quality. Under fixed 10,240-token packing, Transfusion$_{+\text{VPE}}$ is 20.6% cheaper per run, and Show-o2$_{+\text{VPE}}$ is compute-neutral ($< 2\%$ difference). Figure 6 plots FID against training GPU-hours. At inference, AR planning adds a one-time per-image cost of 2–8 s, depending on the number of visual prompt tokens (Table 11).

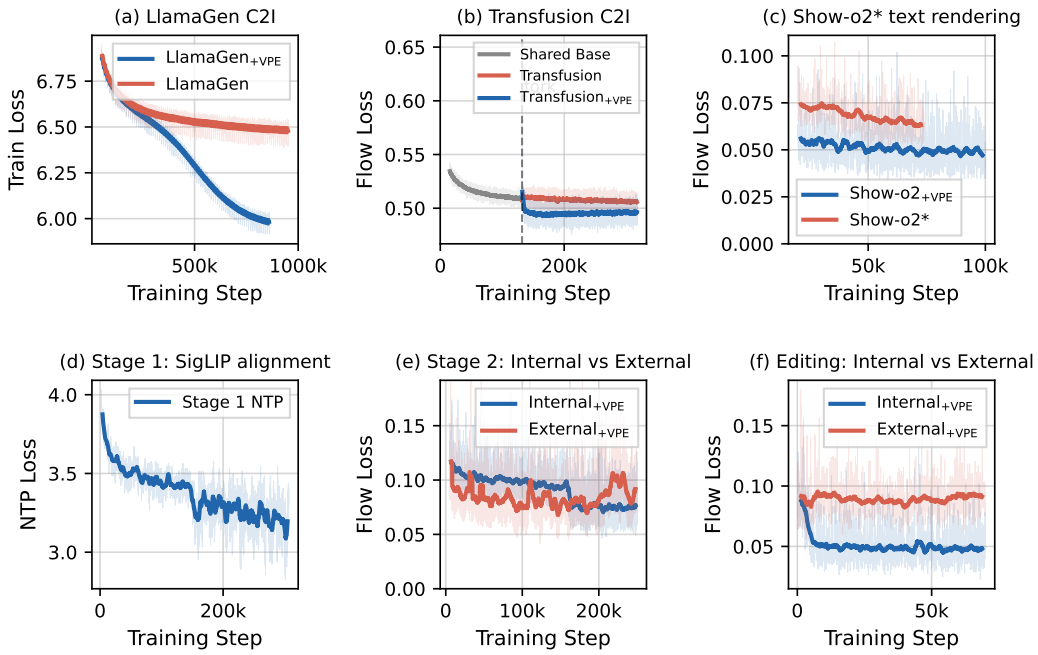

Figure 5: **Training loss curves.** *Top row:* VPE vs. baseline training loss for each framework. *(a)* LlamaGen$_{+\text{VPE}}$ converges to lower loss than LlamaGen. *(b)* Transfusion$_{+\text{VPE}}$ converges to lower flow loss than Transfusion after forking from the shared base. *(c)* Show-o2$_{+\text{VPE}}$ achieves lower flow loss than Show-o2* on text rendering. *Bottom row:* Internal vs. External architecture training. *(d)* Stage-1 SigLIP 2 alignment. *(e)* Stage-2: both architectures achieve similar flow loss. *(f)* Editing: Internal$_{+\text{VPE}}$ achieves lower flow loss than External$_{+\text{VPE}}$, consistent with its superior preservation metrics (Table 5).

Table 10: **Training cost.** Overhead is relative to the baseline of the same framework. For Transfusion, cost covers the 136k steps after the fork.

