DiP: Taming Diffusion Models in Pixel Space

Zhennan Chen[1,2*] Junwei Zhu[2†] Xu Chen[2] Jiangning Zhang[2] Xiaobin Hu[3]
Hanzhen Zhao[3] Chengjie Wang[2] Jian Yang[1] Ying Tai[1‡]
[1]Nanjing University [2]Tencent Youtu Lab [3]National University of Singapore
https://github.com/NJU-PCALab/DiP

Abstract

Diffusion models face a fundamental trade-off between generation quality and computational efficiency. Latent Diffusion Models (LDMs) offer an efficient solution but suffer from potential information loss and non-end-to-end training. In contrast, existing pixel space models bypass VAEs but are computationally prohibitive for high-resolution synthesis. To resolve this dilemma, we propose DiP, an efficient pixel space diffusion framework. DiP decouples generation into a global and a local stage: a Diffusion Transformer (DiT) backbone operates on large patches for efficient global structure construction, while a co-trained lightweight Patch Detailer Head leverages contextual features to restore fine-grained local details. This synergistic design achieves computational efficiency comparable to LDMs without relying on a VAE. DiP is accomplished with up to 10× faster inference speeds than previous method while increasing the total number of parameters by only 0.3%, and achieves an 1.79 FID score on ImageNet 256×256.

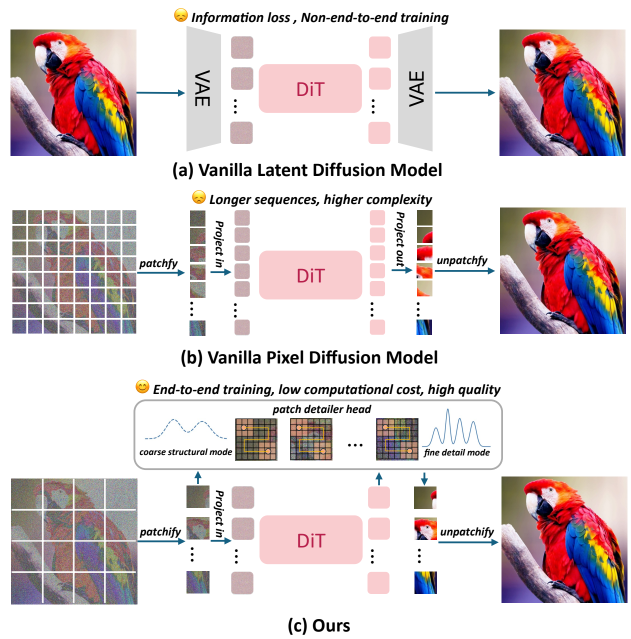

Figure 1. Comparison of vanilla latent diffusion model, vanilla pixel diffusion model and our method. Vanilla LDMs utilize VAEs to balance computational efficiency and generation quality. Vanilla pixel diffusion models use small patches to pursue detailed generation quality. Our method achieves high-quality generation while maintaining efficient end-to-end training in pixel space.

1. Introduction

Diffusion models [18, 19, 26, 37, 40, 43, 46–48, 56, 60] have reshaped the landscape of generative visual content. With its outstanding generative capabilities of fidelity and diversity, they have established new state-of-the-art benchmarks across a multitude of tasks, including image synthesis [3, 6–8, 11, 52, 59, 66, 68, 72, 73], video generation [13, 35, 62, 69], and 3D object creation [63–65], decisively surpassing prior paradigms like Generative Adversarial Networks (GANs) [14, 25, 34, 39, 67, 74]. However, this generative prowess is underpinned by immense computational demands. Consequently, the inherent *trade-off between generation quality and computational efficiency* thus stands as one of the most critical challenges in the field of diffusion models today.

To mitigate this challenge, Latent Diffusion Models (LDMs) [41] have emerged as the de facto standard. By employing a pre-trained autoencoder (VAE) [29] to compress high-resolution images into a compact latent space, LDMs significantly reduce the computational complexity of the iterative denoising process, as shown in Figure 1(a). Nevertheless, this approach is not without its limitations, including *potential information loss* [4, 16, 27, 58] *during VAE compression and a non-end-to-end training pipeline.*

The most direct solution to eliminate the shortcomings of LDMs is to train a diffusion model in pixel space. However, existing pixel space diffusion models [5, 9, 21, 22, 28, 49], particularly those based on the powerful Transformer architecture [51], face a severe scalability issue. As shown in Figure 1(b), to capture fine-grained details, they typically rely on small input patches (*e.g.* 2×2 or 4×4), causing the input sequence length to grow quadratically with image resolution. This quadratic scaling renders *high-resolution*

*Work done during the internship at Tencent. †Project Leader. ‡Corresponding Author.

