LET THE PIXELS THROUGH: RETHINKING LONG SKIP CONNECTIONS FOR SCALABLE PIXEL DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training pixel diffusion models remains less straightforward, owing to the ad hoc training recipes and architecture designs, along with the sensitivity to hyper-parameter choices. We show that strong denoisers can be built by refining existing pixel-native designs from the literature. Specifically, we revisit the long skip connection in U-Net and find that the optimization benefit commonly attributed to it is not necessary for its denoising benefit. We further find that the conventional linear combination of features from the skip and main backbone is inadequate for regimes of various noise levels. Motivated by these observations, we introduce two core modifications. First, we adopt asymmetric long skips, which propagate fine-grained information from input embeddings to decoder layers. Second, we introduce adaGLU, a SwiGLU-like MLP block, in decoder layers to adaptively filter information from the embeddings before being added to main-branch features. The above designs give a simple pixel diffusion transformer architecture. When trained from scratch, it scales smoothly from small-scale baselines up to 1.6B parameters, achieving 1.62 and 1.70 FID on ImageNet-256 and -512, respectively. Besides, it natively supports multimodal inputs, such as text and images, yielding competitive performance in text-to-image generation.

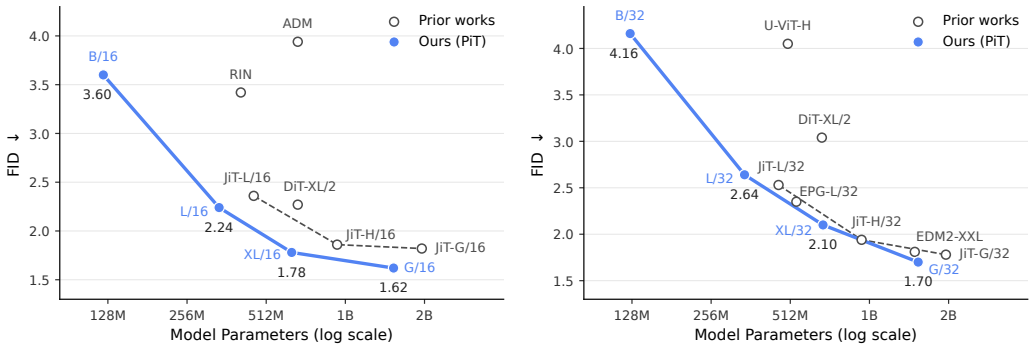
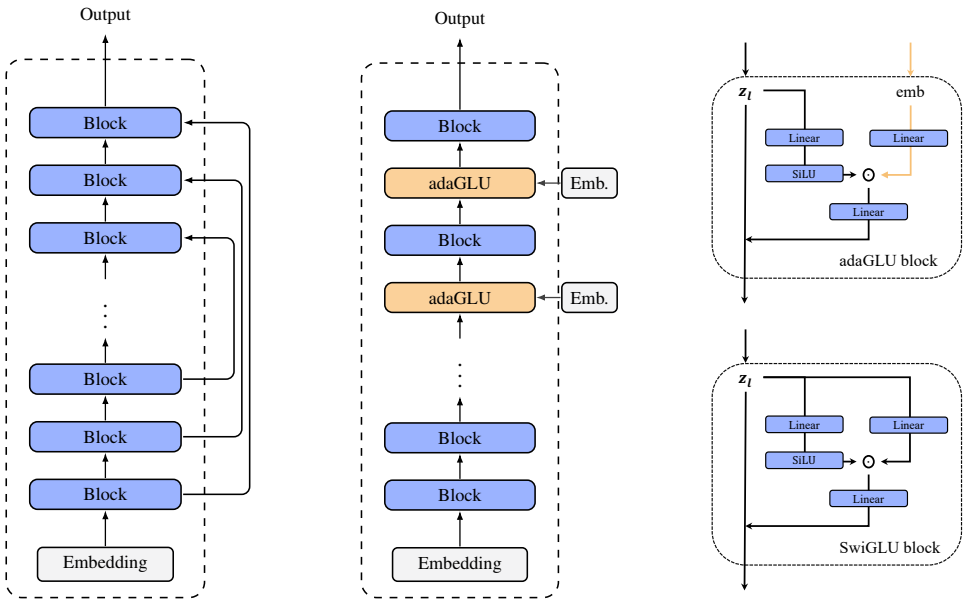


Figure 1: Performance of our models on **(left)** ImageNet-256 and **(right)** ImageNet-512. We use identical training hyper-parameters across all model sizes and resolutions.

1 INTRODUCTION

Pixel-space diffusion models have recently received renewed attention. A key advantage is that they avoid the computational overhead brought by a separate tokenizer in latent diffusion models (LDM) (Rombach et al., 2022), thus being attractive for high-dimensional generation. Recent work has shown that strong pixel-space diffusion can be obtained with relatively simple transformer backbones (Vaswani et al., 2017; Dosovitskiy et al., 2021). Some methods (Li & He, 2025) train directly from pixels, but are sensitive to their architectural design and other hyper-parameters. The other (Yu et al., 2026; Lei et al., 2026) further introduce additional training stages or specialized architecture components. These designs achieve strong performance, but also raise a basic question: how much architectural structure is actually necessary for a strong pixel-space diffusion model?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) U-Net-style long skips. (b) Ours. (c) Comparing adaGLU with SwiGLU.

Figure 2: Overview of our model architecture. In **(a, b)**, we compare our architecture against the standard vision transformer (ViT) model with U-Net-style long-skip connections. The embeddings input to the adaGLU block are the same as that of the first ViT block. **(c)** we show that an adaGLU block shares the same architecture as a standard SwiGLU block.

We revisit a basic component, the long skip connection in U-Net (Ronneberger et al., 2015), a common design in diffusion models. In its standard form, features from shallow encoder layers are directly connected to corresponding decoder layers. The motivation is straightforward. Shallow features preserve fine-grained information that may otherwise be lost after passing through a deep semantic backbone. These long residuals are also commonly believed to facilitate optimization by providing a shortcut for gradient propagation. In U-Net-style designs, these two effects are coupled by the symmetric encoder-decoder topology. Moreover, skip features are typically concatenated with the backbone features and fused by a fixed linear projection. In this paper, we investigate how much of these structures are actually necessary for denoising, and suggest that a strong denoiser can be achieved with an improved design.

