RaPD: Resolution-Agnostic Pixel Diffusion via Semantics-Enriched Implicit Representations

Yanhao Ge[1] **Shanyan Guan**[2] **Weihao Wang**[1,2] **Ying Tai**[3] **Mingyu You**[1,4*]
[1] College of Electronic and Information Engineering, Tongji University
[2] vivo Mobile Communication Co., Ltd. [3] Nanjing University
[4]State Key Laboratory of Autonomous Intelligent Unmanned Systems,
Frontiers Science Center for Intelligent Autonomous Systems, Ministry of Education

Abstract

Natural images are continuous, yet most generative models synthesize them on discrete grids, limiting resolution-flexible generation. Continuous neural fields enable resolution-free rendering, but prior methods introduce continuity only at the decoding stage as an interpolation module, leaving the generative latent space discretized and reconstruction-oriented. We propose **RaPD** (**R**esolution-**a**gnostic **P**ixel **D**iffusion), which performs diffusion in a continuous Neural Image Field (NIF) latent space. RaPD bridges this *reconstruction–generation gap* with *Semantic Representation Guidance* for generation-aware latent learning and a *Coordinate-Queried Attention Renderer* for coordinate-conditioned, scale-aware rendering. A single denoised latent can be rendered at arbitrary resolutions by changing only the query coordinates, keeping diffusion cost fixed. Experiments demonstrate superior generation quality and resolution scalability.

1 Introduction

The visual world is inherently continuous, yet modern image generative models mostly operate on spatially discretized representations. Whether in VAE latent or pixel space [58, 19, 41, 29, 56], their architectures and computation are tied to discrete grids, making higher resolutions require more tokens and compute [84]. As a result, native generation at arbitrary, unseen resolutions remains challenging. In this work, we ask whether image generation can instead be formulated over "continuous pixels", rather than discrete pixels or compressed VAE proxies.

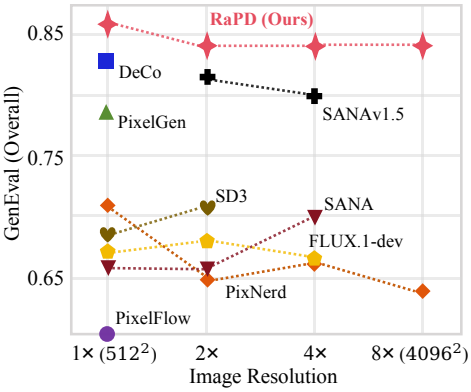


Figure 1: RaPD supports arbitrary-resolution text-to-image generation, achieves competitive GenEval scores against strong baselines. [19, 41, 53, 52, 69, 10]

Existing arbitrary-resolution generation models typically introduce continuity only at the rendering stage. GAN models use coordinate-queried MLPs but suffer from limited generation capacity [1, 64, 4]. Latent diffusion-based models use flexible decoders [50, 71, 39, 12, 25, 72], yet diffusion itself remains in a discrete space, which ties semantics to a discrete representation and inherits the VAE bottleneck [40, 8, 78, 43, 79, 74]. Pixel diffusion removes the VAE [10, 80, 13, 52, 44, 53, 15, 42, 14], but still generates on discrete grids. Even PixNerd [69], the closest attempt, uses a patch-wise neural

*Mingyu You is the corresponding author with email: myyou@tongji.edu.cn.

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

field mainly as an interpolation module over fixed-resolution features, rather than learning a latent that models continuous image signals for generation. This motivates a more fundamental question: can we directly generate continuous image signals?

A straightforward solution is to diffuse existing continuous image representations. As shown in Fig. 2, however, this yields poor results. The bottleneck lies in the representation itself: methods such as LIIF [11] and CLIF [12] are built for interpolation, with reconstruction training that favors local appearance cues and lightweight decoders with limited semantic aggregation. Thus, continuous representations must be made generation-aware to support high-quality arbitrary-resolution synthesis.

To address this gap, we propose **RaPD**, a framework for directly generating continuous image signals. RaPD performs pixel-space diffusion in a continuous Neural Image Field (NIF) latent space, rather than treating neural fields merely as decoding-time interpolation modules. Since existing continuous image representations are reconstruction-oriented, RaPD first reshapes the NIF latent into a generation-aware representation through *Semantic Representation Guidance* (SRG), which injects semantic structure from a vision foundation model and adapts the timestep schedule to high-density continuous latents. To decode such semantically enriched latents, RaPD further introduces a *Coordinate-Queried Attention Renderer* (CQAR), which replaces local implicit decoders with coordinate-conditioned, scale-aware attention for inter-pixel semantic reasoning. At inference, a single denoised NIF latent can be rendered at arbitrary target resolutions by varying only the query coordinate grid, while the diffusion cost remains independent of output size (shown in Fig. 1).

Our contributions are summarized as follows:

1. We propose RaPD, a resolution-agnostic pixel diffusion framework that directly generates continuous image signals in an NIF latent space.
2. We introduce Semantic Representation Guidance to transform reconstruction-oriented continuous latents into generation-aware representations.
3. We design a Coordinate-Queried Attention Renderer for spatially coherent arbitrary-resolution rendering with fixed diffusion cost.

2 Preliminaries

Flow Matching. Flow matching [48, 19] is a widely used training paradigm for diffusion models [65, 29, 67]. It learns a velocity field that transports Gaussian noise to the data distribution. To avoid overloading image notation, we denote a generic flow state by $\boldsymbol{u}_t$. For a data sample $\boldsymbol{u}_1$, a common linear interpolation path is

$$\boldsymbol{u}_t = (1-t)\boldsymbol{\epsilon} + t\boldsymbol{u}_1, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{1}$$

where $t \in [0,1]$. Under this convention, $\boldsymbol{u}_t$ evolves from Gaussian noise at $t=0$ to data at $t=1$, and the target velocity is $\boldsymbol{v}^\star = \boldsymbol{u}_1 - \boldsymbol{\epsilon}$. For image generation, $\boldsymbol{u}_1$ is typically instantiated as a spatially discretized representation, such as VAE latents [58, 57] or image patch tokens [10, 80, 13, 44, 53, 15, 42]. Despite their strong performance, these representations tie the learned velocity field to finite spatial grids and token lengths, which can limit generalization to unseen resolutions [3, 26, 18].

Neural Image Fields. We use $\boldsymbol{x}$ to denote an RGB image and $\boldsymbol{z}$ to denote its latent feature representation. Beyond discrete image grids, an image can be represented as a continuous coordinate-based function, commonly referred to as a Neural Image Field (NIF) [63, 54]. Representative methods such as LIIF [11] and CLIF [12] parameterize an image by a feature map $\boldsymbol{z} \in \mathbb{R}^{C\times H\times W}$, and render RGB values by querying continuous coordinates:

$$\hat{\boldsymbol{x}}(\boldsymbol{q}) = f_\psi\big(\boldsymbol{z}; \boldsymbol{q}, \Delta\boldsymbol{c}\big), \tag{2}$$

where $\boldsymbol{q} \in [-1,1]^2$ is a target coordinate, $\Delta\boldsymbol{c}$ is the cell-size descriptor associated with the target pixel, and f_ψ denotes a coordinate renderer that queries features from $\boldsymbol{z}$ and predicts the RGB value at $\boldsymbol{q}$. By evaluating Eq. (2) over different coordinate grids, the same representation can be rendered at arbitrary resolutions. Existing NIFs are typically trained for interpolation or reconstruction. Given a target RGB image $\boldsymbol{x}_{\mathrm{tar}}$ defined on a coordinate grid $\mathcal{Q}_{\mathrm{tar}}$, the renderer produces $\hat{\boldsymbol{x}}_{\mathrm{tar}} = \{f_\psi(\boldsymbol{z}; \boldsymbol{q}, \Delta\boldsymbol{c}(\boldsymbol{q})) \mid \boldsymbol{q} \in \mathcal{Q}_{\mathrm{tar}}\}$. The reconstruction objective is

$$\mathcal{L}_{\mathrm{rec}} = \|\hat{\boldsymbol{x}}_{\mathrm{tar}} - \boldsymbol{x}_{\mathrm{tar}}\|_1 + \omega\mathcal{L}_{\mathrm{LPIPS}}\big(\hat{\boldsymbol{x}}_{\mathrm{tar}}, \boldsymbol{x}_{\mathrm{tar}}\big), \tag{3}$$

where ω is the LPIPS loss weight. This reconstruction-oriented objective makes NIFs effective interpolators, but their latents are not directly generation-ready.