Setting	Seq. len	Global batch	Steps	GPU-hours	Overhead	Peak mem
LlamaGen	1,025	128	1,000,000	1,520 (H800)	—	37.7 GiB
LlamaGen$_{+\text{VPE}}$	1,221	128	1,000,000	1,808 (H800)	+18.9%	43.8 GiB
Transfusion	10,240 (packed)	32	136,000	2,016 (H800)	—	6.1 GiB
Transfusion$_{+\text{VPE}}$	10,240 (packed)	32	136,000	1,600 (H800)	−20.6%	4.51 GiB
Show-o2*	1,280	288	70,000	2,224 (H200)	—	120.8 GiB
Show-o2$_{+\text{VPE}}$	1,856	192	70,000	2,186 (H200)	$\sim -1.7\%$	120.4 GiB

F.2 Internal$_{+\text{VPE}}$ inference procedure

Algorithm 1 summarizes inference for Internal$_{+\text{VPE}}$ (Section 2.3), taking image editing as the example. Inference runs in two steps: the AR expert first decodes the visual prompt with a KV cache, and the DiT tokens then attend to the AR keys/values and to themselves at every denoising step. For text-to-image generation, the procedure is identical except that there is no reference image $\mathbf{x}_{\text{ref}}$ in the input. Classifier-free guidance is omitted from the algorithm for clarity.

F.3 Progressive schedule ablation

Table 12 compares the progressive schedules of Section 2.2 with fixed alternatives. The progressive schedule gives the best final FID in both frameworks. For LlamaGen, fully exposed prompts perform worst early (23.10 at epoch 50): the model relies heavily on SigLIP 2 tokens while its planning ability is still poor. The gap narrows as planning improves (10.42 at epoch 80), consistent with the shrinking ground-truth gap in Table 13. Fixed 0.5 dropout is more robust early (20.41 at epoch 50) because the

Table 11: **Inference cost** (bs=1, 50 timesteps).

Setting	AR planning	Generation	Peak memory
LlamaGen	—	16.02 s	2.71 GiB
LlamaGen$_{+\text{VPE}}$	196 tokens / 3.04 s	16.34 s	2.99 GiB
Transfusion	—	1.21 s	2.50 GiB
Transfusion$_{+\text{VPE}}$	121 tokens / 2.40 s	1.17 s	2.51 GiB
Show-o2*	—	3.93 s	7.69 GiB
Show-o2$_{+\text{VPE}}$	576 tokens / 8.28 s	4.06 s	7.80 GiB

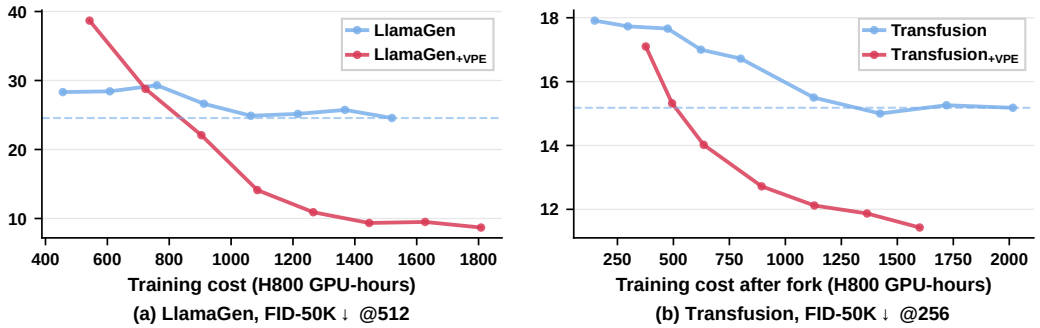

Figure 6: **Quality vs. training compute.** FID-50K against cumulative training GPU-hours. Dashed lines mark the baseline's final FID. For Transfusion, cost is counted from the 130k-step fork.

model keeps a direct conditioning pathway, but it under-uses the semantic plan later and caps the final quality. The progressive schedule balances the two: early masking builds robustness to imperfect plans, and late unmasking lets the model fully use the plan. For Transfusion, a fixed λ_{CE}=0.1 gives the worst early FID among the loss-weighting variants (17.81 at 162k), as the CE gradient dominates flow-matching learning. A fixed λ_{CE}=0.03 dilutes the CE signal in mid-training and limits plan quality. Token dropout with λ_{CE}=0.1 gives the worst final FID (15.68), as the masking strategy that works for discrete tokens is unsuitable for continuous sequences.