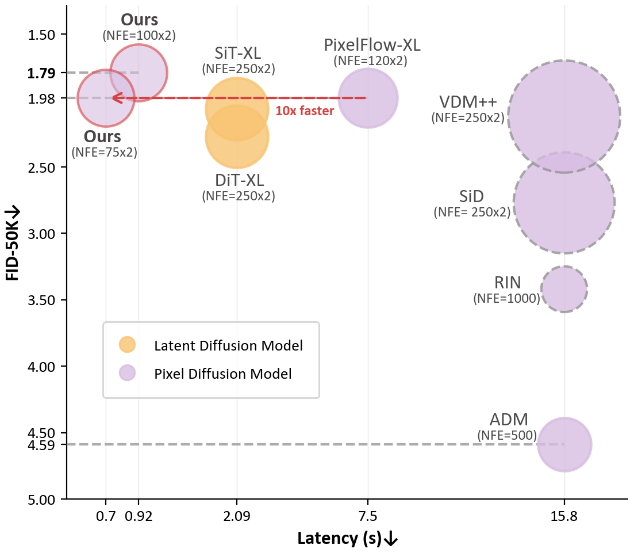

Figure 2. Our method achieves the best FID score with minimal computational cost. (Note: LDM latency includes VAE. The methods marked with dashed lines ( ) are our estimated latency based on the sampling method in the corresponding paper, and should actually be greater than the marked values. The rest methods are the actual test results in the same hardware environment.)

training and inference computationally intractable, creating a formidable barrier to their practical application.

In this paper, we aim to resolve this critical trade-off in pixel diffusion models. We propose an efficient pixel space diffusion framework called DiP. As shown in Figure 1(c), for efficient global structure construction, we employ a DiT [37] backbone. Critically, we configure it to operate on large image patches (*e.g.*, 16×16). This setting choice drastically reduces the input sequence length, aligning it with that of mainstream LDMs operating in latent space. Consequently, our model achieves computational efficiency comparable to LDMs while remaining entirely VAE-free, enabling it to effectively capture the global layout and semantic content of the image. However, operating on large patches alone inevitably leads to blurry outputs lacking high-frequency details. To address this, we introduce a lightweight Patch Detailer Head (only 0.3% increase in total parameters), which is not a post-processing module but an integral component co-trained with the DiT backbone. For each large patch, it receives contextual features from the DiT and leverages its strong local receptive fields to synthesize the missing high-frequency information. This synergistic design allows the DiT backbone to focus on the computationally demanding task of global consistency, while the efficient Patch Detailer Head specializes in local texture and detail restoration. As demonstrated in Figure 2, our approach sets a new state-of-the-art on the efficiency-quality frontier, achieving superior FID scores at significantly lower latency compared to existing methods. Our main contributions are summarized as follows:

- We propose DiP, a new end-to-end pixel diffusion model framework that effectively alleviates the trade-off between generation quality and computational efficiency through synergistic global-local modeling.
- We systematically validate the impact of different architectural designs of our framework, hoping to provide the community with a unified, principled framework.
- On ImageNet generation benchmarks, our framework achieves state-of-the-art performance and lowest inference latency with low training costs.

2. Related Work

Latent Diffusion Models. Latent Diffusion Models (LDMs) [2, 3, 12, 38, 41, 56, 61] have become the de-facto paradigm for large-scale generative modeling due to their computational efficiency and scalability. By performing the diffusion process in a compressed latent space learned by a VAE, LDMs drastically reduce memory and computational costs. Architectural advancements within this paradigm, such as replacing the U-Net [42] backbone with a more scalable Transformer (DiT) [37], have further pushed the boundaries of generation quality. Despite their success, this efficiency comes at a cost: the VAE acts as an information bottleneck, imposing a hard ceiling on the final image fidelity and often introducing subtle reconstruction artifacts [45, 58]. Our work circumvents these limitations by proposing an equally efficient architecture that operates directly in pixel space, thereby eliminating the VAE-induced quality constraints.

Pixel Diffusion Models. Recent years have seen renewed interest in pixel space diffusion models that aim to maximize signal fidelity while addressing computational inefficiency. Early works such as ADM [9] and DDPM [19] demonstrated the power of diffusion but were constrained by the quadratic complexity of their backbones, rendering them impractical for high resolutions. Multi-scale and image patch-based methods [5, 10, 21, 22, 22] further enhance the generation effect by decomposing large images into small patches. However, these methods essentially simulate locality through brute-force training, which leads to extremely low efficiency. Concurrent work JiT [32] demonstrates that high-dimensional data in pixel space can be effectively modeled by predicting clean images. Recent work by PixelNerd [54] leverages a Transformer to process image features, which then conditions a NeRF-like coordinate network to act as a renderer for finely reconstructing each image patch, achieving impressive performance. Nevertheless, PixelNerd tightly couples the success of its method with this specific NeRF-like rendering mechanism, which may limit the exploration of a broader design space. We argue that the key to achieving efficient and high-quality pixel space generation lies not in relying on a specific structure like NeRF, but rather in the design principle of decoupling global structure construction from local detail refinement. Based on this insight, this paper aims to provide a more principled, efficient, and general solution for pixel space diffusion models.

3. Methods

3.1. Preliminaries

A diffusion process gradually perturbs an initial data sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ from the true data distribution into isotropic Gaussian noise:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{1}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. $\{\beta_t\}_{i=1}^{T}$ is a predefined variance schedule that controls the noise level at each step. As $t \to T, \bar{\alpha}_t \to 0$, and the distribution of $\mathbf{x}_T$ converges to a standard normal distribution $p(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$.

This discrete formulation can be generalized to a continuous-time setting via a stochastic differential equation (SDE):

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \tag{2}$$

where $f(\cdot, t)$ is the drift and $g(t)$ is the diffusion coefficient.