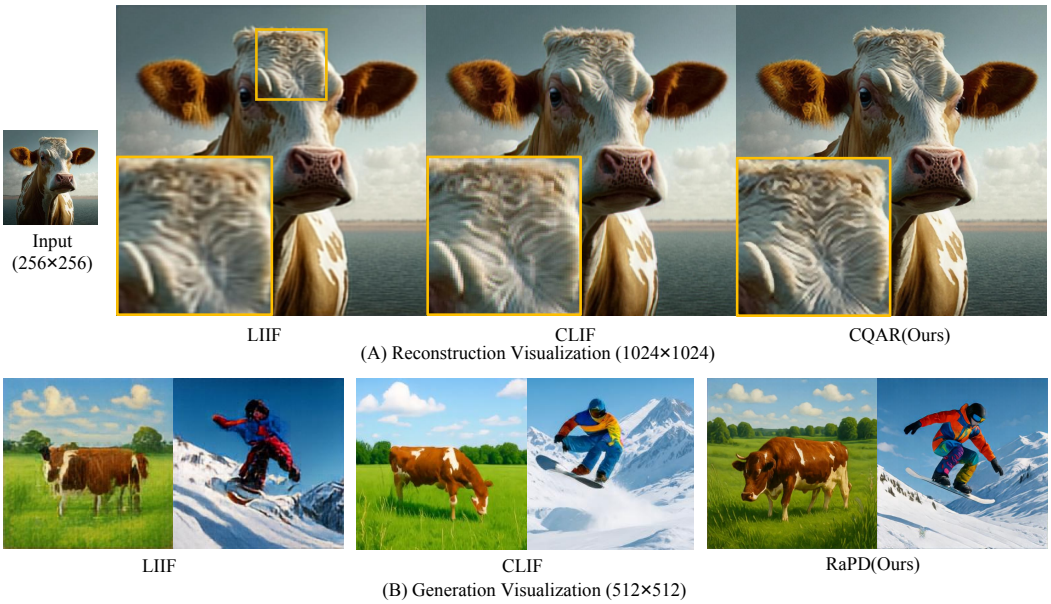


Figure 2: Reconstruction–generation gap of existing NIF representations: strong super-resolution reconstruction (top) but weak text-to-image generation (bottom).

3 Resolution-Agnostic Pixel Diffusion

We present **RaPD**, a framework for text-to-image generation over continuous NIF latents. RaPD learns semantically guided NIF representations for generation and denoises their latent feature maps with flow matching. The denoised latent is then rendered at arbitrary target resolutions by querying a coordinate-conditioned renderer, keeping the diffusion cost independent of output size.

A straightforward baseline is to apply a PixelGen-style DiT denoiser [56, 53] to existing NIF latents, such as LIIF [11] and CLIF [12]. As shown in Fig. 2, these latents reconstruct well but generate poorly, exposing a *reconstruction–generation gap*. RaPD addresses this gap with two components: *Semantic Representation Guidance* (SRG), which makes NIF latents generation-aware, and *Coordinate-Queried Attention Renderer* (CQAR), which renders queried coordinates with inter-pixel semantic reasoning.

3.1 Basic Framework

As shown in Fig. 3, RaPD adopts a two-stage design. Stage-1 learns a generation-aware NIF latent. Given an encoder input image $\boldsymbol{x}_{\text{in}}$, an encoder E_ϕ maps it to a dense latent $\boldsymbol{z} = E_\phi(\boldsymbol{x}_{\text{in}}) \in \mathbb{R}^{C\times H\times W}$ without spatial downsampling. Given a target coordinate $\boldsymbol{q}$, CQAR renders the corresponding RGB value as

$$\hat{\boldsymbol{x}}(\boldsymbol{q}) = f_\psi^{\text{CQAR}}\big(\boldsymbol{z}; \boldsymbol{q}, \Delta\boldsymbol{c}\big), \tag{4}$$

where f_ψ^{CQAR} queries local features from $\boldsymbol{z}$ at coordinate $\boldsymbol{q}$, and $\Delta\boldsymbol{c}$ is the cell-size used by CQAR. An image at resolution $H' \times W'$ is rendered by evaluating Eq. (4) over all target coordinates. Unlike conventional NIFs trained mainly for interpolation, RaPD trains E_ϕ and f_ψ^{CQAR} with SRG so that $\boldsymbol{z}$ preserves coordinate-rendering capability while acquiring semantic structure useful for generation.

We instantiate E_ϕ with an EDSR-style encoder [47, 11] without spatial downsampling, producing a dense latent at the encoder-input resolution. In Stage-2, E_ϕ and f_ψ^{CQAR} are frozen. We compute per-channel statistics $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$ over the training set and train a DiT-style denoiser D_θ, following the PixelGen design [53], over normalized NIF latents $\tilde{\boldsymbol{z}} = (\boldsymbol{z} - \boldsymbol{\mu}_z)/\boldsymbol{\sigma}_z$. The denoiser is conditioned on frozen Qwen3-1.7B [76] text embeddings $\boldsymbol{y}_{\text{cond}} \in \mathbb{R}^{128\times 2048}$ and trained to predict the clean normalized latent, which is converted to a velocity for the flow-matching loss. At inference, integrating the learned velocity field produces a generated normalized latent $\tilde{\boldsymbol{z}}_{\text{gen}}$, which is denormalized and rendered by CQAR at any requested resolution.

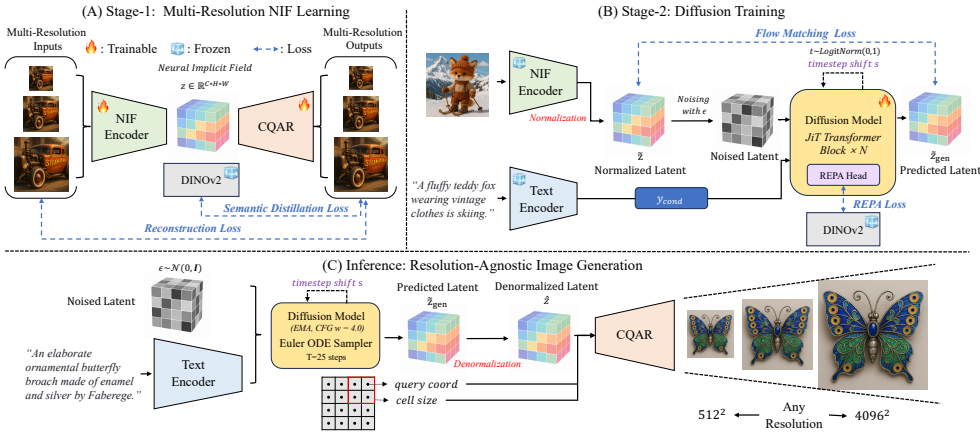


Figure 3: Overview of RaPD. (A) Stage-1: semantically guided multi-resolution NIF learning. (B) Stage-2: text-conditioned flow matching in normalized NIF latents. (C) Inference: render one denoised latent at arbitrary resolutions via CQAR.

3.2 Semantic Representation Guidance

Existing NIF latents are optimized for interpolation and reconstruction, making them poorly suited for generative modeling. We propose *Semantic Representation Guidance* (SRG), a unified strategy that makes NIF latents generation-aware at both the representation and diffusion-training levels. SRG consists of two components: *Semantic Distillation*, which injects foundation-model semantics into the dense NIF latent during Stage-1, and *Latent-Density-Aware Timestep Shift*, which adapts Stage-2 flow matching to the high-density SRG-shaped latent.

Semantic Distillation. How can a NIF latent be made generation-aware while retaining its ability to render continuous coordinates? We argue that the latent should not be learned from pixel reconstruction alone, but should also inherit semantic structure from a vision foundation model. We adopt a frozen DINOv2 ViT-B/14 [55] as the semantic teacher, producing patch features $\boldsymbol{t} \in \mathbb{R}^{768\times H_p\times W_p}$. Since the NIF latent $\boldsymbol{z}$ has no spatial downsampling and a different channel dimension, we pool $\boldsymbol{z}$ to the DINOv2 patch grid and project it with 1×1 convolution, yielding student features $\boldsymbol{z}_{\text{proj}} \in \mathbb{R}^{768\times H_p\times W_p}$. This projection branch is used only for distillation; the original NIF latent keeps its spatial layout and channel dimension for coordinate-based rendering.

The alignment loss comprises two terms that capture both local semantic content and global relational structure [79, 78, 74]. First, a cosine loss enforces per-position semantic agreement up to a margin:

$$\mathcal{L}_{\text{mcos}} = \frac{1}{N}\sum_{i=1}^{N} \max\big(0,\, 1 - m_{\cos} - \cos(\boldsymbol{z}_{\text{proj}}^{(i)}, \boldsymbol{t}^{(i)})\big), \tag{5}$$

where $N = H_p \times W_p$. Second, a distance-matrix similarity loss preserves the teacher's pairwise relational geometry:

$$\mathcal{L}_{\text{mdms}} = \frac{1}{K^2}\sum_{i,j} \max\big(0,\, |\boldsymbol{D}_z[i,j] - \boldsymbol{D}_t[i,j]| - m_{\text{dist}}\big), \tag{6}$$

where $\boldsymbol{D}_z$ and $\boldsymbol{D}_t$ are cosine-similarity matrices computed over K sampled spatial positions from $\boldsymbol{z}_{\text{proj}}$ and $\boldsymbol{t}$, respectively. The final semantic distillation loss is $\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{mcos}} + \mathcal{L}_{\text{mdms}}$. We use $m_{\cos} = 0.5$, $m_{\text{dist}} = 0.25$, and $K = 256$.