We first examine whether the optimization effect is essential. In specific, we detach the gradient pass through the skip connections and find that it does not actually hurt performance. This reveals that, to retain the benefit of the skip, the conventional pairwise encoder-decoder structure is not always necessary in all situations. In practice, the optimization issue only becomes visible when we increase model depth. However, we show that this issue can be properly addressed with standard weight initialization. Motivated by the gradient-detach experiment, we then remove the pairwise encoder-decoder connections and route the input embedding directly to decoder layers. Surprisingly, this asymmetric design further improves performance. These observations together suggest that optimization benefit commonly attributed to long skips is not essential for their denoising benefit. The long skip connections are better understood as a pathway for low-level information than as a solution for optimization difficulty.

Then, this raises another question about how the skip features should be combined with the deep features in the backbone. In diffusion, the answer may depend on the noise level. At low noise, the input retains considerable image information and is useful for reconstruction. At high noise, the same information is less reliable. Concatenation followed by a fixed linear layer, as in common practice, does not distinguish these cases. It lacks a basic gating mechanism to adapt to regimes at various noise levels, yet directly merges features by channel-wise linear combination. We therefore test nonlinear gating functions and find that a SwiGLU-style formulation (Shazeer, 2020; Dauphin et al., 2017) gives a substantial improvement. Motivated by this result, we repurpose the standard

SwiGLU as a gate module, which we call *adaGLU*, for features from the long skip pathway. Figure 2c compares its structure against a standard SwiGLU block. Specifically, adaGLU rectifies the skip feature with the activation from the main backbone. The skip feature provides low-level information, while the main-branch feature controls how much of this information is used. After this, we fuse features by simple addition.

The above modifications give a simple transformer architecture, which we call Pixel Diffusion Transformer **(PiT)**. We provides an overview of its structure in Figure 2b. Our model *relies on no tokenizer, no AdaLN, no in-context embedding, no pre-training, and no auxiliary loss*. We mainly conduct experiments on ImageNet-256 and ImageNet-512, referring to ImageNet-1K (Deng et al., 2009) at 256×256 and 512×512 resolution, respectively. When trained from scratch, it reaches 1.62 FID on ImageNet-256 and 1.70 FID on ImageNet-512. Besides, our network also natively supports multi-modal inputs by treating them uniformly as a token sequence. To show this, we train a text-to-image generation model at 512×512 resolution by concatenating text embeddings with image tokens as model input. We evaluate it on GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024), scoring 0.82 and 84.3, respectively.

Our observations do not imply that U-Net-style architectures are ineffective. Rather, they suggest that their structural assumptions are not necessary for retaining the useful function of long skips. We believe this provide a simpler way to reason about cross-layer information routing in diffusion denoisers. We summarize our contributions below.

1. We revisit the role of long skip connections in pixel diffusion and show that their denoising benefit can be separated from their advantage in optimization, allowing the conventional U-Net topology to be replaced by a simpler asymmetric information pathway. We reinterpret the long skip connection as a functional component rather than an architectural template.
2. We strengthen its feature fusion mechanism by introducing adaGLU. It repurposes the standard SwiGLU as a gating module, allowing the backbone features to modulate the low-level information carried by the skip.
3. We show that PiT performs strongly when trained from scratch across a wide range of model sizes, reaching 1.62 FID on ImageNet-256 and 1.70 FID on ImageNet-512. We further show that the same architecture naturally extends to text-to-image generation.

2 Related Work

Pixel-space Diffusion Model have recently regained attention. By removing the pretrained VAE tokenizer, they avoid the reconstruction bottleneck, while offering better efficiency in direct image modeling. Recent works can be roughly divided into two categories. The first group uses specialized training recipes (Chen et al., 2025c; Wang et al., 2025a; Hoogeboom et al., 2023; Lei et al., 2023). EPG (Lei et al., 2026) and USP (Chu et al., 2025) introduces self-supervised pretraining before diffusion training. Though achieving impressive performance, they suggest that competitive pixel diffusion often depends on extra training objectives/stages. In contrast, JiT (Li & He, 2025) offers a workaround by reshaping the DiT (Peebles & Xie, 2023) architecture and parameterizing the denoising model to directly predict clean images. When trained from scratch, it yields competitive performance to top-performing models. However, it is sensitive to hyper-parameter choices, such as in-context embedding and Dropout (Srivastava et al., 2014). In comparison, our model requires no specialized recipe, presenting clean scaling behavior without extensive hyper-parameter tuning.

Denoising Network. The diffusion model architecture has undergone a series of advancements from U-Net like CNNs (Song et al., 2021; Ho et al., 2020; Dhariwal & Nichol, 2021) to transformer-based architecture (Peebles & Xie, 2023; Bao et al., 2023a; Jabri et al., 2023; Tian et al., 2024). Recently, a notable trend is to incorporate shallow-to-deep information routing, such as DDT (Wang et al., 2025b), and PixelDiT. Such designs can be effective, but they also introduce additional architectural structure beyond a standard transformer. Another work, DAR (Xu et al., 2026), employs attention residual (Team et al., 2026) to replace the fixed U-Net-style skip connections with learnable long residuals. The drawback is the additional heavy cost of computing the adaptive weights for each residual outweighs the benefits, showing a poor trade-off. In contrast, our work examines how much architectural structure that pathway actually needs, given that shallow-to-deep information routing is

Table 1: We compare representative skip connection designs of prior diffusion models, including EDM2 (Karras et al., 2024), Simple Diffusion (SD) (Hoogeboom et al., 2023), and UniDiffuser (Bao et al., 2023b). We omit bias terms for simplicity. $\boldsymbol{h}, \boldsymbol{z} \in \mathbb{R}^d$ are features from the skip connection and the backbone, respectively. $\boldsymbol{W} \in \mathbb{R}^{d \times 2d}$ is a linear layer. LN is layer normalization (Ba et al., 2016).

	EDM2	SD	UniDiffuser	**Ours**
Gating $\boldsymbol{G}(\boldsymbol{h}, \boldsymbol{z})$	$\boldsymbol{h}$	$\boldsymbol{h}$	$\boldsymbol{h}$	adaGLU($\boldsymbol{h}$, RMSNorm($\boldsymbol{z}$))
Fusion $\boldsymbol{F}(\boldsymbol{h}, \boldsymbol{z})$	$\boldsymbol{W}$Concat($\boldsymbol{h}, \boldsymbol{z}$)	$\frac{1}{\sqrt{2}}$(LN($\boldsymbol{h}$) + LN($\boldsymbol{z}$))	LN($\boldsymbol{W}$Concat($\boldsymbol{h}, \boldsymbol{z}$))	$\boldsymbol{G}(\boldsymbol{h}, \boldsymbol{z}) + \boldsymbol{z}$
Skip topology	Symmetric	Symmetric	Symmetric	Asymmetric
Backbone	CNN	Hybrid	ViT	ViT

useful. We show that such a pathway can naturally evolve from the well-established U-Net-style long skip connection design.