The trajectory of this reverse process is governed by a corresponding probability flow ordinary differential equation (ODE):

$$d\mathbf{x} = \left[f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt. \tag{3}$$

Learning to generate data is thus equivalent to learning the score function $\log p_t(\mathbf{x})$ or the associated vector field of this ODE. To train a neural network for this task, several objectives have been proposed. DDPM trains a model $\epsilon_\theta(\mathbf{x}_t, t)$ to predict the noise component θ from a noisy sample x_t:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon}\left[\left\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\right\|^2\right]. \tag{4}$$

Flow Matching (FM) [12] provides a simulation-free paradigm for directly learning the vector field. It defines a conditional probability path $p_t(\mathbf{x} \mid \mathbf{x}_0)$ and a corresponding target vector field $u_t(\mathbf{x})$. A network $v_\theta(\mathrm{x}, t)$ is then trained to regress this field by minimizing the loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, p_t(\mathbf{x}|\mathbf{x}_0)}\left[\left\|u_t(\mathbf{x}) - v_\theta(\mathbf{x}, t)\right\|^2\right]. \tag{5}$$

3.2. Motivation

DiT models the long-range dependencies of an image by partitioning it into a sequence of patches, thereby forming a coherent global structure. However, while the self-attention mechanism excels at modeling macroscopic relationships between patches, it compresses the rich spatial information within each patch into a single, flattened token. This design introduces an inherent limitation: the model can adeptly learn the coarse-level layout and arrangement of patches but struggles to model the fine-grained textures and high-frequency details within each patch, consequently limiting the upper bound of its image generation performance.

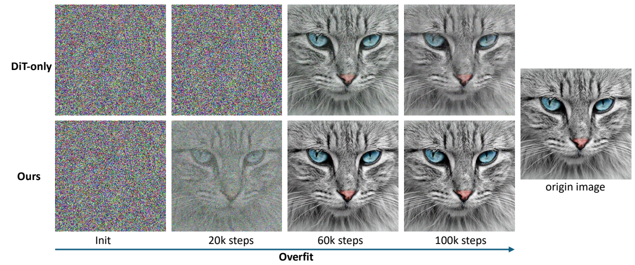

Figure 3. Overfitting the DiT-only model using a single image in pixel space leads to poor detail reconstruction. Introducing a local inductive bias achieves better reconstruction and accelerates convergence. Please zoom in for details.

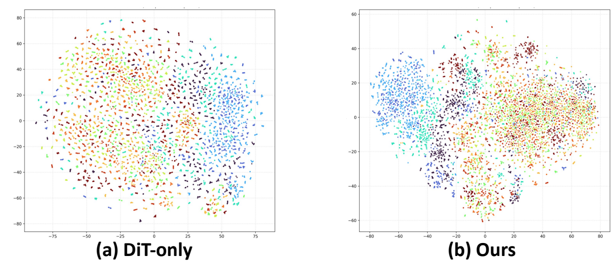

Figure 4. The t-SNE visualization of feature space. In the ImageNet validation set, 100 samples were randomly selected from each of the 10 classes for feature visualization. Features are extracted using DiT-only and our method, with each class shown in a distinct color.

To empirically validate this, we conduct a preliminary experiment by overfitting a DiT model on a single high-resolution image in the pixel space. As shown in Figure 3, the model successfully captures the global layout and color palette but fails to render fine textures and sharp edges, resulting in a blurry reconstruction. This result demonstrates that when a DiT architecture operates directly on images, it suffers from a lack of inductive bias [1, 15, 24, 57] at the local level, rendering it incapable of achieving precise pixel-level reconstruction within each patch.

This motivates our core design principle: to augment the global Transformer with a dedicated module that explicitly re-injects this missing inductive bias for local details. In this way, our model can leverage the computational efficiency afforded by large patch sizes while simultaneously generating high-quality images with fine-grained details. As shown in Figure 4, our method achieves tighter intra-class clusters and clearer inter-class separation, whereas vanilla DiT exhibits more mixed distributions. This means that the introduction of local inductive bias can more effectively integrate local textures and edge cues in pixel space and thus improves high-level semantic separability and feature con-

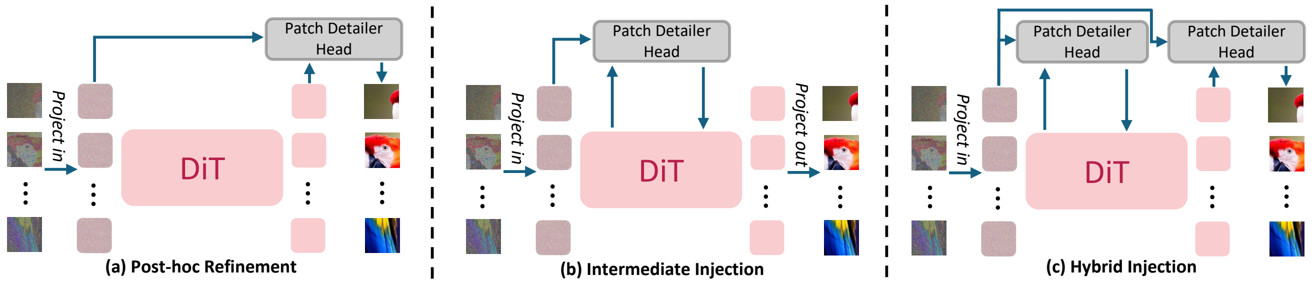

Figure 5. Patch Detailer Head with local inductive bias was placed at different locations in the model. The results in Sec. 4.3 show that all three methods offer gains compared to DiT-only.

sistency. Such improvements are expected to yield more stable structural alignment and better detail during generation process.