Unlike prior vision-feature alignment applied to downsampled VAE latents or denoiser features [79, 78, 74], our alignment supervises a dense NIF latent directly. The projection branch transfers semantic structure without changing the latent layout, while the margin-based losses preserve compatibility with pixel-level reconstruction. Thus, semantic alignment acts here as formative supervision for a generation-aware continuous representation, rather than an auxiliary regularizer on an existing latent.

Latent-Density-Aware Timestep Shift. Semantic Distillation shapes the NIF latent, but Stage-2 must still learn a flow over this unusually dense representation. Unlike compressed VAE latents, the

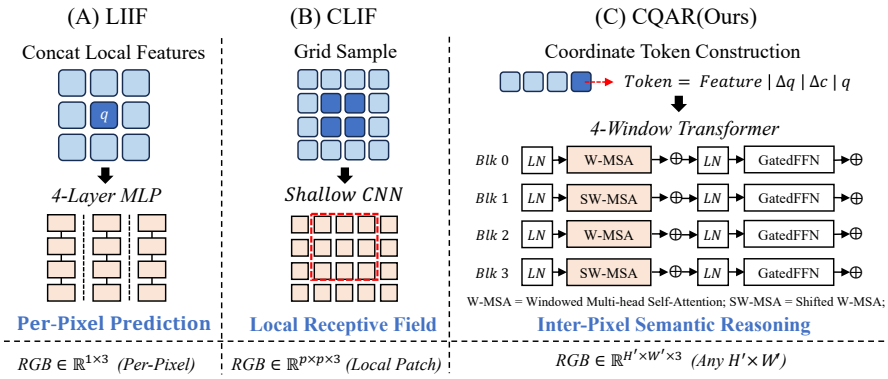


Figure 4: Implicit decoder comparison. (A) LIIF: per-pixel MLP. (B) CLIF: shallow CNN. (C) CQAR: coordinate-conditioned windowed attention over output tokens.

NIF latent has no spatial downsampling and retains a high-resolution coordinate-rendering layout. In flow matching, the timestep shift factor s controls the sampled timestep distribution:

$$t_0 = \mathrm{sigmoid}(\xi), \quad \xi \sim \mathcal{N}(0,1), \qquad t = \frac{t_0}{t_0 + (1-t_0)\cdot s}. \tag{7}$$

A larger s helps the model prioritize heavy noise stages (small t). These stages are the most critical for generating decent content. Standard VAE-latent diffusion models typically use small shift factors, with $s \in [1, 3]$, since their latents are spatially compressed. For NIF latents, we follow the resolution-dependent shift principle [19] and adopt a larger shift to compensate for the increased information density of the latent. As analyzed in Sec. 4.2, both insufficient and excessive shifts degrade generation quality; we empirically set $s = 32$ in RaPD. Together, Semantic Distillation and Latent-Density-Aware Timestep Shift form SRG: the former shapes the latent into a semantic representation, while the latter adapts diffusion training to recover this representation during generation.

3.3 Coordinate-Queried Attention Renderer

While SRG provides the NIF latent with semantic structure, the renderer must exploit this structure during coordinate-based decoding. Existing implicit decoders such as LIIF [11] and CLIF [12] predict pixels from local latent features using pointwise MLPs or shallow CNNs. These local operators are effective for interpolation, but cannot reason across queried pixels and therefore underuse the semantic features injected by SRG.

We address this limitation with the *Coordinate-Queried Attention Renderer* (CQAR), shown in Fig. 4. The key distinction of CQAR is that attention is performed over queried output-coordinate tokens, rather than over the input latent grid. For a target resolution $H' \times W'$, CQAR constructs one token per output coordinate and processes these tokens with Swin-style windowed self-attention [49]. Each token is built from the queried latent feature, absolute coordinate, relative offset, and target cell size, making the renderer coordinate-grounded and scale-aware. Windowed self-attention over these output tokens enables inter-pixel reasoning within and across local windows while keeping rendering cost scalable.

Coordinate Token Construction. Given an input latent $\boldsymbol{z} \in \mathbb{R}^{C\times H\times W}$ and a target coordinate $\boldsymbol{q}_{i,j}$, CQAR forms an output-coordinate token by

$$\boldsymbol{h}_{i,j} = \phi_{\mathrm{in}}\big(\boldsymbol{z}_{\mathrm{nn}}(\boldsymbol{q}_{i,j}), \Delta\boldsymbol{q}_{i,j}, \Delta\boldsymbol{c}_{i,j}, \boldsymbol{q}_{i,j}\big), \tag{8}$$

where $\boldsymbol{z}_{\mathrm{nn}}(\boldsymbol{q}_{i,j})$ is the nearest-neighbor latent feature, $\Delta\boldsymbol{q}_{i,j} = \boldsymbol{q}_{i,j} - \boldsymbol{q}^*_{i,j}$ is the offset from the nearest latent-grid center $\boldsymbol{q}^*_{i,j}$, $\Delta\boldsymbol{c}_{i,j}$ is the relative cell-size descriptor converted from the raw pixel extent, and $\boldsymbol{q}_{i,j} \in [-1, 1]^2$ is the absolute coordinate. The projected tokens are processed by windowed Transformer blocks and mapped to RGB values. Thus, each token attends to neighboring output-coordinate tokens, enabling inter-pixel reasoning within and across local windows that pointwise implicit decoders lack.

3.4 Training and Inference

RaPD is trained in two stages. Stage-1 (Fig. 3 (A)) learns a semantically guided NIF latent. Stage-2 (Fig. 3 (B)) freezes the NIF encoder and CQAR, then trains a text-conditioned DiT denoiser over normalized NIF latents with flow matching. During inference (Fig. 3 (C)), the denoiser generates one normalized NIF latent, and CQAR renders the latent on any target coordinate grid.

3.4.1 Training

Stage-1: Generation-Aware NIF Learning. Stage-1 trains the NIF encoder and CQAR to learn a generation-aware continuous latent while preserving coordinate-based renderability. Specifically, the encoder E_ϕ maps an input image $\boldsymbol{x}_{\text{in}}$ to a dense NIF latent $\boldsymbol{z} = E_\phi(\boldsymbol{x}_{\text{in}})$, and CQAR renders this latent over target coordinates. To prevent the latent from being reconstruction-oriented, we optimize it with the following loss:

$$\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{rec}} + w_{\text{adapt}}\mathcal{L}_{\text{distill}}, \quad w_{\text{adapt}} = w_{\text{base}} \cdot \text{clamp}\left(\frac{\|\nabla_{\phi_L}\mathcal{L}_{\text{rec}}\|}{\|\nabla_{\phi_L}\mathcal{L}_{\text{distill}}\| + \delta_1}, 0, 10^8\right), \quad (9)$$

where $\mathcal{L}_{\text{distill}}$ is the semantic distillation loss defined in Sec. 3.2, ϕ_L denotes the parameters of the encoder's last convolutional layer, $\delta_1 = 10^{-4}$, and $w_{\text{base}} = 0.1$. This adaptive weighting balances the reconstruction and semantic objectives, shaping $\boldsymbol{z}$ into a generation-aware NIF latent without sacrificing reconstruction fidelity.

We further adopt multi-resolution supervision to expose CQAR to varying output cell sizes. Given a training image, we sample $r \sim \mathcal{U}[1, 4]$ and extract a high-resolution crop of size $(\lfloor rH_0 \rceil, \lfloor rW_0 \rceil)$ as the target image $\boldsymbol{x}_{\text{tar}}$, where $(H_0, W_0) = (256, 256)$ is the fixed Stage-1 encoder input resolution, giving target resolutions up to 1024×1024. We downsample $\boldsymbol{x}_{\text{tar}}$ to obtain $\boldsymbol{x}_{\text{in}}$ and encode it into the NIF latent space. The queried coordinates and cell sizes are defined with respect to the full high-resolution crop, so that the same latent is trained to render RGB values under different output cell sizes using per-pixel reconstruction loss in Eq. (3). After Stage-1, per-channel statistics $\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z$ are computed over training-set latents for Stage-2 normalization.