3 Background

Diffusion Models (Ho et al., 2020; Song et al., 2021) are a class of generative models that learn to generate data by reversing a pre-defined forward diffusion process. Given a noise distribution of standard Gaussian distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ and the target data distribution $p_{data}(\boldsymbol{x})$, the forward process traverses the data towards the noise distribution We adopt the diffusion model of variance-exploding (VE) forward SDE defined in EDM (Karras et al., 2022): $d\boldsymbol{x} = \sqrt{2t}d\mathbf{w}$. Thus, the transition kernel from clean image $\boldsymbol{x}(0) \sim p_{data}(\boldsymbol{x})$ to $\boldsymbol{x}(t)$ is defined as $p(\boldsymbol{x}(t)|\boldsymbol{x}(0)) = \mathcal{N}(\boldsymbol{x}(t); \boldsymbol{x}(0), t^2\boldsymbol{I})$, $t \in [\sigma_{min}, \sigma_{max}]$. We choose $\sigma_{min} = 0.002$ and σ_{max} = 80. When $t = \sigma_{max}$, $\boldsymbol{x}(t)$ approximates the noise distribution $\mathcal{N}(0, \sigma_{max}^2\boldsymbol{I})$. The reverse process of the variance-exploding forward SDE is defined as

$$d\boldsymbol{x} = [-2t\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}(t))]dt + \sqrt{2t}d\bar{\mathbf{w}}. \tag{1}$$

Here, $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}(t))$ is the score function, and $\bar{\boldsymbol{w}}$ is the standard Wiener process that flows backward in time. The diffusion model trains a denoiser f_θ to approximate a reparameterization of $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}(t))$ in the pre-defined time interval. In specific, it's optimized with a simple MSE loss

$$\mathcal{L} = \mathbb{E}_{t, \boldsymbol{x}(0), \boldsymbol{x}(t)} \left[w(t) \| \boldsymbol{x}(0) - f_\theta(\boldsymbol{x}(t), t) \|_2^2 \right]. \tag{2}$$

$w(t)$ is a scalar weighting function. We adopt the above $\boldsymbol{x}$-prediction formulation throughout our work. To generate images, starting from $t = \sigma_{max}$, we iteratively solve the corresponding probability flow ordinary differential equation (PF ODE) (Song et al., 2021) of the reverse diffusion process in equation 1 with ODE solvers such as Euler-Maruyama or Heun (Karras et al., 2022). The PF ODE is depicted by

$$d\boldsymbol{x} = [-t\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}(t))]dt. \tag{3}$$

4 Method

Starting from a simple experiment that detaches the gradient from the skip connections, we examine how the skip connection design can be improved, and subsequently investigate the optimization issues that arise when scaling up the model.

4.1 Problem Formulation

Let $\boldsymbol{h}, \boldsymbol{z} \in \mathbb{R}^d$ denote the feature from the skip connection and the feature of the main branch respectively. The feature fusion mechanism in a network equipped with U-Net-style skip connections can be generally depicted as

$$\boldsymbol{z} = \boldsymbol{F}(\boldsymbol{G}(\boldsymbol{h}, \boldsymbol{z}), \boldsymbol{z}). \tag{4}$$

Here, $\boldsymbol{G}$ is the gating function and $\boldsymbol{F}$ is the fusion operator that combines the gated output and $\boldsymbol{z}$. In Table 1, we list the designs of skip connections in prior works. It suggests that they often use an identity gating function $\boldsymbol{G}(\boldsymbol{h}, \boldsymbol{z}) = \boldsymbol{h}$ and a linear fusion operator that combines $\boldsymbol{h}$ and $\boldsymbol{z}$ via channel-wise linear combination (concatenation + linear map). They leave the deep merging of features to the main backbone, thus paying less attention to the design of the fusion mechanism itself.

Table 2: Improve skip connection design.

	FID
JiT-B[1]	38.1
ViT-B	
+ symmetric skips	52.1
+ detach grad.	50.1
+ adaGLU	41.2
+ asymmetric skips	47.8
+ adaGLU	**36.2**

Table 3: Ablate gating function $G(\boldsymbol{h}, \boldsymbol{z})$. $\boldsymbol{W}_{bottleneck} \in \mathbb{R}^{d' \times d}$ and $d' < d$. $\boldsymbol{W}_1$ maps features back to $\mathbb{R}^d$, $\boldsymbol{W}_2, \boldsymbol{W}_3 \in \mathbb{R}^{D \times d}$ and $D > d$.

$G(\boldsymbol{h}, \boldsymbol{z})$	FID
$\boldsymbol{W}_{bottleneck}\boldsymbol{h}$	54.1
$\boldsymbol{W}_1(\boldsymbol{h} \cdot \text{SiLU}(\boldsymbol{z}))$	57.1
$\boldsymbol{W}_1(\boldsymbol{W}_2\boldsymbol{h} \cdot \text{Tanh}(\boldsymbol{W}_3\boldsymbol{z}))$	39.0
$\boldsymbol{W}_1(\boldsymbol{W}_2\boldsymbol{h} \cdot \text{Sigmoid}(\boldsymbol{W}_3\boldsymbol{z}))$	37.6
$\boldsymbol{W}_1(\boldsymbol{W}_2\boldsymbol{h} \cdot \text{SiLU}(\boldsymbol{W}_3\boldsymbol{z}))$	**36.2**

Settings. In discussions below, we adopt the design in Bao et al. (2023b) (third column in Table 1) as a starting point. By default, we use a 12-layer 768-dim ViT-B with a large patch size of 16×16 as the base model. Images, class labels, and time conditioning are treated equally as input tokens. We train all models for 200K steps with a batch size of 1024 on ImageNet-256. We report FID measured on 10k samples produced by an Euler ODE sampler with 50 sampling steps.

4.2 Streamline the Skip Connection Design

A common belief is that U-Net-style skip connections facilitate training by alleviating the gradient vanishing problem. To test this hypothesis, we compare two ViT-B models, both equipped with U-Net-style skip connections. One is trained as usual, while the other is trained by detaching the gradient from the skip connections. This forces the supervisory signal to flow exclusively backward through the main backbone. Surprisingly, the simple modification yields a slightly better FID score than the baseline (50.1 v.s. 52.1), suggesting that the commonly assumed role of skip connections in mitigating gradient vanishing may not be necessary in this scenario.