Table 12: **Progressive schedule ablation** (FID-50K↓, no CFG). All variants share the same visual prompts, data, and compute, and differ only in the schedule.

(a) LlamaGen (p_{mask})

Ep	Base	Prog. 0.95→0.05	Fully exposed	Fixed 0.5
50	29.31	22.07	23.10	**20.41**
60	26.63	**14.12**	16.01	16.92
70	24.89	**10.91**	11.14	13.81
80	25.16	**9.35**	10.42	10.26

(b) Transfusion (λ_{CE})

Step	Base	Prog. 0.03→0.1	Fixed 0.1	Fixed 0.03	0.1 + tok. drop
162k	17.66	**17.10**	17.81	17.21	23.42
184k	16.72	**14.02**	14.41	15.45	19.37
226k	15.00	**12.12**	14.66	13.13	16.70
266k	15.18	**11.43**	14.78	12.49	15.68

F.4 Ground-truth vs. predicted visual prompts

We teacher-force the ground-truth SigLIP-VQ tokens of ImageNet validation images at inference to measure the gap between predicted and oracle plans (Table 13). Ground-truth prompts consistently lower FID in both frameworks, and the gap shrinks as training proceeds (LlamaGen: -7.62 at epoch 50 to -5.02 at epoch 80, and Transfusion: -7.46 at 162k to -3.56 at 266k). The gain of VPE therefore comes from the semantic plan and grows with the quality of the predicted plan.

F.5 Controlled discrete vs. continuous comparison

Our main C2I experiments validate VPE within matched pairs in two different frameworks, so they do not compare token types directly. We therefore run a controlled experiment in the same

Algorithm 1 Inference procedure of Internal$_{+\text{VPE}}$ for image editing.

$\mathbf{s} \leftarrow \text{AR.GENERATE}([\mathbf{x}_{\text{ref}}, t])$ ▷ Step A: causal decoding with KV cache
$\mathbf{KV}_{\text{AR}} \leftarrow$ keys/values of $[\mathbf{x}_{\text{ref}}, t, \mathbf{s}]$ at every layer ▷ fixed during Step B
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $k = 1, \ldots, K$ **do** ▷ Step B: denoising
  $\mathbf{v} \leftarrow \text{DiT}(\mathbf{z}, \tau_k;\, \mathbf{KV}_{\text{AR}})$ ▷ DiT queries attend to $[\mathbf{KV}_{\text{AR}}; \mathbf{KV}_{\text{DiT}}]$ (Eq. 4)
  $\mathbf{z} \leftarrow \mathbf{z} + (\tau_{k+1} - \tau_k)\, \mathbf{v}$
end for
return $\text{VAE.DECODE}(\mathbf{z})$

Table 13: **Ground-truth vs. predicted visual prompts** (FID-50K↓, no CFG).

	Baseline	VPE (predicted)	VPE (GT)	Δ (GT − pred)
LlamaGen				
Epoch 50	29.31	22.07	14.45	−7.62
Epoch 60	26.63	14.12	7.28	−6.84
Epoch 70	24.89	10.91	5.57	−5.34
Epoch 80	25.16	9.35	4.33	−5.02
Transfusion				
162k	17.66	17.10	9.64	−7.46
184k	16.72	14.02	9.98	−4.04
226k	15.00	12.12	8.30	−3.82
266k	15.18	11.43	7.87	−3.56