Stage-2: Diffusion Training. Once finished with semantic distillation, Stage-2 trains a text-conditioned DiT denoiser D_θ over the learned NIF latent. For each image, we encode $\boldsymbol{x}_{\text{in}}$ into $\boldsymbol{z} = E_\phi(\boldsymbol{x}_{\text{in}})$ and normalize it as $\tilde{\boldsymbol{z}} = (\boldsymbol{z} - \boldsymbol{\mu}_z)/\boldsymbol{\sigma}_z$. Here the data endpoint $\boldsymbol{u}_1$ in Eq. (1) is instantiated as $\tilde{\boldsymbol{z}}$. We sample t following Eq. (7), draw $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and train the denoiser with

$$\boldsymbol{u}_t = (1-t)\boldsymbol{\epsilon} + t\tilde{\boldsymbol{z}}, \quad \boldsymbol{v}_\theta = \frac{D_\theta(\boldsymbol{u}_t, t, \boldsymbol{y}_{\text{cond}}) - \boldsymbol{u}_t}{1-t}, \quad \mathcal{L}_{\text{FM}} = \|\boldsymbol{v}_\theta - (\tilde{\boldsymbol{z}} - \boldsymbol{\epsilon})\|_2^2. \quad (10)$$

Following JiT [44], D_θ directly predicts the clean normalized latent; the denominator is clamped to at least 10^{-8} in implementation. The final Stage-2 loss is $\mathcal{L}_{\text{Stage2}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}$, where $\mathcal{L}_{\text{REPA}}$ aligns intermediate denoiser features with frozen DINOv2 features [79]; we use $\lambda_{\text{REPA}} = 0.5$.

3.4.2 Inference

At inference, we initialize Gaussian noise with the normalized NIF latent shape and integrate the learned velocity field with Euler ODE sampling [66], using classifier-free guidance [28]. This produces a generated normalized latent $\tilde{\boldsymbol{z}}_{\text{gen}}$, which is denormalized as $\hat{\boldsymbol{z}} = \tilde{\boldsymbol{z}}_{\text{gen}}\boldsymbol{\sigma}_z + \boldsymbol{\mu}_z$. For a target size (H', W'), we follow the LIIF coordinate convention [11]:

$$\boldsymbol{q}_{i,j} = \left(\frac{2i+1}{H'} - 1,\ \frac{2j+1}{W'} - 1\right), \quad \boldsymbol{c}_{i,j} = \left(\frac{2}{H'},\ \frac{2}{W'}\right), \quad (11)$$

where $\boldsymbol{c}_{i,j}$ is the raw pixel extent from which CQAR derives $\Delta\boldsymbol{c}_{i,j}$. Evaluating Eq. (4) with $\hat{\boldsymbol{z}}$ on this grid yields $\hat{\boldsymbol{x}}_{H' \times W'}$. Thus, only the query grid changes across resolutions; diffusion cost remains fixed, while rendering cost scales with queried tokens, as shown in Tab. 4.

4 Experiments

Dataset and Evaluation. Following PixelGen [53] and DeCo [52], we use BLIP3o [5] as the primary training corpus. RaPD is pretrained on approximately 36M images and fine-tuned on 60k curated high-quality images in 512×512 for improved visual fidelity and text-image alignment. We evaluate reconstruction with rFID [78] and text-to-image generation with GenEval [23] and DPG-Bench [32], following the PixelGen [53] protocol for fair comparison.



Figure 5: Qualitative 512^2 samples: RaPD vs. pixel-diffusion baselines [53, 52, 69].

Table 1: Text-to-image generation on GenEval [23] (512×512). ‡ indicates our re-runs of released models under the same seed and environment.

Method	#Params	GenEval ↑						
		Single Obj.	Two Obj.	Counting	Colors	Position	Color Attr.	Overall
▽ *Latent Diffusion Models*								
LDM	1.4B	0.92	0.29	0.23	0.70	0.02	0.05	0.37
SD3	8B	0.98	0.84	0.66	0.74	0.40	0.43	0.68
FLUX.1-dev	12B	0.99	0.81	0.79	0.74	0.20	0.47	0.67
SANA	1.6B	0.99	0.79	0.63	0.88	0.18	0.47	0.66
▽ *Pixel Diffusion Models*								
PixelFlow-XL/4	0.9B	–	–	–	–	–	–	0.60
PixNerd-XXL/16	1.2B	0.97	0.86	0.44	0.83	0.71	0.53	0.73
DeCo-XXL/16	1.1B	**1.00**	0.92	0.72	0.91	**0.80**	**0.79**	**0.86**
PixelGen-XXL/16	1.1B	0.99	0.88	0.59	0.90	0.70	0.70	0.79
PixNerd-XXL/16 (‡)	1.2B	0.98	0.82	0.43	0.81	0.66	0.55	0.71
DeCo-XXL/16 (‡)	1.1B	0.99	0.92	0.65	0.91	0.71	0.77	0.83
PixelGen-XXL/16 (‡)	1.1B	0.99	0.88	0.58	0.91	0.70	0.67	0.79
RaPD	1.1B	0.99	**0.93**	0.72	**0.93**	0.77	0.74	0.85
RaPD (256→512)	1.1B	0.99	**0.93**	**0.77**	0.92	0.79	0.78	**0.86**

Baselines. RaPD is compared with representative latent- and pixel-diffusion models. Latent-diffusion models include LDM [58], SD3 [19], FLUX.1-dev [41], PixArt-α [7], and SANA [75]. Pixel-diffusion models include PixelFlow [10], PixNerd [69], DeCo [52], and PixelGen [53].

Implementation details. In Stage 1, we train the NIF encoder and renderer for 10k steps on the 36M-image corpus. Semantic distillation uses the final-layer patch tokens from a frozen DINOv2 ViT-B/14 encoder as the teacher feature $\boldsymbol{t}$. In Stage 2, the diffusion model is pretrained for 260k steps and fine-tuned for 40k steps on $8\times$NVIDIA B200 GPUs, taking about 5 days in total. At inference, we use 25-step Euler ODE sampling with CFG scale 4.0. More details can be found in Appendix.

4.1 Main Results

4.1.1 Text-to-Image Generation Compared with Baseline Methods

Qualitative Comparison. Fig. 5 compares RaPD with recent pixel-diffusion baselines at 512×512. Across diverse prompts, RaPD produces more coherent global structure, cleaner local details, and fewer artifacts than competing methods. It better preserves object identity, reduces failures



Figure 6: Arbitrary-resolution generation. Left: GenEval [23] prompts rendered at multiple output scales. Right: DPG-Bench [32] examples showing strong long-form prompt fidelity.

Table 2: Text-to-image generation results on DPG-Bench [32] (512×512). ‡ denotes results obtained by running the released models with the same random seed and environment as RaPD.

Method	#Params	DPG-Bench ↑					
		Global	Entity	Attribute	Relation	Other	Average
▽ *Latent Diffusion Models*							
SD v2	0.9B	77.7	78.1	74.9	80.7	80.7	68.1
PixArt-α	0.6B	75.0	79.3	78.6	82.6	77.0	71.1
SANA	1.6B	90.3	91.2	89.0	88.9	92.0	85.5
▽ *Pixel Diffusion Models*							
PixelFlow-XL/4	0.9B	–	–	–	–	–	77.9
PixNerd-XXL/16	1.2B	80.5	87.9	87.2	**91.3**	72.8	80.9
DeCo-XXL/16	1.1B	–	–	–	–	–	81.4
PixNerd-XXL/16 (‡)	1.2B	87.2	87.1	88.9	87.1	**89.8**	81.5
PixelGen-XXL/16 (‡)	1.1B	81.4	88.8	86.7	88.9	88.9	80.4
DeCo-XXL/16 (‡)	1.1B	**88.6**	88.4	88.9	87.6	86.0	81.2
RaPD	1.1B	85.1	87.7	<u>88.9</u>	<u>89.6</u>	84.1	<u>81.6</u>
RaPD (256→512)	1.1B	81.3	**89.9**	**89.7**	88.9	81.2	**82.9**

such as broken contours, duplicated parts, and distorted fine structures, and more reliably satisfies compositional constraints involving multiple objects, attributes, and spatial relations.

Quantitative Comparison. Tab. 1 and Tab. 2 evaluate standard text-to-image quality at 512×512 on GenEval [23] and DPG-Bench [32]. RaPD (256→512) follows pixel diffusion methods by pretraining at 256^2, continued pretraining at 512^2, and final SFT for fair comparison. On GenEval, RaPD reaches an overall score of 0.86, outperforming PixelGen and the reproduced DeCo results. The improvement is most evident in compositional categories, including two-object, counting, and color, suggesting that the semantic-aligned NIF latent provides useful semantic structure for text-conditioned synthesis. On DPG-Bench, RaPD reaches an average score of 82.9, outperforming the reproduced pixel-diffusion baselines with strong attribute and relation scores.