Based on this observation, we then remove the pair-wise encoder-decoder shortcuts and redistribute the skip connections so that only the input embeddings are input to decoder layers. This retains the shortcuts to preserve fine-grained details from the input. As shown in Table 2, this asymmetric design improves the FID from 52.1 to 47.8. These results indicate that the primary function of the long skip is to provide an information pathway from the input toward the deeper layers, rather than to facilitate optimization. This perspective also motivates a reconsideration of how skip features should be fused with the backbone.

Denoising fundamentally operates across varying noise levels, and the usefulness of the input signal varies accordingly. At low noise levels, the perturbed input $\boldsymbol{x}_t$ still preserves substantial image information and can therefore directly support image reconstruction. In this regime, routing this information directly into deep layers can reduce the burden on the backbone to reproduce an identity-like mapping. However, at high noise levels, the input is heavily corrupted and provides substantially less reliable information. In this regime, the network should be able to suppress the contribution of the skip pathway when it is not useful.

To implement such adaptive routing, we compare several gating functions $\boldsymbol{G}(\boldsymbol{h}, \boldsymbol{z})$ in Table 3, ranging from a bottleneck linear layer to nonlinear MLP-based designs. The bottleneck linear layer reduces the dimensionality of the skip features as an information bottleneck, whereas the nonlinear variants allow the skip representation to be modulated by the semantics encoded in the backbone features. Among them, the SwiGLU formulation achieves the lowest FID. Specifically, we define

$$\text{adaGLU}(\boldsymbol{h}, \boldsymbol{z}) = \boldsymbol{W}_1(\boldsymbol{W}_2\boldsymbol{h} \cdot \text{SiLU}(\boldsymbol{W}_3\boldsymbol{z})). \tag{5}$$

Here, $\boldsymbol{W}_1 \in \mathbb{R}^{d \times D}$, $\boldsymbol{W}_2 \in \mathbb{R}^{D \times d}$, $\boldsymbol{W}_3 \in \mathbb{R}^{D \times d}$, $D > d$. Based on our current settings in Table 2, the feature fusion with adaGLU can be depicted by

$$\boldsymbol{z} = \text{Norm}\Big(\boldsymbol{W}\text{Concat}\big(\text{adaGLU}(\boldsymbol{h}, \boldsymbol{z}), \boldsymbol{z}\big)\Big). \tag{6}$$

$\boldsymbol{W} \in \mathbb{R}^{d \times 2d}$. We name this formulation as Post-Norm adaGLU as it mirrors the structure of a Post-Norm ViT block, where the normalization layer is applied at the end of computation. The term

[1]We reproduce with official implementation (https://github.com/LTH14/JiT), while replacing the fp32 attention with flash-attn in bf16 for training efficiency.

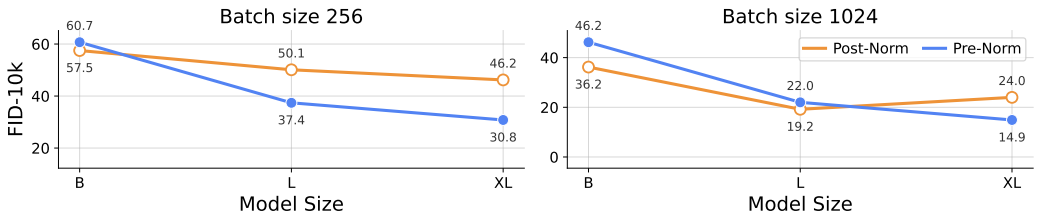


Figure 3: Comparing performance of Pre-Norm and Post-Norm adaGLU formulations across model sizes and training batch sizes. When trained with 1024 batch size, the Post-Norm adaGLU exhibits FID spikes at XL model size.

SiLU($W_3\boldsymbol{z}$) provides a semantic-dependent gate that modulates the information carried by the skip feature $\boldsymbol{h}$. In this way, the backbone representation determines how much of the low-level information should be propagated at a given denoising state. As shown in Table 2, the adaGLU substantially improves generation performance over the standard identity gating function in both the symmetric and asymmetric skip topology settings.

4.3 Scale up model size

We observe that the optimization bottlenecks are exposed when further increasing model size. In Table 4, when we initially increase the depth from 12 to 20 layers, the nearly doubled parameters yield only a marginal improvement. This suggests that the deeper model is not effectively exploiting its additional capacity. We attribute this degradation to the accumulation of activation variance in deep residual networks. A residual block can be written as $\boldsymbol{z}_{l+1} = \boldsymbol{z}_l + f_l(\boldsymbol{z}_l), l \in [0, L-1]$, L is model depth. Assume the residual branch is uncorrelated with the main stream activations at initialization. Under conventional parameter initializations, the residual branch $f_l(\boldsymbol{z}_l)$ contributes non-negligible accumulated variance, leading to $\text{Var}(\boldsymbol{z}_l) + \text{Var}(f_l(\boldsymbol{z}_l))$. Consequently, the activation magnitude can grow with depth, degrading performance in sufficiently deep networks. In gradient backpropagation, it can progressively amplify or attenuate different gradient directions. Thus, without controlling the scale of the residual branches, both the forward activations and backward gradients become increasingly poorly conditioned, making deeper models harder to optimize.

Table 4: Address optimization bottleneck with proper weight init.

model	FID
12-layer	36.2
20-layer w/o init	35.0
20-layer w init	**22.0**

To address this, we adopt a variance-preserving initialization strategy. Specifically, we follow standard practice (Glorot & Bengio, 2010; He et al., 2015) and initialize weights of linear layers by independently sampling from a Gaussian distribution $\mathcal{N}(0, 1/d_{in})$, where d_{in} is the input dimension of each layer. Then, we scale the output matrices of MLP and attention blocks by $\frac{1}{\sqrt{2*L}}$ (Marion et al., 2025; Zhang et al., 2019), where L is the model depth. For each adaGLU block, we zero-initialize the output projection matrix ($\boldsymbol{W}_1$ in equation 5). We initialize the bias of all linear layers to zero. Together, these choices bound the activation variance at initialization, ensuring that the forward pass can travel through an arbitrary number of initial random layers without magnitude explosion or decay. In early experiments, We find that scaling the weight matrices according to the depth of the corresponding encoder or decoder is slightly better than using the total model depth. We defer a formal analysis to the Appendix D.