Transfusion framework, switching only the image token type. Continuous-specific modules (timestep embedding, latent projection) are replaced by a discrete codebook embedding, giving comparable sizes (continuous 0.37B/0.39B, discrete 0.38B/0.40B for baseline/VPE). The discrete variant uses the LlamaGen VQ-VAE [15] ($16\times$ downsampling, codebook 16,384, rFID 2.19, PSNR 20.79 dB) and the continuous variant the FLUX VAE (rFID 0.13, PSNR 30.20 dB), both with 256 tokens at 256×256. Both are initialized from Qwen2.5-0.5B-Instruct (excluding embeddings), use packed sequences of 10,240 tokens, and train 130k shared steps plus 136k branched steps, with progressive masking (discrete) or progressive loss weights (continuous). VPE improves both token types (Table 14): discrete FID drops from 29.53 to 16.08 (-45%) and continuous from 15.18 to 11.43 (-25%), and the discrete–continuous gap shrinks from 14.35 to 4.65 FID.

Table 14: **Controlled discrete vs. continuous comparison** (FID-50K↓, ImageNet 256×256, same Transfusion framework).

Step	Continuous	Continuous$_{+\text{VPE}}$	Discrete	Discrete$_{+\text{VPE}}$
72k	29.05	—	40.03	—
96k	22.07	—	36.46	—
130k	18.26	—	32.98	—
162k	17.66	17.10	31.32	30.38
184k	16.72	14.02	29.58	19.59
226k	15.00	12.12	30.46	16.78
266k	15.18	**11.43**	29.53	**16.08**

F.6 Additional text-to-image evaluations

Table 15(a) reports GenAI-Bench [49] VQAScore. Both models perform on par with T2I-R1 [51] (basic/advanced overall: Internal$_{+\text{VPE}}$ 0.89/0.72, External$_{+\text{VPE}}$ 0.88/0.73, T2I-R1 0.88/0.73) and score above SD-XL [55] and Midjourney v6. On DSG-1k [50], Internal$_{+\text{VPE}}$ and External$_{+\text{VPE}}$ score 77.8 and 78.1 overall. Table 15(b) repeats GenEval with 3 seeds. Table 16 reports the LAION aesthetic score [56] on TextAtlas. VPE does not trade visual quality for textual fidelity, as the aesthetic score improves on all three subsets (*e.g.*, 4.08→4.37 on StyleTextSynth). The slightly lower CLIP score of Show-o2$_{+\text{VPE}}$ in Table 3 is likely an artifact of CLIP's 77-token text limit, which truncates the long TextAtlas prompts and is insensitive to whether the rendered text is correct.

Table 15: **Additional T2I benchmarks.** (a) GenAI-Bench [49] VQAScore, with baseline rows from the GenAI-Bench paper and T2I-R1 [51]. (b) GenEval with 3 seeds (mean±std).

(a) GenAI-Bench

	Basic						Advanced					
Method	Attr.	Scene	Spat.	Act.	Part	Overall	Count	Diff.	Comp.	Neg.	Univ.	Overall
SD-XL [55]	0.84	0.84	0.82	0.83	0.89	0.83	0.71	0.73	0.69	0.50	0.66	0.63
Midjourney v6	0.88	0.87	0.87	0.87	0.91	0.87	0.78	0.78	0.79	0.50	0.76	0.69
T2I-R1 [51]	0.87	0.89	0.89	0.87	0.87	0.88	0.81	0.82	0.78	0.60	0.73	0.73
Internal$_{+\text{VPE}}$	0.89	0.91	0.90	0.89	0.88	0.89	0.79	0.80	0.77	0.58	0.76	0.72
External$_{+\text{VPE}}$	0.88	0.89	0.89	0.87	0.88	0.88	0.79	0.81	0.77	0.59	0.76	0.73

(b) GenEval, 3 seeds

Method	Single	Two	Count	Color	Pos.	C.Attr.	Overall
Internal$_{+\text{VPE}}$	0.98±0.01	0.91±0.02	0.68±0.02	0.86±0.02	0.76±0.02	0.55±0.03	0.79±0.01
External$_{+\text{VPE}}$	0.97±0.01	0.89±0.02	0.67±0.03	0.85±0.02	0.74±0.03	0.66±0.03	0.80±0.01

Table 16: **Aesthetic quality on TextAtlas** (LAION aesthetic score [56]↑).