4.1.2 Evaluation of Arbitrary-Resolution Generation.

Qualitative and Quantitative Results. Fig. 6 shows RaPD results at 1024^2, 2048^2, and 4096^2 on GenEval and DPG-Bench examples. The NIF latent is learned with multi-resolution supervision up to 1024^2, and RaPD supports rendering beyond this supervision range at 2048^2 and 4096^2. The high-resolution samples remain coherent and contain clear boundaries and rich local details. The DPG-Bench results further indicate strong fidelity to long and complex prompts at large output scales.

Table 3: Comparison with high-resolution generation baselines.

Method	#Params	1024^2		2048^2	
		GenEval ↑	DPG ↑	GenEval ↑	DPG ↑
SANA	1.6B	0.66	**84.8**	0.70	83.1
SANA-v1.5	1.6B	0.82	84.5	0.80	**83.8**
FLUX.1-dev	12B	0.68	83.1	0.66	82.8
RaPD (256→1024)	1.1B	**0.85**	80.3	**0.85**	80.1



Figure 7: Ablations of Semantic Distillation and CQAR.

Quantitative Comparison with High-Resolution Methods We compare RaPD ($256 \rightarrow 1024$), progressively trained to 1024^2, with high-resolution baselines at matched output sizes. It achieves the best GenEval score of 0.85 at both 1024^2 and the unseen 2048^2 resolution (Table 3). Its lower DPG scores indicate room for improvement in detailed prompt adherence.

Time and Computational Cost. Tab. 4 shows that RaPD scales much more favorably than fixed-grid pixel diffusion at high resolutions. RaPD keeps the expensive denoising stage nearly fixed, while PixelGen and DeCo grow quadratically with the output resolution, leading to a large increase in inference time. At 4096^2, RaPD requires only 2.13s, whereas PixelGen and DeCo both exceed 61s.

4.2 Model Analysis

Ablation on Semantic Distillation and REPA. Removing Semantic Distillation or replacing DINOv2 with CLIP weakens semantic structure (Fig. 7). Removing REPA causes a larger drop under the same training budget (Table 5), suggesting that direct guidance of the denoiser is particularly important for exploiting NIF latents. Stage-1 distillation improves the latent representation and provides additional gains even with REPA enabled.

Exploration of Latent-Density-Aware Timestep Shift. Fig. 8 studies the Latent-Density-Aware Timestep Shift in learning our dense NIF latent. Small shift values lead to weak object formation and poor semantic organization, showing that schedules suited to compressed VAE latents do not transfer well to our full-resolution NIF representation. Increasing the shift substantially improves

Table 4: Per-image inference time from $1\times$ to $8\times$ output scales on a B200 GPU. Values in parentheses indicate the increase over 512×512.

Method	512×512 ($1\times$)		1024×1024 ($2\times$)		2048×2048 ($4\times$)		4096×4096 ($8\times$)	
	#Tokens	Time (s)	#Tokens	Time (s)	#Tokens	Time (s)	#Tokens	Time (s)
PixelGen	1,024	0.19	4,096	0.60	16,384	4.72	65,536	61.46 (+61.27)
DeCo	1,024	0.21	4,096	0.67	16,384	5.03	65,536	62.72 (+62.51)
RaPD	1,024	0.23	1,024	0.34	1,024	0.68	1,024	2.13 (+1.9)

Table 5: Ablation on Semantic Distillation and REPA.

Setting	GenEval ↑	DPG ↑
w/o Stage-1 Semantic Distillation	0.83	78.9
w/o Stage-2 REPA	0.61	56.6
RaPD (Full model)	**0.85**	**81.6**

generation quality, and $s = 32$ provides a better trade-off between clear object layout and rich visual detail. When the shift becomes too large ($s = 48$), the generated images become overly simplified.

The Effect of CQAR. Fig. 2 compares CQAR with LIIF- and CLIF-style renderers on image reconstruction. CQAR produces sharper reconstructions with clearer boundaries and better recovery of fine structures. Following DC-AE 1.5 [9], Fig. 9 visualizes the learned NIF features and shows that CQAR preserves clearer spatial details in the feature maps. Fig. 7 further compares the generation results. RaPD achieves the lowest rFID and the best visual quality, indicating that CQAR effectively alleviates the reconstruction–generation gap of continuous NIF representations.

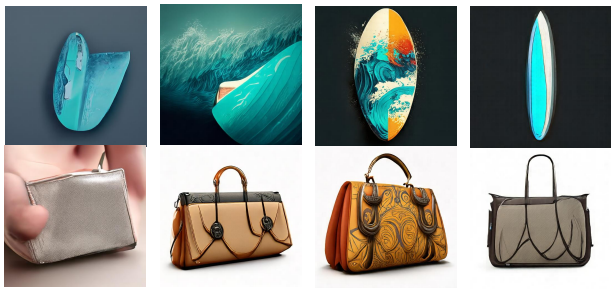


Figure 8: Generation under different timestep shift (prompt: "*a photo of a surfboard / handbag* ").

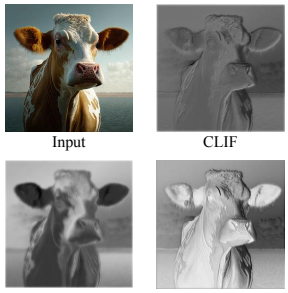


Figure 9: Latent feature visualization.

5 Conclusion and Limitations

In this paper, we presented **RaPD**, which performs diffusion in a continuous Neural Image Field latent and decodes it through a coordinate-queried windowed-attention renderer, achieving resolution-agnostic image generation while keeping the diffusion cost fixed regardless of the output size. Three directions naturally extend this work: *(i)* scaling to larger datasets and models; *(ii)* extending the continuous NIF latent to controllable generation and editing; and *(iii)* hardware-aware inference acceleration to fully realize RaPD's efficiency potential at high resolutions.

Limitations. RaPD still has several limitations. *(1)* It is trained on approximately 36 million BLIP3o images, substantially fewer than the data used by commercial-scale models such as SD3 and FLUX, which limits its absolute generation quality. *(2)* The frozen Qwen3-1.7B text encoder is relatively compact and may constrain complex prompt understanding compared with systems paired with larger language models. *(3)* The current inference stack does not yet incorporate FlashAttention [16], fused window-attention kernels for CQAR, FP8 quantization, or compiler-level fusion; the reported measurements reflect the current implementation, with further efficiency improvements left to future work.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62473290, the National Key R&D Program of China under Grant No. 2024YFB3311801, the Shanghai Municipal Science and Technology Commission under Grant No. 25511102800, the Shanghai Municipal Commission of Economy and Information Technology under Grant No. 2024-GZL-RGZN-01008.

Competing Interests

The authors declare no competing interests.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *CVPR*, pages 14278–14287, 2021.

[2] 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 (2023). *URL https://cdn. openai. com/papers/dall-e-3. pdf*, 6, 2023.

[3] Jiazi Bu, Pengyang Ling, Yujie Zhou, Pan Zhang, Tong Wu, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Hiflow: Training-free high-resolution image generation with flow-aligned guidance. *arXiv preprint arXiv:2504.06232*, 2025.

[4] Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. In *ECCV*, pages 170–188. Springer, 2022.

[5] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. 2023.

[7] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-$alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[8] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. 2025.

[9] Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19628–19637, 2025.

[10] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow, 2025.

[11] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, pages 8628–8638, 2021.

[12] Yinbo Chen, Oliver Wang, Richard Zhang, Eli Shechtman, Xiaolong Wang, and Michael Gharbi. Image neural field diffusion models. In *CVPR*, pages 8007–8017, 2024.

[13] Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. *arXiv preprint arXiv:2511.18822*, 2025.

[14] Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Jiawei Chen, Zhuoqi Zeng, Wei Zhang, Chengjie Wang, Jian Yang, and Ying Tai. L2p: Unlocking latent potential for pixel generation. *arXiv preprint arXiv:2605.12013*, 2026.

[15] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *ICML*, 2024.

[16] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.

[17] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021.

[18] Ruoyi Du, Dongliang Chang, Kaiyue Pang, Yi-Zhe Song, and Zhanyu Ma. DemoFusion: Democratising high-resolution image generation with no $$$. In *CVPR*, pages 6814–6824, 2024.

[19] 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 *ICML*, 2024.

[20] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. 2024.

[21] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. 2024.

[22] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *CVPR*, pages 10021–10030, 2023.

[23] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, volume 36, pages 52132–52152, 2023.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[25] Tao Han, Wanghan Xu, Junchao Gong, Xiaoyu Yue, Song Guo, Luping Zhou, and Lei Bai. Infgen: A resolution-agnostic paradigm for scalable image synthesis. In *ICCV*, pages 17941–17950, 2025.

[26] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *ICLR*, 2023.

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[28] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. 2022.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[30] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, pages 13213–13232. PMLR, 2023.

[31] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18062–18071, 2025.

[32] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024.

[33] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *CVPR*, pages 1575–1584, 2019.

[34] David H. Hubel and Torsten N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, pages 106–154, 1962.

[35] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[37] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020.

[38] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, pages 24174–24184, 2024.