Stabilize training with Pre-Norm formulation. Post-Norm architectures are known to become increasingly difficult to optimize at large model scales (Liu et al., 2023), making optimization more sensitive to hyperparameter choices. We observe a similar phenomenon in our experiments with the Post-Norm adaGLU defined in equation 6. In particular, when training a 30-layer and 1152-dimension ViT-XL model, both the loss and FID exhibit notable spikes (Appendix Figure 6). We therefore introduce a Pre-Norm variant in which the normalization layer is applied to the backbone feature before input to adaGLU and replace the conventional fusion operator with simple addition. Specifically, we define

$$\boldsymbol{z} = \text{adaGLU}(\boldsymbol{h}, \text{Norm}(\boldsymbol{z})) + \boldsymbol{z}. \tag{7}$$



Figure 4: Quantitative results of class-to-image generation on ImageNet-512. We display more results in the Appendix.

We refer to this formulation as Pre-Norm adaGLU. We compare it with the Post-Norm adaGLU defined in equation 6 in Figure 3 across various model sizes and training batch sizes. The Pre-Norm formulation avoids the training instability (or FID spike) observed for the Post-Norm formulation at large model sizes (XL size). Besides, its scaling behavior also shows better consistency across different training batch sizes.

4.4 Discussion

Our final architecture combines the two modifications identified above: asymmetric long skip connections and adaptive feature fusion with adaGLU. The resulting design retains a standard transformer backbone while using the long skip primarily as a pathway for low-level information. Figures 2b and 2c show the overall architecture and the Pre-Norm adaGLU, respectively. We also compare our final solution to prior U-Net variants in Table 1.

We further examine two architectural choices. First, the input feature $\boldsymbol{h}$ used by adaGLU is taken from the tokens immediately after patch embedding and before the first ViT layer. One alternative is to introduce a separate patch-embedding layer for each adaGLU block. Our experiments (Appendix Table 12) on these two settings show only marginal differences in FID on ImageNet-256, so we use the shared input embedding for simplicity. Second, we use an equal number of encoder and decoder layers. For example, a 12-layer model contains 6 encoder layers and 6 decoder layers. Although increasing the number of decoder layers monotonically improves performance in our ablations (Appendix Table 12), we retain a uniform layer allocation throughout the experiments and find that it's sufficient for strong generation performance.

5 Experiment

Settings. We use the same set of hyper-parameters across all experiments, without scale- or resolution-specific tuning. The resulting performance is therefore obtained without optimizing the training configuration for each setting. For example, we use 2e-4 learning rate, Adam optimizer with betas equal (0.9, 0.95), and 0.9999 ema decay, 1024 batch size. For text-to-image generation, we reduce the batch size to 512 while keeping all other settings unchanged. Besides, we uses patch size that linearly scales with input image resolution (e.g., 16 and 32 for resolutions 256 and 512, respectively), maintaining consistent computation cost across various resolutions. We defer more implementation details to the Appendix.

Table 5: Results on ImageNet-256. We categorize the diffusion models trained with additional self-supervised learning (SSL) stages or using external semantic encoders as *w SSL*. Following DiT, we report GFLOPs of our model as: $M \cdot N$, where M denotes model parameters and N is the token sequence length. For latent diffusions, we display the parameters of the VAE decoder and the diffusion model sequentially.

method	FID↓	IS↑	pre.	rec.	epoch	#params	GFLOPs
w SSL							
USP DiT-XL/2 (Chu et al., 2025)	2.33	267.0	-	-	240	84 + 675M	312 + 119
PixelGen (Ma et al., 2026b)	1.83	293.6	0.79	0.63	160	676M	
PixelU-H/16 (Guo et al., 2026)	1.63	305.9	0.79	0.64	600	1.2B	137
PixelDiT-XL (Yu et al., 2026)	1.61	292.7	0.78	0.64	320	797M	311
EPG-G/16 (Lei et al., 2026)	1.58	298.4	0.80	0.63	1600	1391M	321
HyperDiT-H (He et al., 2026)	1.56	306.5	0.80	0.64	600	952M	-
REPA (Yu et al., 2025)	1.42	305.7	0.80	0.65	800	84 + 675M	312 + 119
RAE (Zheng et al., 2025)	**1.13**	262.9	-	-	800	415 + 839M	107 + 146
train from scratch							
LDM (Rombach et al., 2022)	3.60	247.7	0.87	0.48	167	55 + 400M	336 + 104
U-ViT-H/2 (Bao et al., 2023a)	2.29	263.9	0.82	0.57	400	84 + 501M	312 + 133
MaskDiT (Zheng et al., 2024)	2.28	276.6	0.80	0.61	1600	84 + 675M	312 + 119
DiT-3B-G (Gao et al., 2024)	2.10	304.4	0.82	0.60	340	3B	-
DAR (Xu et al., 2026)	2.05	270.1	0.82	0.60	400	740M	-
JiT-G/16 (Li & He, 2025)	1.82	292.6	0.79	0.62	600	2B	383
SiD2 (Hoogeboom et al., 2025)	1.72	-	-	-	-	-	137
PiT-B/16	3.60	237.9	0.83	0.50	800	123M	32
PiT-L/16	2.24	259.8	0.81	0.59	800	339M	87
PiT-XL/16	1.78	278.7	0.79	0.62	800	640M	165
PiT-G/16	**1.62**	293.1	0.79	0.64	800	1.6B	403

Table 6: Results on ImageNet-512. The settings are the same as Table 5.

method	FID↓	IS↑	prec.	rec.	epoch	#params	Gflops
EPG-L/32 (Lei et al., 2026)	2.35	295.4	0.82	0.57	800	540M	113
DeCo-XL/16 (Ma et al., 2026a)	2.22	290.0	0.80	0.60	340	682M	-
U-ViT-H/2 (Bao et al., 2023a)	4.05	263.8	0.84	0.48	400	84 + 501M	1260 + 133
SiT-XL/2 (Ma et al., 2024)	2.62	252.2	0.84	0.57	600	84 + 675M	1260 + 525
SiD2 (Hoogeboom et al., 2025)	2.19	-	-	-	-	-	137
PixelU-H/32	1.92	322.1	0.80	0.58	600	1.2B	-
PixelDiT-XL	1.81	278.6	0.78	0.67	850	797M	1352
EDM2 (Karras et al., 2024)	1.81	-	-	-	-	1.5B	552
JiT-G/32 (Li & He, 2025)	1.78	306.8	-	-	600	2B	384
PiT-B/32	4.16	256.8	0.84	0.46	800	126M	33
PiT-L/32	2.64	267.7	0.83	0.54	800	343M	88
PiT-XL/32	2.10	290.9	0.81	0.60	800	646M	167
PiT-G/32	**1.70**	313.2	0.80	0.62	800	1.6B	405