3.3. Framework

Based on the above observation, we introduce a framework for high-quality image generation that operates directly in pixel space. DiP first employs a DiT to model the global structure and long-range dependencies of the image. Subsequently, a lightweight Patch Detailer Head refines the output at the patch level, introducing a local inductive bias to synthesize high-frequency details.

Global Structure Construction (DiT Backbone). Given a noisy image $x_t \in \mathbb{R}^{H\times W\times 3}$ at timestep t, we first partition it into a sequence of non-overlapping patches. Each patch has a size of $P\times P$ (we set P=16), resulting in a sequence of $N=(H\times W)/P^2$ patches. This patching strategy ensures our pixel space model maintains a computational footprint comparable to latent space DiT models. Along with a timestep embedding and positional embeddings, they are fed into a series of DiT blocks to produce a sequence of context-aware output features $S_{\text{global}} \in \mathbb{R}^{N\times D}$, where D is the feature dimension.

Local Detail Refinement (Patch Detailer Head). The Patch Detailer Head operates independently and in parallel on each patch. For each patch i, it takes two inputs: the corresponding global context map s_i and the original noisy pixel patch $p_i \in \mathbb{R}^{3\times P\times P}$, where $s_i \in \mathbb{R}^{D\times 1\times 1}$ is obtained by reshaping and expanding S_{global}. Its objective is to leverage the global context from S_{global} to accurately interpret the local noisy information in p_i, ultimately predicting the corresponding noise component $\epsilon_i \in \mathbb{R}^{3\times P\times P}$ for that patch. After processing all N patches in parallel, the resulting sequence of predicted noise patches $\{\epsilon_i\}_{i=1}^{N}$ is reassembled into a full-resolution noise prediction map.

3.4. Architecture Design

Exploring Patch Detailer Head Architectures. We investigated several architectures for the Patch Detailer Head, each embodying a different form of inductive bias. Our goal is to present a simple, effective and highly efficient design.

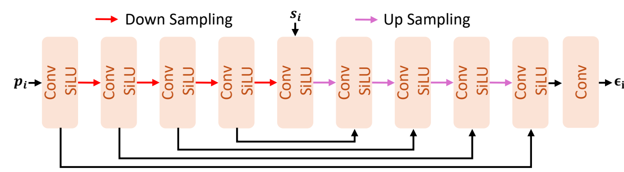

Figure 6. Patch Detailer Head framework. This design introduces the local inductive bias that DiT-only lacks with a low number of parameters, resulting in a high-quality image with rich detail.

- *Standard MLP.* As a simple baseline, we used MLP that takes the feature vector s_i and a flattened noisy patch p_i as input. While straightforward, this design lacks any inherent spatial bias, treating all pixels within the patch as an unordered set.
- *Coordinate-based MLP.* To introduce spatial awareness, a design inspired by NeRF can be adopted [54]. For each pixel within p_i, we concatenate its normalized 2D coordinates. s_i is used to dynamically generate the weights of a small, coordinate-based MLP. This implicitly learns a continuous function of the image patch, but it lacks the strong priors for local texture and structure that convolutions provide.
- *Intra-Patch Attention.* We explored using a small Transformer to operate on the pixels within each patch. Each $P\times P$ patch is treated as a sequence of P^2 pixel tokens. This allows for complex, content-aware interactions between pixels but is computationally intensive and may not be as efficient as convolutions for learning local patterns.
- *Convolutional U-Net (Our Final Choice).* We found that a lightweight convolutional U-Net provided the best performance. The hierarchical structure of downsampling and upsampling paths, combined with skip connections, is exceptionally well-suited for capturing multi-scale spatial features and ensuring local continuity. The inherent inductive biases of convolutions (locality and translation equivariance) are highly effective for denoising local textures and edges. As shown in Figure 6, we instantiate the Patch Detailer Head with a shallow U-Net, which includes 4 downsampling and 4 upsampling blocks. Each block consists of a sequence of Convolution, SiLU activation and pooling layer. The global feature vector s_i is

Method	ImageNet 256×256								
	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	Latency↓	Epochs	NFE	Params
Latent Generative Models									
LDM [41]	3.60	-	247.7	0.87	0.48	-	170	250x2	400M+86M
DiT-XL [37]	2.27	4.60	278.2	0.83	0.57	2.09s	1400	250x2	675M+86M
MaskDiT-G [71]	2.28	5.67	276.6	0.80	0.61	-	1600	79x2	675M+86M
SiT-XL [33]	2.06	4.50	270.3	0.82	0.59	2.09s	1400	250x2	675M+86M
FlowDCN-XL [53]	2.00	4.33	263.1	0.82	0.58	-	400	250x2	618M+86M
Pixel Generative Models									
CDM [20]	4.88	-	158.7	-	-	-	2160	4100	-
ADM [9]	3.94	6.14	215.8	**0.83**	0.53	15.80s	400	500	554M
JetFormer-L [50]	6.64	-	-	0.69	0.56	-	500	-	2.8B
SiD [21]	2.77	-	211.8	-	-	-	800	250×2	2.0B
VDM++ [28]	2.12	-	278.1	-	-	-	-	250×2	2.46B
RIN [23]	3.42	-	182.0	-	-	-	480	1000	410M
Farmer/16 [70]	3.96	-	250.6	0.79	0.50	-	320	-	1.9B
PixelFlow-XL/4 [5]	1.98	5.83	282.1	0.81	0.60	7.50s	320	120x2	677M
DiP-XL/16	2.16	4.79	276.8	0.82	0.61	0.92s	160	100x2	631M
DiP-XL/16	1.98	**4.57**	**282.9**	0.80	0.62	**0.70s**	320	75x2	631M
DiP-XL/16	**1.79**	4.59	281.9	0.80	**0.63**	0.92s	600	100x2	631M