[39] Jinseok Kim and Tae-Kyun Kim. Arbitrary-scale image generation and upsampling using latent diffusion model and implicit neural decoder. In *CVPR*, pages 9202–9211, 2024.

[40] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[41] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[42] Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no vae: End-to-end pixel-space generative modeling via self-supervised pre-training. 2026.

[43] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *ICCV*, pages 18262–18272, 2025.

[44] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2026.

[45] Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. 2025.

[46] Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. 2025.

[47] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR-W*, pages 136–144, 2017.

[48] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2023. doi: 10.48550/arXiv.2210.02747.

[49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[50] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024.

[51] 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 *ECCV*, pages 23–40. Springer, 2024.

[52] Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. November 2025.

[53] Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. 2026.

[54] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[55] 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. February 2024.

[56] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4172–4182. IEEE, 2023.

[57] 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, 2023.

[58] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[60] Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. December 2025.

[61] Claude E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1): 10–21, 1949. doi: 10.1109/JRPROC.1949.232969.

[62] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. 2025.

[63] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 33:7462–7473, 2020.

[64] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *CVPR*, pages 10753–10764, 2021.

[65] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[66] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2020.

[67] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[68] Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Flowdcn: Exploring dcn-like architectures for fast image generation with arbitrary resolution. In *NeurIPS*, pages 87959–87977, 2024.

[69] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. August 2025.

[70] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. April 2025.

[71] ZiDong Wang, Zeyu Lu, Di Huang, Cai Zhou, Wanli Ouyang, and Lei Bai. Fitv2: Scalable and improved flexible vision transformer for diffusion model. *arXiv preprint arXiv:2410.13925*, 2024.

[72] Zidong Wang, Lei Bai, Xiangyu Yue, Wanli Ouyang, and Yiyuan Zhang. Native-resolution image synthesis. June 2025.

[73] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. June 2025.

[74] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. July 2025.

[75] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

[76] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. May 2025.

[77] Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *NeurIPS*, 37:56166–56189, 2024.

[78] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, pages 15703–15712, 2025.

[79] 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, 2025.

[80] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. November 2025.

[81] Xiaoyu Yue, Zidong Wang, Zeyu Lu, Shuyang Sun, Meng Wei, Wanli Ouyang, Lei Bai, and Luping Zhou. Diffusion models need visual priors for image generation. 2024.

[82] Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. June 2025.

[83] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[84] Min Zhao, Bokai Yan, Xue Yang, Hongzhou Zhu, Jintao Zhang, Shilong Liu, Chongxuan Li, and Jun Zhu. UltraImage: Rethinking resolution extrapolation in image diffusion transformers, 2025. arXiv:2512.04504.

[85] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. October 2025.

[86] Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. Farmer: Flow autoregressive transformer over pixels. October 2025.

[87] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *ICML*, 2024.

Appendix

This supplementary material includes the following:

A. Related Work

B. More Implementation Details

C. Additional Experimental Results

A Related Work

A.1 Image Generation and Resolution Flexibility

Image generative models have evolved from GANs [24, 35–37] through denoising diffusion probabilistic models [65, 29, 66, 67] to the Diffusion Transformer (DiT) paradigm [56, 51, 21], which has become the dominant backbone for high-quality image synthesis. Latent Diffusion Models (LDMs) [58, 57, 19, 41] dramatically reduce computational cost by operating in a compressed VAE latent space, but this compression introduces an information bottleneck that degrades fine-grained spatial details and fixes the output resolution [40, 8, 78, 43]. Motivated by this, pixel diffusion methods [80, 13, 44, 52, 10, 53, 15, 42, 30] bypass the VAE entirely to preserve complete spatial information, with recent work demonstrating competitive quality at resolutions up to 1024×1024 in pixel space. Tab. 6 summarizes the key differences among these method categories.

Achieving resolution flexibility within these frameworks remains challenging. Within the LDM paradigm, several methods pair the diffusion process with flexible decoders to support variable output resolutions [50, 71, 39, 12, 25, 72, 68]; however, the diffusion process itself still operates on a fixed-resolution compressed latent, confining the model's semantic planning to a single scale. Early GAN-based approaches replace spatial convolutions with coordinate-queried MLPs for multi-scale synthesis [1, 64, 4], but suffer from training instability and degraded quality at out-of-distribution resolutions [27]. Among pixel-space methods, PixNerd [69] supports rendering beyond the training grid through a patch-wise neural field decoder. RaPD likewise retains a discrete latent feature grid, but focuses on jointly designing semantically guided NIF representation learning and coordinate-conditioned attention rendering. Relative to fixed-grid pixel diffusion, it decouples diffusion cost from output resolution: a single generated latent can be rendered on different output grids.

A.2 Implicit Neural Representations for Generation

Implicit neural representations (INRs) [63, 54] model signals as continuous functions mapping coordinates to values. In the 2D image domain, LIIF [11] and MetaSR [33] established the paradigm of decoding a shared feature map at arbitrary resolutions via coordinate-conditioned MLPs. Subsequent work applied this paradigm to generation: IDM [22] performs INR decoding at every denoising step but at prohibitive cost; INFD [12] applies diffusion to INR-based latents using a convolutional

Table 6: Comparison of representative image generation method categories. **Pixel-Space**: the forward/reverse diffusion process operates in pixel space without lossy VAE compression. **Arb. Resolution Output**: supports synthesizing images at arbitrary target resolutions at inference time. **Continuous Semantic Prior**: the generative prior models spatial continuity with semantically-aware features rather than treating pixels as discrete grid elements. **Decoder Type**: the decoding mechanism used to produce pixel values. Per-category methods: Latent Diffusion Models [58, 19, 41]; Latent Diffusion w/ Flexible Resolution [50, 71, 39, 12, 25, 72]; Pixel Diffusion w/ Fixed Resolution [80, 13, 44, 53, 52, 15, 42]; Pixel Diffusion w/ Partial Continuity [69]. ✓ = fully supported; ○ = partially supported; × = not supported.

Method Category	Pixel-Space	Arb. Resolution Output	Continuous Semantic Prior	Decoder Type
Latent Diffusion Models	×	×	×	Fixed Conv
Latent Diffusion w/ Flexible Resolution	×	○	×	INR / Conv
Pixel Diffusion w/ Fixed Resolution	✓	×	×	None
Pixel Diffusion w/ Partial Continuity	✓	○	○	Partial INR
Ours (RaPD)	✓	✓	✓	CQAR

rendering module (CLIF); and InfGen [25] replaces the VAE decoder with a transformer-based generator for single-step extrapolation to 4K.

However, all existing INR-based representations are still optimized for reconstruction, not generation, suffering from two intertwined limitations. First, their latent features, trained with pixel-level losses as listed in Eq. 3, encode low-level texture statistics rather than the semantic structure required by a generative model. Second, their decoders (pointwise MLPs [11] or shallow CNNs [12]) lack the receptive field to exploit semantic features for coherent synthesis. Although recent works on foundation-model alignment, such as REPA [79, 74, 43] and VA-VAE [78] leverage this principle to accelerate diffusion training and balance VAE capacity respectively, our reconstruction–generation gap is fundamentally different: it concerns the semantic content of the NIF latent itself, and arises even when latent capacity is unbounded. Whereas REPA aligns denoiser features and VA-VAE shapes a separate VAE post-hoc, our *Semantic Distillation* directly shapes the NIF latent itself during its very first training stage, making generation-aware structure available to the diffusion process from the start.

RaPD addresses both limitations jointly: Semantic Representation Guidance (SRG) enriches the latent with semantic structure through a foundation-model distillation procedure, and a Coordinate-Queried Attention Renderer (CQAR) replaces local decoders with coordinate-conditioned, scale-aware Transformer blocks [49] for inter-pixel semantic reasoning at arbitrary resolutions.

B More Implementation Details

B.1 Pseudocode of RaPD

We summarize the training pipeline (Stage-1 + Stage-2) of RaPD in Alg. 1 and the arbitrary-resolution inference pipeline in Alg. 2.

Algorithm 1 RaPD: training pipeline (Stage-1 + Stage-2).