Training & Evaluation. For class-to-image generation, we train models on ImageNet-1K for 1000K steps (800 epochs). We report the FID (Heusel et al., 2017), IS (Salimans et al., 2016), Precision and Recall (Kynkäänniemi et al., 2019) of models with 50K samples. We use reference statistics from ADM (Dhariwal & Nichol, 2021). For text-to-image generation, we use FLAN-T5-Large encoder (Chung et al., 2022) to encode prompt conditions. Following the settings of MiniT2I (Wang et al., 2026), we first train the model on CC12M (Changpinyo et al., 2021) for 500K steps and then fine-tune it for 40K steps on 120K text-image pairs from three datasets, including Open-Dalle3[2], ShareGPT (Chen et al., 2025b), and BLIP3o-60k (Chen et al., 2025a). We evaluate the model performance on GenEval and DPG-Bench and generate four images per test prompt without prompt rewriting. We generate samples utilizing interval classifier-free guidance (Ho & Salimans, 2022; Kynkäänniemi et al., 2024) and the Heun ODE sampler (Karras et al., 2022) with 50 sampling steps.

[2]https://huggingface.co/datasets/OpenDatasets/dalle-3-dataset

Table 7: Benchmark text-to-image generation performance on GenEval and DPG-Bench. We display parameters of the text encoder and diffusion model sequentially.

method	GenEval	DPG-Bench	#params
PixelDiT-T2I (Yu et al., 2026)	0.78	83.7	2.6+1.3B
PixelGen-XXL/16 (Ma et al., 2026b)	0.79	-	1.7+1.1B
DeCo-XXL/16 (Ma et al., 2026a)	**0.86**	81.4	1.7+1.1B
PiT-L/32	0.82	**84.3**	341+344M



Figure 5: Qualitative results of text-to-image generation. More results are deferred to the Appendix.

5.1 Class-to-image generation

We display our results on ImageNet-256 and ImageNet-512 in Table 5 and Table 6 respectively. When compared with prior methods, PiT reaches competitive performance. On ImageNet-256, it surpasses several recent methods trained with an additional training stage or external models, including USP and PixelGen, and is on par with PixelDiT and EPG.

Comparing with models with U-Net-style skip connection, such as U-ViT, PixelU, DAR, and EDM2, our models yield outstanding results. Among them, EDM2 shows notable performance with its magnitude-preserving network designs. In comparison, our results reveal that such additional structural complexity is unnecessary for building a strong pixel denoiser.

Scalability. PiT maintains strong performance as the model size increases from the small-scale baseline to 1.6B parameters. This scaling behavior becomes stronger and more pronounced on ImageNet-512, due to the increased complexity of the generative modeling task.

5.2 Text-to-image Generation

We show that our model natively supports text and image input without relying on cross attention or the two-stream architecture as in MMDiT (Esser et al., 2024). Using a large patch size of 32, the input tokens to our model comprise three parts, including time condition (1), text embeddings (256), and image tokens (256), resulting in a total of 513 tokens. In Table 7, our middle-sized PiT-L/32 model reaches generation performance on par with prior works.

6 Discussions & Limitations

Discussions. U-Net was once a standard architecture for diffusion model. Our study revisits its design from a different perspective. We find that the useful role of the long skip does not require its symmetric encoder-decoder structure. The skip can instead be viewed primarily as a pathway for low-level information, and its fusion with the backbone can be made adaptive to the denoising state. This leads to a simpler formulation in which the main backbone remains a standard transformer and the long skip is implemented as an asymmetric, noise-adaptive pathway.

Limitations. Our experiments are primarily conducted on ImageNet image generation. Although PiT scales to 1.6B parameters in our experiments, its behavior at substantially larger scales remains unclear. We also did not evaluate PiT on video generation or other high-dimensional sequential data, where the benefit of the simplified routing design may slightly differ. Finally, while the token-based formulation suggests that PiT can be incorporated into systems beyond image generation, we have not evaluated adaGLU as a standalone component in existing large language model architectures.

7 AI TOOL USAGE DISCLOSURE

In this work, we used generative AI tools to assist with polishing and improving the clarity of our writing. We carefully reviewed all AI-assisted content to ensure that they accurately reflect our original ideas and intended meaning. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

8 ETHICS STATEMENT

This work does not involve human subjects or the collection of private or sensitive data. We use existing public datasets and benchmarks for image and text-to-image generation. We acknowledge that these datasets may contain inherited biases or undesirable content, which may be reflected in the resulting models and samples.

9 REPRODUCIBILITY STATEMENT

To ensure reproducibility of our work, we provide detailed model architectures, training and sampling hyper-parameters, evaluation protocols, and ablation settings in Section 5 of the main paper and Section A in the Appendix. The experimental setup is kept consistent across model scales and resolutions.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models, 2023a. URL https://arxiv.org/abs/2209.12152.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, pp. 1692–1717. PMLR, 2023b.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts, 2021. URL https://arxiv.org/abs/2102.08981.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset, 2025a. URL https://arxiv.org/abs/2505.09568.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation, 2025b. URL https://arxiv.org/abs/2506.18095.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow, 2025c. URL https://arxiv.org/abs/2504.07963.

Xiangxiang Chu, Renda Li, and Yong Wang. Usp: Unified self-supervised pretraining for image generation and understanding. *arXiv preprint arXiv:2503.06132*, 2025.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks, 2017. URL https://arxiv.org/abs/1612.08083.

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

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

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

Peng Gao, Le Zhuo, Dongyang Liu, Ruoyi Du, Xu Luo, Longtian Qiu, Yuhang Zhang, Chen Lin, Rongjie Huang, Shijie Geng, Renrui Zhang, Junlin Xi, Wenqi Shao, Zhengkai Jiang, Tianshuo Yang, Weicai Ye, He Tong, Jingwen He, Yu Qiao, and Hongsheng Li. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers, 2024. URL https://arxiv.org/abs/2405.05945.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023. URL https://arxiv.org/abs/2310.11513.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. Pixelu: A u-shaped transformer for efficient end-to-end pixel diffusion, 2026. URL https://arxiv.org/abs/2606.27760.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015. URL https://arxiv.org/abs/1502.01852.