Table 1. Comparison of the performance of different methods on ImageNet 256×256 with Euler solver and CFG. Performance metrics are annotated with ↑ (higher is better) and ↓ (lower is better). Our method achieves the best FID score. Furthermore, compared to other pixel diffusion models, we achieve the best performance across all metrics with the lowest latency.

concatenated channel-wise with the downsampling output at the bottleneck. This design allows the global semantic information to guide the local refinement process effectively while keeping the parameter count minimal.

We provide experimental evidence for this part in Sec. 4.3. Furthermore, we present a preliminary theoretical analysis in Appendix to model the necessity and effectiveness of the Patch Detailer Head, aiming to offer deeper insights.

Placement of the Patch Detailer Head. Since the main weakness of DiT backbone being trained directly in pixel space is its lack of local awareness, a natural design question arises: does introducing Patch Detailer Head at different locations in the model also bring gains? As shown in Figure 5, we investigated three placement strategies:

- *Post-hoc Refinement.* The Patch Detailer Head is placed only after the final DiT block. This creates a clean separation of concerns: the DiT is solely responsible for global modeling, and Patch Detailer Head is solely responsible for local refinement.
- *Intermediate Injection.* The Patch Detailer Head is inserted between DiT blocks. The refined patch representations are then projected back and fed into the subsequent DiT blocks.
- *Hybrid Injection.* Patch Detailer Head are placed both at an intermediate stage and at the end of the DiT.

Our experiments in Sec. 4.3 revealed that all three strategies yield comparable performance gains over the baseline DiT. However, the Post-hoc Refinement strategy has a unique advantage: by placing the Head at the end, we treat the standard DiT architecture as a fixed, black-box backbone. This approach requires no modification to the DiT's internal structure, greatly simplifying implementation and potentially allowing for the use of pre-trained DiT checkpoints. Given its optimal balance between high performance and implementation simplicity, we adopt the post-refinement strategy as the final architecture.

4. Experiments

4.1. Setup

Implementation Details. Our experiments are conducted on the class-conditional ImageNet dataset and original images are center-cropped and resized to 256×256 resolution. We set global batch size to 256. We use DDT [55], a variant of DiT, as our model backbone and apply an Exponential

Figure 7. Qualitative samples from our model trained at 256 × 256 resolution with classifier-free guidance scale of 4.0. DiP demonstrates fine-grained detail, and high visual quality.

Moving Average (EMA) on the model weights with a decay factor of 0.9999. In Patch Detailer Head, the kernel size of the middle layers is set to 3, the padding to 1, and the kernel size of the last convolutional layer is set to 1. Unless otherwise specified, all samples were generated using the Euler-100 solver. More details are included in Appendix.

Evaluation Protocol. To ensure a comprehensive and rigorous assessment of our model's generative capabilities, we adhere to the evaluation protocol established by ADM [9]. We employ a suite of standard quantitative metrics to measure performance across different dimensions. Specifically, we use the Fréchet Inception Distance (FID) [17] to assess overall realism and fidelity, the Spatial FID (sFID) [36] to evaluate spatial and structural coherence, and the Inception Score (IS) [44] to measure class-conditional diversity. Furthermore, we report Precision (Prec.)/Recall (Rec.) [30] to respectively quantify the fidelity of individual samples and the model's ability to cover the true data distribution. All metrics are calculated using 50,000 generated samples.

4.2. Main Result

Performance. Table 1 presents a comprehensive comparison against recent SOTA methods with classifier-free guidance scheduling with guidance interval [31]. After 600 training epochs, DiP achieved an FID of 1.79 without requiring a pre-trained VAE, surpassing potentially diffusion models such as DiT-XL (FID 2.27) and SiT-XL (FID 2.06), which require longer training times. DiP outperforms the previous best pixel-based model, PixelFlow-XL/4 (FID 1.98), and significantly exceeds others like ADM (FID 3.94) and VDM++ (FID 2.12). Even with a shorter training schedule of 160 epochs, our model reaches a competitive FID of 2.16, outperforming established models like DiT-XL that require much longer training.

Figure 7 presents qualitative samples of DiP at 256×256 resolution. These visualizations reveal rich detail, demonstrating the effectiveness of introducing local inductive bias. More visualization samples are provided in Appendix.

Computational Cost Comparison. DiP's parameter count (631M) is significantly smaller than other pixel models, such as VDM++ (2.0B) and Farmer (1.9B). DiP reaches its best performance with only 320 epochs, which is over $4\times$ more efficient than DiT-XL and SiT-XL (1400 epochs) and substantially faster than many other pixel-based methods like CDM (2160 epochs). In single-image inference speed tests, DiP (0.92s) is more than $2.2\times$ faster than DiT-XL (2.09s) and more than $8\times$ faster than the previous best pixel model, PixelFlow-XL (7.50s). Furthermore, in 75-step inference, DiP (0.70s) achieved the same FID score as PixelFlow-XL with a speed more than $10\times$ faster.