Stage-1: NIF Semantic Distillation

Sample low-/high-resolution image pair $\boldsymbol{x}_{\mathrm{lr}}, \boldsymbol{x}_{\mathrm{hr}}$ following Sec. 3.4.1
$\boldsymbol{z} \leftarrow E(\boldsymbol{x}_{\mathrm{lr}})$ ▷ encode to NIF latent at full spatial resolution
$\hat{\boldsymbol{x}}_{\mathrm{hr}} \leftarrow G(\boldsymbol{z}, \boldsymbol{q}_{\mathrm{hr}}, \Delta \boldsymbol{c}_{\mathrm{hr}})$ ▷ render the HR patch via CQAR
$\boldsymbol{z}_{\mathrm{proj}} \leftarrow \mathrm{Proj}(\mathrm{Pool}(\boldsymbol{z}))$ ▷ project $\boldsymbol{z}$ to the DINOv2 patch space
$\mathcal{L}_{\mathrm{Stage1}} \leftarrow \mathcal{L}_{\mathrm{rec}} + w_{\mathrm{adapt}} \cdot \mathcal{L}_{\mathrm{distill}}$ ▷ reconstruction + semantic distillation
Update E, G by $\nabla \mathcal{L}_{\mathrm{Stage1}}$; after training, compute per-channel $\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z$ on the training set

Stage-2: Flow Matching with Generation-aware NIF

$\tilde{\boldsymbol{z}} \leftarrow (E(\boldsymbol{x}) - \boldsymbol{\mu}_z)/\boldsymbol{\sigma}_z$ ▷ normalized NIF latent serves as the data endpoint $\boldsymbol{x}_1$
$\xi \sim \mathcal{N}(0,1)$, $t \leftarrow \mathrm{shift}_s(\mathrm{sigmoid}(\xi))$ ▷ logit-normal t with the SRG-prescribed shift s
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$; $\boldsymbol{x}_t \leftarrow (1-t)\,\boldsymbol{\epsilon} + t\,\tilde{\boldsymbol{z}}$ ▷ noised latent at timestep t
$\tilde{\boldsymbol{z}}_\theta \leftarrow D_\theta(\boldsymbol{x}_t, t, \boldsymbol{y}_{\mathrm{cond}})$ ▷ denoiser predicts the clean normalized latent
$\mathcal{L}_{\mathrm{Stage2}} \leftarrow \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{REPA}} \cdot \mathcal{L}_{\mathrm{REPA}}$ ▷ flow matching + REPA feature alignment
Update θ by $\nabla \mathcal{L}_{\mathrm{Stage2}}$

Algorithm 2 RaPD: arbitrary-resolution inference.

Input: text prompt $\boldsymbol{y}$ and target output size (H', W')
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with the shape of $\tilde{\boldsymbol{z}}$ ▷ initialize Gaussian noise at the NIF latent shape
$\tilde{\boldsymbol{z}}_\theta \leftarrow \mathrm{EulerODE}(D_\theta, \boldsymbol{\epsilon}, \boldsymbol{y}_{\mathrm{cond}})$ ▷ denoise once with CFG; cost independent of (H', W')
$\hat{\boldsymbol{z}} \leftarrow \tilde{\boldsymbol{z}}_\theta \cdot \boldsymbol{\sigma}_z + \boldsymbol{\mu}_z$ ▷ denormalize back to the encoder latent scale
$(\boldsymbol{q}, \boldsymbol{c}) \leftarrow \mathrm{CoordGrid}(H', W')$ ▷ construct query coordinates and cell sizes for the target
$\hat{\boldsymbol{x}}_{H' \times W'} \leftarrow G(\hat{\boldsymbol{z}}, \boldsymbol{q}, \boldsymbol{c})$ ▷ a single CQAR pass renders the final image
return $\hat{\boldsymbol{x}}_{H' \times W'}$

B.2 Continuous Coordinate System

Following LIIF [11], the spatial domain is normalized to $[-1, 1]^2$. For a target output resolution (H', W'), we generate:

- A coordinate grid $\mathbf{q} \in \mathbb{R}^{H' \times W' \times 2}$, where $\mathbf{q}_{i,j}$ gives the center of output pixel (i, j) in normalized coordinates.
- A cell-size grid $\mathbf{c} \in \mathbb{R}^{H' \times W' \times 2}$ with $\mathbf{c}_{i,j} = (2/H', 2/W')$, encoding the spatial extent of each output pixel as an explicit scale indicator. Smaller cells correspond to higher output resolutions, providing the renderer with resolution awareness.

B.3 Latent Normalization

After Stage 1 training, per-channel mean $\boldsymbol{\mu}_z \in \mathbb{R}^C$ and standard deviation $\boldsymbol{\sigma}_z \in \mathbb{R}^C$ are computed over the training dataset. The normalized latent is:

$$\tilde{\mathbf{z}} = \frac{\mathbf{z} - \boldsymbol{\mu}_z}{\max(\boldsymbol{\sigma}_z, 10^{-6})}, \tag{12}$$

which approximately follows $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, ensuring well-conditioned inputs for the diffusion model. The inverse transformation $\mathbf{z} = \tilde{\mathbf{z}} \cdot \boldsymbol{\sigma}_z + \boldsymbol{\mu}_z$ is applied before CQAR rendering.

B.4 CQAR Architecture Details

Window Transformer Blocks. Each of the 4 blocks applies:

$$\mathbf{x}' = \mathbf{x} + \mathrm{W-MSA}\big(\mathrm{LN}(\mathbf{x})\big), \tag{13}$$

$$\mathbf{x}'' = \mathbf{x}' + \mathrm{GatedFFN}\big(\mathrm{LN}(\mathbf{x}')\big), \tag{14}$$

where $\mathrm{W-MSA}$ is windowed multi-head self-attention with window size $W = 8$ and $N_h = 4$ heads. Odd-indexed blocks use shifted windows (shift $W/2$) with an attention mask for cross-window information exchange.

Windowed Self-Attention with Relative Position Bias. Within each $W \times W$ window:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} + \mathbf{B}\right)\mathbf{V}, \tag{15}$$

where $d_h = D/N_h$ is the head dimension and $\mathbf{B} \in \mathbb{R}^{W^2 \times W^2}$ is a learned relative position bias parameterized by a table of size $(2W-1)^2 \times N_h$.

Shifted-Window Self-Attention (SW-MSA). To enable cross-window information exchange without breaking the linear cost of windowed attention, every odd-indexed block uses Swin-style shifted-window attention [49]. Concretely, the token grid is first cyclically shifted by $(W/2, W/2)$, partitioned into the same $W \times W$ windows, and processed by the windowed attention in the equation above. Tokens that originate from different real windows but are merged into the same shifted window are blocked from attending to each other through an additive attention mask:

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{if tokens } i, j \text{ belong to the same real window,} \\ -\infty, & \text{otherwise,} \end{cases} \tag{16}$$

which is added to the pre-softmax logits. After attention, the cyclic shift is reversed so that the spatial layout is preserved. This shifted–regular alternation propagates information across window boundaries across consecutive blocks while keeping the per-block attention cost at $O(W^2)$ per token.

Gated Feed-Forward Network.

$$\mathrm{GatedFFN}(\mathbf{x}) = \mathrm{Linear}\big(\mathrm{GELU}(\mathbf{x}_1) \odot \mathbf{x}_2\big), \quad [\mathbf{x}_1, \mathbf{x}_2] = \mathrm{Linear}(\mathbf{x}), \tag{17}$$

with expansion factor $2\times$. The entire renderer is convolution-free, consisting exclusively of linear layers, layer normalization, and attention. A final LayerNorm and linear projection ($D \to 3$) produce the RGB output $\hat{\mathbf{x}} \in \mathbb{R}^{3 \times H' \times W'}$.

B.5 PixelGen-style DiT Denoiser Architecture

We adopt a PixelGen-style DiT denoiser [56, 53]:

1. **Patch embedding**: unfolds $\tilde{\mathbf{z}} \in \mathbb{R}^{16\times512\times512}$ into 16×16 patches, yielding 1024 image tokens compressed through a bottleneck ($4096 \to 1024 \to 1536$).
2. **Text encoder**: frozen Qwen3-1.7B [76] produces 128 tokens at dimension 2048, projected to hidden dimension 1536.
3. **Text-refinement blocks**: 4 blocks of self-attention on text tokens with AdaLN modulation from the timestep embedding.
4. **Image denoising blocks**: 16 DiT blocks [56] with cross-attention between image queries and text keys/values, using RoPE on image tokens, RMSNorm on Q/K, SwiGLU FFN, and gated attention. Both attention and FFN are modulated by AdaLN from the timestep condition.
5. **Final layer**: LayerNorm followed by linear projection, folding output back to $\tilde{\mathbf{z}}_\theta \in \mathbb{R}^{16\times512\times512}$.

The denoiser uses hidden dimension 1536 with 24 attention heads. A REPA loss [79] (weight 0.5) aligns the denoiser image-token features at transformer block index 8 with the final-layer patch-token features of a frozen DINOv2 ViT-B/14 encoder.

Timesteps follow the logit-normal distribution $t = \mathrm{sigmoid}(\xi),\ \xi \sim \mathcal{N}(0,\,1)$, shifted via $t' = t/(t + (1-t)\cdot s)$ with $s=32$ and $\delta_2 = 10^{-8}$.

B.6 Other Details

Table 7 summarizes all hyperparameters.

Table 7: RaPD model specifications and training hyperparameters.