Yu He, Lichen Ma, Zipeng Guo, Xinyuan Shan, Jingling Fu, Dong Chen, Junshi Huang, and Yan Li. Hyperdit: Hyper-connected transformers for high-fidelity pixel-space diffusion, 2026. URL https://arxiv.org/abs/2605.15741.

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. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images, 2023. URL https://arxiv.org/abs/2301.11093.

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, 2025. URL https://arxiv.org/abs/2410.19324.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024. URL https://arxiv.org/abs/2403.05135.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation, 2023. URL https://arxiv.org/abs/2212.11972.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proc. CVPR*, 2024.

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.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. URL https://arxiv.org/abs/1904.06991.

Jiachen Lei, Peng Cheng, Zhongjie Ba, and Kui Ren. Masked diffusion models are fast learners. *arXiv preprint arXiv:2306.11363*, 2023.

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. URL https://arxiv.org/abs/2510.12586.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2025. URL https://arxiv.org/abs/2511.13720.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers, 2023. URL https://arxiv.org/abs/2004.08249.

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, 2024. URL https://arxiv.org/abs/2401.08740.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation, 2026a. URL https://arxiv.org/abs/2511.19365.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Improving pixel diffusion with perceptual supervision, 2026b. URL https://arxiv.org/abs/2602.02493.

Pierre Marion, Adeline Fermanian, Gérard Biau, and Jean-Philippe Vert. Scaling resnets in the large-depth regime, 2025. URL https://arxiv.org/abs/2206.06929.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. URL https://arxiv.org/abs/1606.03498.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, Yutian Chen, Junjie Yan, Ming Wei, Y. Zhang, Fanqing Meng, Chao Hong, Xiaotong Xie, Shaowei Liu, Enzhe Lu, Yunpeng Tai, Yanru Chen, Xin Men, Haiqing Guo, Y. Charles, Haoyu Lu, Lin Sui, Jinguo Zhu, Zaida Zhou, Weiran He, Weixiao Huang, Xinran Xu, Yuzhi Wang, Guokun Lai, Yulun Du, Yuxin Wu, Zhilin Yang, and Xinyu Zhou. Attention residuals, 2026. URL https://arxiv.org/abs/2603.15031.

Yuchuan Tian, Zhijun Tu, Hanting Chen, Jie Hu, Chao Xu, and Yunhe Wang. U-dits: Downsample tokens in u-shaped diffusion transformers, 2024. URL https://arxiv.org/abs/2405.02730.

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.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion, 2025a. URL https://arxiv.org/abs/2507.23268.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer, 2025b. URL https://arxiv.org/abs/2504.05741.

Xianbang Wang, Hanhong Zhao, Yiyang Lu, Kangyang Zhou, Linrui Ma, and Kaiming He. Minit2i: A minimalist baseline for text-to-image generation, 2026. URL https://peppaking8.github.io/#/post/minit2i.

Chao Xu, Maohua Li, Qirui Li, Yixuan Xu, Yanke Zhou, Yunhe Li, Cuifeng Shen, Hanlin Tang, Kan Liu, Tao Lan, Lin Qu, and Shao-Qun Zhang. Rethinking cross-layer information routing in diffusion transformers, 2026. URL https://arxiv.org/abs/2605.20708.

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. URL https://arxiv.org/abs/2410.06940.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation, 2026. URL https://arxiv.org/abs/2511.20645.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization, 2019. URL https://arxiv.org/abs/1901.09321.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders, 2025. URL https://arxiv.org/abs/2510.11690.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. In *Transactions on Machine Learning Research (TMLR)*, 2024.

A EXPERIMENT SETTINGS

Training. We list the training and sampling hyper-parameters used in our experiments in Table 9. As we adopt the variance-exploding diffusion model, to mitigate the model input variance, we scale the model input $\boldsymbol{x}(t)$ with a coefficient $c_{in} = 1/\sqrt{t^2 + \sigma_{data}^2}$ following Karras et al. (2022).

Evaluation. In class-conditional image generation experiments (Table 5 and 6), we generate 50 samples per class, aligned with the settings in JiT (Li & He, 2025).

Architecture. We list our architectural settings in Table 8.

Table 8: The architecture settings of PiTs.

model	layers	dim	mlp dim	adaGLU dim	heads
PiT-B	12	768	2048	2048	12
PiT-L	20	1152	3072	3072	16
PiT-XL	30	1152	3072	3072	16
PiT-G	30	1536	6144	6144	16

Table 9: Training & sampling hyper-parameters in main experiments including class-to-image and text-to-image generation.

hyper-parameter	value
training	
data augmentation	horizontal flip
mixed precision	bf16
batch size	1024
learning rate	2e-4
optimizer	Adam, β_1, β_2 = 0.9, 0.95
patch size	image_size/16
diffusion setting	$\sigma_{min}, \sigma_{max}, \sigma_{data}$ = 0.002, 80, 0.5
noise scale	image_size/64
weighting function	$\frac{1}{t_n - t_{n-1}}, t_n \in [\sigma_{min}, \sigma_{max}], N = 1280$
time sampler	$\log(t) \sim \mathcal{N}(\mu, \delta^2), \mu = -1.2, \delta = 1.6$
ema decay	0.9999
cfg drop	0.1
sampling	
sampler	Heun
sampling step	50
cfg interval	$[0.19, 1.61]$ (if used)

B APPLE-TO-APPLE COMPARISON WITH JIT

Following the evaluation settings of JiT, we report FID scores of PiT models measured with the statistics of the training data, in Table 10. Note that the FID scores are slightly different from the numbers reported in our main experiments, where we use the statistics provided by ADM.

On both datasets, utilizing the same training recipe, our PiT models consistently benefit from longer training time and larger model size. On ImageNet-256, our model shows stronger scalability without the risk of overfitting. On ImageNet-512, with increased complexity, the number of model parameters becomes more essential for final generation performance. However, our 1.6B PiT-G/32 model surpasses JiT-G of 2B parameters. On both datasets, our models require neither additional in-context embeddings nor extensive hyper-parameter tuning, exhibiting clean scaling behavior. In Table 11, we evaluate a JiT-B model trained without in-context embeddings. The results reveal JiT's heavy reliance on in-context embeddings.