Method	ImageNet 256×256							
	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	Training Cost	Latency	Params
Scaling Up DiT								
DiT-only (26 Layers, 1152 Hidden Dim)	5.28	6.56	243.8	0.74	0.55	84×8 GPU Hours	0.88s	629M
DiT-only (32 Layers, 1152 Hidden Dim)	4.91	6.44	251.7	0.74	0.56	103×8 GPU Hours	1.05s	772M
DiT-only (26 Layers, 1280 Hidden Dim)	4.28	6.26	249.6	0.77	0.56	103×8 GPU Hours	1.06s	776M
DiT-only (26 Layers, 1536 Hidden Dim)	2.83	5.16	**285.6**	0.80	0.57	149×8 GPU Hours	1.49s	1.1B
Different Patch Detailer Head								
Standard MLP	6.92	7.27	210.9	0.79	0.41	93×8 GPU Hours	0.91s	630M
Intra-Patch Attention	2.98	5.16	275.0	0.80	0.56	96×8 GPU Hours	0.94s	630M
Coordinate-based MLP	2.20	4.49	284.6	0.80	0.58	123×8 GPU Hours	0.95s	700M
Convolutional U-Net	**2.16**	**4.79**	276.8	**0.82**	**0.61**	92×8 GPU Hours	0.92s	631M

Table 2. Impact of different design schemes on computational overhead and performance.

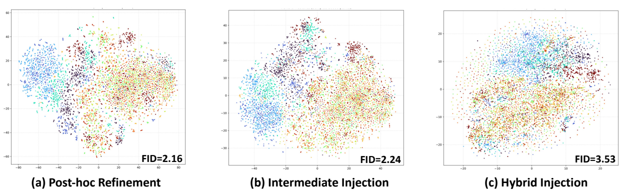

Figure 8. The t-SNE visualization of feature space. Features are extracted using Post-hoc Refinement, Intermediate Injection, and Hybrid Injection, with each class shown in a distinct color.

4.3. Analysis

In this section, we analyzed the trade-off between generation quality and computational cost during the development of DiP, and at the same time explained the rationality of the Patch Detailer Head we designed.

Patch Detailer Head vs. Scaling Up DiT. A common strategy to improve generative models is to increase the model size. However, our findings indicate that this is a suboptimal approach for pixel space diffusion models. As shown in Table 2, increasing the DiT's depth from 26 to 32 layers yields only a marginal improvement (FID from 5.28 to 4.91) at a considerable cost in parameters and training time. It also means that the effectiveness of our Patch Detailer Head comes from the introduction of effective local inductive biases, rather than increasing network depth.

In contrast, widening the model proves more effective for quality improvement. For instance, scaling the hidden dimension to 1536 reduces the FID to 2.83. This substantial quality gain comes at a prohibitive cost: a 74.9% increase in parameters (from 629M to 1.1B), a 77.4% rise in training cost (from 84×8 to 149×8 GPU hours), and a 69.3% (from 0.88s to 1.49s) increase in inference latency. This highlights a critical challenge with monolithic scaling, where significant computational resources are required for performance.

Experimental Results of Exploring Patch Detailer Head Architectures. We further investigated different architectures for the Patch Detailer Head to understand the importance of inductive bias in local patch refinement. (1) The Standard MLP performs poorly (6.92 FID), even worse than the DiT-only baseline. This is expected, as it lacks any spatial inductive bias, treating patch pixels as an unordered set and failing to capture crucial local structures. (2) The Intra-Patch Attention shows a significant improvement over the MLP (FID 2.98). This indicates that content-aware relationships between pixels are valuable. Its training and inference costs are only slightly higher than our final choice, but its actual memory overhead is about twice that of the final solution. (3) The Coordinate-based MLP achieves 2.20 FID (we are based on a reproduction of [54]). By explicitly conditioning on pixel coordinates, it effectively introduces spatial awareness. However, it requires more parameters (700M) and a longer training time (123×8 GPU hours) compared to our final choice, and its implicit continuous representation may lack the strong, built-in priors for local patterns that convolutions provide. (4) The Convolutional U-Net increases the number of parameters by only 0.3% (from 629M

Method	ImageNet 512×512					
	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑	Params
Latent Generative Models						
DiT-XL [37]	3.04	5.02	**0.84**	0.54	240.8	675M+86M
MaskDiT-G [71]	2.50	5.10	0.83	0.56	256.3	675M+86M
SiT-XL [33]	2.62	**4.18**	**0.84**	0.57	252.2	675M+86M
FlowDCN-XL [53]	2.44	4.53	**0.84**	0.54	252.8	618M+86M
Pixel Generative Models						
ADM [9]	3.85	5.86	**0.84**	0.53	221.7	554M
SiD [21]	3.02	-	-	-	248.7	2.00B
VDM++ [28]	2.65	-	-	-	278.1	2.46B
RIN [23]	3.95	-	-	-	216.0	410M
DiP-XL/32	**2.31**	4.48	**0.84**	**0.58**	**291.68**	631M

Table 3. Comparison of the performance of different methods on ImageNet 512×512 with CFG. Performance metrics are annotated with ↑ (higher is better) and ↓ (lower is better). Our method remains competitive at higher resolutions.