Method	StyleTextSynth	TextScenesHQ	TextVisionBlend
Ground truth	4.72	4.72	3.87
Show-o2*	4.08	4.01	3.40
Show-o2$_{+\text{VPE}}$ (ours)	**4.37**	**4.23**	**3.70**

F.7 Additional editing evaluation

PIE-Bench focuses on preservation, so we also evaluate on I2EBench [54] with the original instructions (Table 17). Internal$_{+\text{VPE}}$ is better on all low-level dimensions and comparable on high-level ones, consistent with an advantage specific to fine-grained detail preservation.

Table 17: **I2EBench results** (higher is better).

(a) Low-level editing

Task	Internal	External
Deblurring	**0.345**	0.333
HazeRemoval	**0.537**	0.438
Lowlight	**0.240**	0.107
RainRemoval	**0.583**	0.460
ShadowRemoval	**0.362**	0.353
SnowRemoval	**0.449**	0.439
WatermarkRemoval	**0.499**	0.424
RegionAccuracy	**0.536**	0.509

(b) High-level editing

Task	Internal	External
Counting	**0.130**	0.082
DirectionPerception	**0.471**	0.434
ObjectRemoval	0.446	0.446
Replacement	0.294	**0.331**
BGReplacement	0.496	**0.525**
ColorAlteration	0.246	**0.254**

G Limitations

Scale and design choices. VPE has not been validated on larger-scale models, and scaling behavior remains future work. We study only SigLIP 2 as the visual semantic representation, and other semantic representations (CLIP, DINOv2) may offer different trade-offs. The progressive training schedule introduces hyperparameters ($p_0, p_1, k, \lambda_0, \lambda_1$) that we have not extensively tuned. Better schedules may yield further improvements.

Scope of the comparisons. Our C2I experiments validate VPE within matched pairs in different frameworks. Only the experiment in Appendix F.5 compares discrete and continuous tokens under a single framework. The internal-vs-external comparison covers the evaluated configurations: the editing ablations (Table 5b) show that the preservation gap stems from semantic-only DiT conditioning, and other external designs that inject reference features can largely close it. Our T2I and editing

evaluations cover GenEval, GenAI-Bench, DSG-1k, TextAtlas, PIE-Bench, and I2EBench, but not human studies.

Failure cases. We observe two failure modes. (1) When the generated SigLIP 2 tokens deviate from the intended semantics (*e.g.*, rare categories with limited training data), the rendering stage inherits the error. (2) In editing, complex spatial rearrangements (*e.g.*, "move the cat's right leg onto its left leg") remain challenging, as SigLIP 2 tokens capture semantic content rather than precise spatial transformations.

H Broader Impact

This work improves the quality and efficiency of image generation models, which carries both positive and negative societal implications. On the positive side, more efficient training reduces computational cost and energy consumption, making image generation research more accessible. On the negative side, higher-quality image generation could be misused to create misleading or harmful visual content such as deepfakes. We note that VPE is a general training technique applied to existing open-source architectures and does not introduce new risks beyond those already present in the underlying models. We encourage the community to develop and deploy appropriate safeguards, including watermarking and content provenance tools, alongside advances in generation quality.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Appendix G.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: N/A

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Training details, hyperparameters, data sources, and evaluation protocols are provided in Sections 4 and Appendix C.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: See project page.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Full details are provided in Section 4 and Appendix C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: N/A

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Section 4 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: N/A

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See Appendix H.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: N/A

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: N/A

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: N/A

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: N/A

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: N/A

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We use LLM as our backbone model. See Section 4 and Appendix C.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.