Component	Parameter	Value
Stage 1: NIF AutoEncoder		
Encoder (EDSR)	Channels / ResBlocks	16 / 8
CQAR Renderer	Hidden / Blocks / Heads / Window	256 / 4 / 4 / 8
Distill Teacher	Model	DINOv2 ViT-B/14 (frozen)
Distill Margins	$m_{\cos}$ / m_{dist} / w_{base}	0.5 / 0.25 / 0.1
Training	Steps / LR / Batch	10K / 2×10^{-4} / 8
Stage 2: Flow Matching Denoiser		
DiT Denoiser	Hidden / Blocks / Heads	1536 / 16 / 24
	Patch size / Bottleneck	16 / 1024
Text Encoder	Qwen3-1.7B (frozen)	128 tokens, dim 2048
Flow Matching	Timestep shift s	**32.0**
	t sampling	LogitNormal ($\mu=0.0$, $\sigma=1.0$)
	δ_2	10^{-8}
	REPA weight	0.5
Training	Steps / LR / EMA decay	300K / 2×10^{-4} / 0.9999
	Precision / Grad clip / Batch	bfloat16 / 1.0 / 256
Inference		
Sampler	Euler ODE / 25 steps	CFG $w=4.0$
Resolution	Arbitrary	Tested 512–4096 (1×–8×)

Datasets. Stage 1 uses high-resolution images from BLIP3o for autoencoder training with multi-resolution supervision. Stage 2 uses BLIP3o image–text pairs (long and short caption variants) streamed via WebDataset at 512×512 resolution. We evaluate on GenEval [23] and DPG-Bench [32].

C Additional Experimental Results

C.1 Additional Qualitative Comparisons

To complement the qualitative results presented in the main paper, we provide additional side-by-side comparisons between RaPD and representative baselines on a broader set of prompts and target resolutions in Fig. 10. Across diverse content categories (objects, scenes, portraits, and text-rich compositions), RaPD consistently produces sharper textures, more coherent global structure, and stronger prompt alignment than competing methods. The advantage is particularly evident at higher

target resolutions, where baseline approaches either degrade in fine-grained detail or fail to scale beyond their training grid, whereas RaPD maintains visual fidelity by simply varying the CQAR query coordinate grid over the same denoised latent.

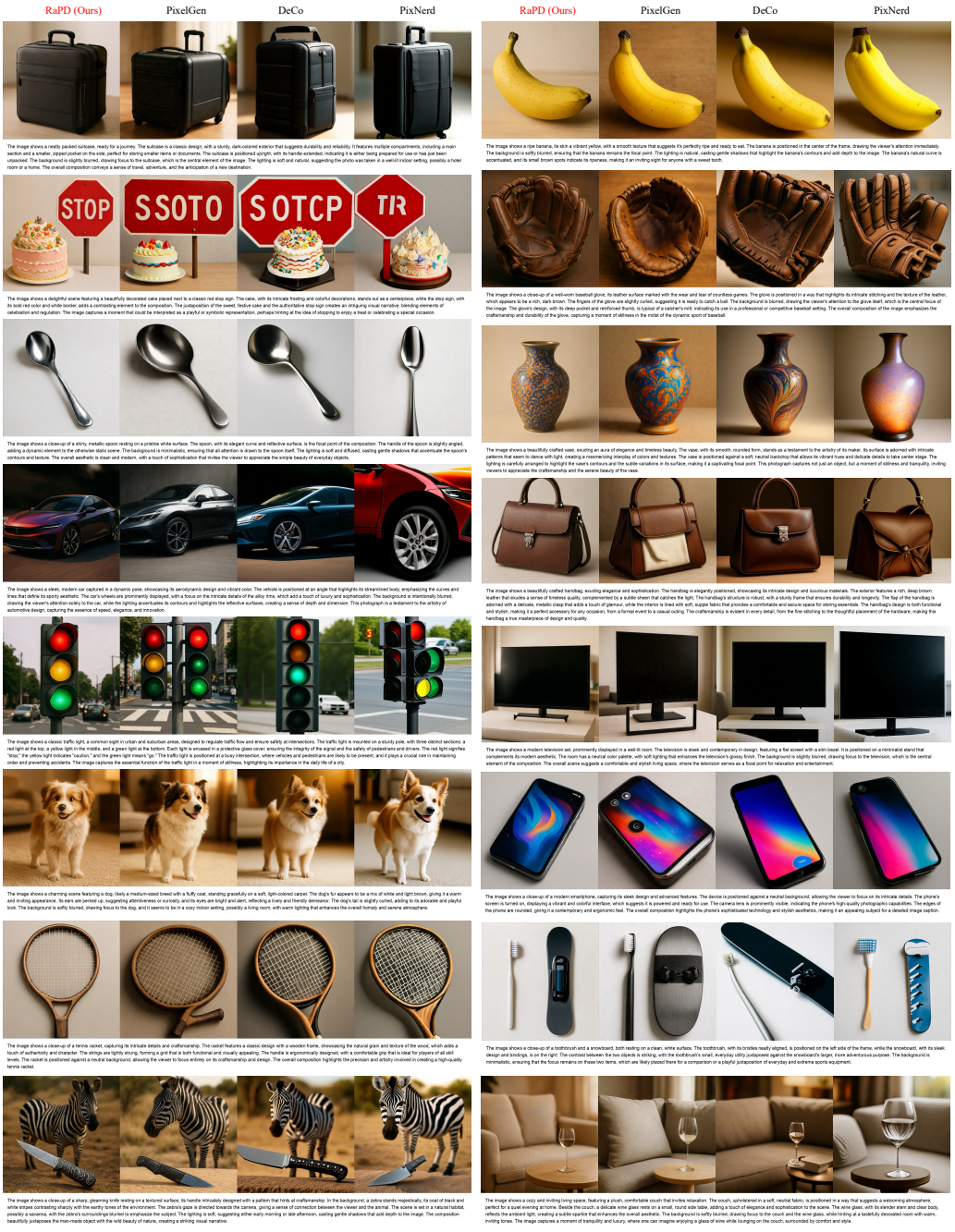


Figure 10: Additional qualitative text-to-image comparisons between RaPD and baselines. RaPD better preserves fine-grained detail and prompt fidelity, especially at higher resolutions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly summarize the main contributions, they are consistent with the methods and results represented in the paper.

   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 Sec. 5.

   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: The paper does not include any theoretical results, assumptions, or proofs.

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: The full method (encoder, CQAR, SRG, JiT denoiser) is described in Sec. 3; complete architecture and hyperparameter specifications are provided in Appendix Sec. B.4, B.5, and B.6, and training/data protocols are listed in Appendix Sec. B.6.

   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: The training data (BLIP3o) is publicly available, and all evaluation benchmarks (GenEval, DPG-Bench) are public. The code and pretrained checkpoints will be released upon acceptance to facilitate reproducibility; we do not release them at submission time to preserve anonymity.

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: The experimental protocol is described in Sec. 4; full hyperparameters (learning rate, batch size, optimizer, EMA decay, training steps, precision, gradient clipping, sampling settings, CFG scale, and timestep shift) are summarized in Tab. 7 of Appendix Sec. B.6.

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: [Yes]

Justification: Following standard practice in large-scale text-to-image generation, our quantitative results (GenEval, DPG-Bench, rFID) are reported on the full evaluation sets shown in Tab. 1 and Tab. 2

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: Training/inference compute (training steps, batch size, precision, sampler steps) and resolution-vs-cost scaling are reported in Tab. 7 of Appendix Sec. B.6 and Tab. 4 of Sec. 4. The diffusion cost is fixed regardless of the output size, and only the CQAR decoding cost scales with the output token count.

   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: This work is foundational research on resolution-agnostic image generation; it does not involve human subjects, sensitive personal data, or deployments that violate the NeurIPS Code of Ethics. All third-party assets are used under their respective licenses.

   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: Positive impacts include broader access to high-quality, resolution-flexible image synthesis with reduced compute, lowering the barrier for creative and educational applications. Potential negative impacts such as misuse for disinformation or non-consensual imagery are inherent to text-to-image generation in general; we discuss these in Sec. 5.

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: [Yes]

Justification: Upon release, we plan to distribute pretrained model weights with a usage license that prohibits malicious uses.

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: [Yes]

Justification: All third-party assets used in this work are properly cited. We use each asset within its respective license.

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: [Yes]

    Justification: The newly introduced asset is the RaPD model. Its full architecture and training procedure are documented in Sec. 3 and Appendix Sec. B.4, B.5, and B.6; on release, the model card and documentation will accompany the released checkpoints.

    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: This work does not involve crowdsourcing or research with human subject.

    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: This work does not involve research with human subjects, so IRB approval is not applicable.

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: [N/A]

Justification: We use Qwen3-1.7B as a frozen pre-trained text encoder, which is a standard component for text-conditioned image generation and not a novel methodological contribution. No LLM is used as a non-standard or original component of the proposed method.

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.