Figure 9. Qualitative samples from our model trained at 512×512 resolution with classifier-free guidance scale of 4.0. DiP showcases fine-grained detail and rich diversity at higher resolutions.

to 631M) and achieves the best FID score with the lowest computational cost among all Patch Detailer Head. Its success can be attributed to highly relevant inductive biases of convolutions. It is well-suited for capturing and preserving the continuity of local textures and edges, making it the most efficient and effective architecture for performing patch-level detail optimizations.

In summary, our experimental results clearly demonstrates that introducing an appropriate local inductive bias via a Patch Detailer Head is key to performance improvement over the scaling up DiT baseline. Among the architectures explored, the Convolutional U-Net strikes the optimal balance between best generation quality and minimal computational cost, making it our definitive choice.

Experimental Results of Placement of the Patch Detailer Head. We tested the effects of introducing Patch Detailer Head at different locations in the model. As shown in Figure 8, all three introduction modes showed significant improvements compared to DiT-only (FID 5.28). From the feature visualization results, Hybrid Injection performed worse in clustering than the other two, which may be due to multiple local inductive biases potentially disrupting the original structure, leading to performance degradation. Post-hoc Refinement achieves the best performance, a result attributed to the synergy between global build and local refinement, while its implementation is simple and easily extensible.

4.4. Ablation Study

Performance on ImageNet 512×512. As shown in Table 3, on 512×512 resolution, DiP achieved the best FID score , surpassing previous methods. Figure 9 illustrates the sampling results at 512×512, demonstrating that DiP can also generate high-quality images. We present more qualitative results in Appendix.

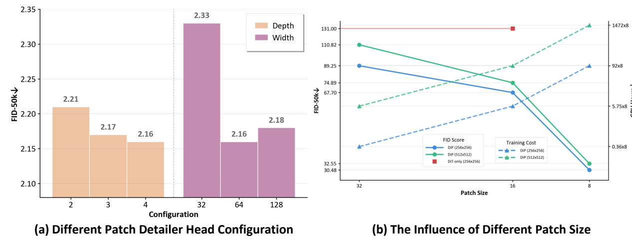

Figure 10. (a). Performance differences between different Patch Detailer Head configurations. Depth is defined as the number of down/up-sampling stages, and width corresponds to the number of base channels in the convolutional layers. (b). Performance and computational overhead differences of different patch sizes.

Impact of Patch Detailer Head Configuration. Our findings reveal distinct trends for depth and width. As we increase the depth, we observe a consistent improvement in generation quality, although the gains exhibit diminishing returns and eventually saturate. This suggests that a multi-scale feature hierarchy is crucial for the Patch Detailer Head to effectively synthesize high-frequency details across. Conversely, blindly increasing the width will not lead to a sustained performance improvement. This indicates that the role of Patch Detailer Head is not to perform complex feature transformations, but rather to render specific details.

Impact of Patch Size. Small patch size offer superior performance but also significantly increase computational overhead. Our method can use large patch to shorten the input sequence length, making our model's computational efficiency comparable to mainstream LDMs. As shown in Figure 10(b), at the higher 512×512 resolution, DiP maintains a significant performance margin over a DiT-only baseline using smaller patch size. This validates that our approach provides a robust solution for efficient, high-resolution synthesis directly in pixel space.

5. Conclusion

In this paper, we addressed the fundamental trade-off between generation quality and computational efficiency in pixel diffusion models. We introduced DiP, a new end-to-end pixel space diffusion framework that resolves this dilemma through a synergistic global-local modeling approach. By employing a DiT backbone on large image patches, we achieve computational efficiency comparable to LDMs for modeling global structures. This is complemented by a co-trained, lightweight Patch Detailer Head that expertly restores high-frequency details, effectively bypassing the need for a VAE. Our extensive experiments on the ImageNet benchmark demonstrate that achieves superior FID scores with significantly lower inference latency and training costs. In the future, we plan to apply the DiP framework to text to image and text to video tasks to further explore the capabilities of this solution.

Acknowledgment

This work was supported by Natural Science Foundation of Jiangsu Province: BK20241198, the Gusu Innovation and Entrepreneur Leading Talents: No. ZXL2024362 and Natural Science Foundation of China: No. 62406135.

References

[1] Jie An, De Wang, Pengsheng Guo, Jiebo Luo, and Alex Schwing. On inductive biases that enable generalization in diffusion transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2024. 3

[2] BlackForest. Black forest labs; frontier ai lab, 2024. 2

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 1, 2

[4] 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. *arXiv preprint arXiv:2410.10733*, 2024. 1

[5] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025. 1, 2, 5

[6] Zhennan Chen, Rongrong Gao, Tian-Zhu Xiang, and Fan Lin. Diffusion model for camouflaged object detection. In *ECAI 2023*, pages 445–452. IOS Press, 2023. 1

[7] Zhennan Chen, Yajie Li, Haofan Wang, Zhibo Chen, Zhengkai Jiang, Jun Li, Qian Wang, Jian Yang, and Ying Tai. Ragd: Regional-aware diffusion model for text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19331–19341, 2025.

[8] En Ci, Shanyan Guan, Yanhao Ge, Yilin Zhang, Wei Li, Zhenyu Zhang, Jian Yang, and Ying Tai. Describe, don't dictate: Semantic image editing with natural language intent. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19185–19194, 2025. 1

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2, 5, 6, 7

[10] Zheng Ding, Mengqi Zhang, Jiajun Wu, and Zhuowen Tu. Patched denoising diffusion models for high-resolution image synthesis. In *The twelfth international conference on learning representations*, 2023. 2

[11] Nikai Du, Zhennan Chen, Shan Gao, Zhizhou Chen, Xi Chen, Zhengkai Jiang, Jian Yang, and Ying Tai. Textcrafter: Accurately rendering multiple texts in complex visual scenes. *arXiv preprint arXiv:2503.23461*, 2025. 1

[12] 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. 2, 3

[13] Tiehan Fan, Kepan Nan, Rui Xie, Penghao Zhou, Zhenheng Yang, Chaoyou Fu, Xiang Li, Jian Yang, and Ying Tai. Instancecap: Improving text-to-video generation via instance-aware structured caption. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28974–28983, 2025. 1

[14] 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. 1

[15] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022. 3

[16] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pages 393–411. Springer, 2024. 1

[17] 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. 6

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[20] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 5

[21] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023. 1, 2, 5, 7

[22] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024. 1, 2

[23] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022. 5, 7

[24] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. *arXiv preprint arXiv:2310.02557*, 2023. 3

[25] 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. 1

[26] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024. 1

[27] Maciej Kilian, Varun Jampani, and Luke Zettlemoyer. Computational tradeoffs in image synthesis: Diffusion,

masked-token, and next-token prediction. *arXiv preprint arXiv:2405.13218*, 2024. 1

[28] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36: 65484–65516, 2023. 1, 5, 7

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[30] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 6

[31] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024. 6

[32] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. 2

[33] 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. 5, 7

[34] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[35] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024. 1

[36] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 6

[37] 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. 1, 2, 5, 7

[38] 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. *arXiv preprint arXiv:2307.01952*, 2023. 2

[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1

[40] 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. 1

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 5

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[43] 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. 1

[44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6

[45] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025. 2

[46] 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. 1

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[48] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[49] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023. 1

[50] Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. *arXiv preprint arXiv:2411.19722*, 2024. 5

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[52] Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024. 1

[53] 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 *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 87959–87977, 2024. 5, 7

[54] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025. 2, 4, 7

[55] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025. 5

[56] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Yujun Lin, Zhekai Zhang, Muyang Li, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 1, 2

[57] Tao Yang, Cuiling Lan, Yan Lu, and Nanning Zheng. Diffusion model with cross attention as an inductive bias for disentanglement. *Advances in Neural Information Processing Systems*, 37:82465–82492, 2024. 3

[58] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025. 1, 2

[59] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 1

[60] 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. 1

[61] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[62] Xuying Zhang, Xiaoshuai Sun, Yunpeng Luo, Jiayi Ji, Yiyi Zhou, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Rstnet: Captioning with adaptive attention on visual and non-visual words. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15465–15474, 2021. 1

[63] Xuying Zhang, Yutong Liu, Yangguang Li, Renrui Zhang, Yufei Liu, Kai Wang, Wanli Ouyang, Zhiwei Xiong, Peng Gao, Qibin Hou, et al. Tar3d: Creating high-quality 3d assets via next-part prediction. *arXiv preprint arXiv:2412.16919*, 2024. 1

[64] Xuying Zhang, Bo-Wen Yin, Yuming Chen, Zheng Lin, Yunheng Li, Qibin Hou, and Ming-Ming Cheng. Temo: Towards text-driven 3d stylization for multi-object meshes. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 19531–19540, 2024.

[65] Xuying Zhang, Yupeng Zhou, Kai Wang, Yikai Wang, Zhen Li, Shaohui Jiao, Daquan Zhou, Qibin Hou, and Ming-Ming Cheng. Ar-1-to-3: Single image to consistent 3d object generation via next-view prediction. *arXiv preprint arXiv:2503.12929*, 2025. 1

[66] Chen Zhao, Weiling Cai, Chenyu Dong, and Chengwei Hu. Wavelet-based fourier information interaction with frequency diffusion adjustment for underwater image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8281–8291, 2024. 1

[67] Chen Zhao, Weiling Cai, Chengwei Hu, and Zheng Yuan. Cycle contrastive adversarial learning with structural consistency for unsupervised high-quality image deraining transformer. *Neural Networks*, 178:106428, 2024. 1

[68] Chen Zhao, En Ci, Yunzhe Xu, Tiehan Fan, Shanyan Guan, Yanhao Ge, Jian Yang, and Ying Tai. Ultrahr-100k: Enhancing uhr image synthesis with a large-scale high-quality dataset. *Advances in Neural Information Processing Systems*, 2025. 1

[69] Chen Zhao, Jiawei Chen, Hongyu Li, Zhuoliang Kang, Shilin Lu, Xiaoming Wei, Kai Zhang, Jian Yang, and Ying Tai. Luve: Latent-cascaded ultra-high-resolution video generation with dual frequency experts. *arXiv preprint arXiv:2602.11564*, 2026. 1

[70] 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. *arXiv preprint arXiv:2510.23588*, 2025. 5

[71] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023. 5, 7

[72] Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation controller for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6818–6828, 2024. 1

[73] Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3dis: Depth-driven decoupled instance synthesis for text-to-image generation. *arXiv preprint arXiv:2410.12669*, 2024. 1

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1