Table 10: Apple-to-apple comparison with JiT across various model sizes. We report model parameters from ImageNet-512 experiments.

model	#params	IN256	IN512	epoch
JiT-L	462M	2.36	2.53	600
PiT-L	343M	2.40	2.92	600
		2.23	2.68	800
JiT-H	956M	1.86	1.94	600
PiT-XL	674M	1.89	2.31	600
		1.76	2.10	800
JiT-G	2B	1.82	1.78	600
PiT-G	1.6B	1.61	1.83	600
		1.56	1.72	800

Table 11: Performance of JiT-B model without in-context embeddings.

model	#params	FID
JiT	131M	38.1
JiT w/o in-context emb.	131M	47.1
PiT-B	123M	36.2

Table 12: We ablate several design choices, including decoder depth, block sequence in decoder layer, and input embedding used in adaGLU.

(a) encoder-decoder depth.

enc	dec	FID-10K
6	24	29.1
15	15	30.8
24	8	33.2

(b) block sequence.

	FID-10K
adaGLU-attn-mlp	30.8
attn-adaGLU-mlp	30.6

(c) input embeddings.

	FID-10K
shared	15.26
independent	15.78

C Additional Experiment Results

By default, we run experiments with PiT-XL/16 on ImageNet-256. We report FID measured with the statistics of ImageNet-256 training data. The default settings are highlighted in blue color. We mainly present four additional experiments, including three ablation studies on model architecture and an additional experiment with representation alignment loss in Yu et al. (2025).

In the ablation studies, we show that increasing decoder depth monotonically improves performance (Table 12). We also show that both rearranging the position of the adaGLU block and utilizing input embeddings extracted from independent patch embedding layers produce similar results as the default settings (Table 12 and 12). When trained with representation alignment, the FID in Table 13 noticeably drops from 1.78 to 1.64.

D Model Initialization

Let $\boldsymbol{x}_l$ denote the input tensor to the l-th transformer layer, which consists of an attention block $f_1(\cdot)$ and an MLP block $f_2(\cdot)$. The forward pass of a Pre-LN block is defined as:

$$\boldsymbol{h}_l = \boldsymbol{x}_l + f_1(\text{LN}(\boldsymbol{x}_l))$$
$$\boldsymbol{x}_{l+1} = \boldsymbol{h}_l + f_2(\text{LN}(\boldsymbol{h}_l))$$

We evaluate the variance $\boldsymbol{V}_l = \text{Var}(\boldsymbol{x}_l)$ sequentially through the network, assuming the residual branches $f_1(\text{LN}(\boldsymbol{x}_l))$, $f_2(\text{LN}(\boldsymbol{h}_l))$ are uncorrelated with the main stream activations $\boldsymbol{x}_l$ at initializa-

Table 13: Comparing performance of PiT-XL/16 models, one is trained from scratch trained and the other is trained with representation alignment loss. We use default training settings of REPA and align the features of the 10-th layer with the output of DINOv3-B model (Siméoni et al., 2025).

	FID	sFID	IS	pre.	rec.	epoch
from scratch	1.78	4.35	278.7	0.79	0.62	800
w REPA	1.64	4.30	284.8	0.80	0.62	800

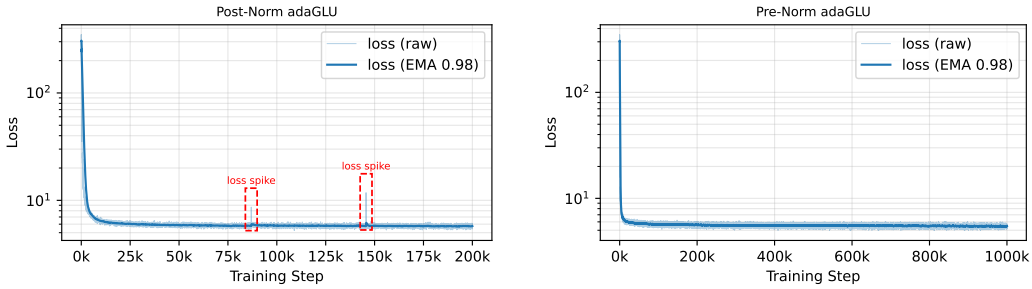


Figure 6: Loss curve of Pre-Norm (right) and Post-Norm (left) adaGLU. We observe a notable loss spike when training with Post-Norm adaGLU. In contrast, the training of Pre-Norm adaGLU is stable.

tion. We assume the model input has a zero mean. The derivation is not a rigorous mathematical analysis but an intuitive explanation of the underlying rationale of our initialization strategy.

Variance after the attention block. The LayerNorm operation standardizes its input such that $\text{Var}(\text{LN}(\boldsymbol{x}_l)) = 1$. Under He initialization (fan-in mode), the weight matrices are scaled to preserve the variance of normalized inputs through the linear layers. Let the variance of the unscaled attention output be a constant $\text{Var}(f_1(\text{LN}(\boldsymbol{x}_l))) = C_A \approx 1$. By scaling the output matrix by $\frac{1}{\sqrt{2L}}$, the variance of the attention branch is

$$\text{Var}\left(\frac{1}{\sqrt{2L}} f_1(\text{LN}(x_l))\right) = \left(\frac{1}{\sqrt{2L}}\right)^2 \text{Var}(f_1(\text{LN}(x_l))) = \frac{C_A}{2L}$$

The variance of the intermediate representation h_l is the sum of the main stream variance and the branch variance:

$$\begin{aligned}\text{Var}(h_l) &= \text{Var}(x_l) + \frac{C_A}{2L} \\ &= V_l + \frac{C_A}{2L}\end{aligned}$$

Variance after the MLP Block. Similarly, the intermediate tensor h_l is normalized before entering the MLP, yielding $\text{Var}(\text{LN}(h_l)) = 1$. He initialization preserves this unit variance through the hidden layers, producing an unscaled output variance $C_M \approx 1$. Scaling the MLP output matrix by $\frac{1}{\sqrt{2L}}$ results in a branch variance of $\frac{C_M}{2L}$. Adding this to the intermediate main stream gives the variance at the output of the l-th layer:

$$\begin{aligned}V_{l+1} &= \text{Var}(h_l) + \frac{C_M}{2L} \\ &= V_l + \frac{C_A}{2L} + \frac{C_M}{2L} \\ &= V_l + \frac{C_A + C_M}{2L}\end{aligned}$$

Variance after L layers. To find the final variance V_L after passing through all L layers, we unroll the recurrence relation from the initial embedding x_0:

$$\begin{aligned}V_L &= V_0 + \sum_{l=1}^{L} \frac{C_A + C_M}{2L} \\ &= V_0 + L\left(\frac{C_A + C_M}{2L}\right) \\ &= V_0 + \frac{C_A + C_M}{2}\end{aligned}$$

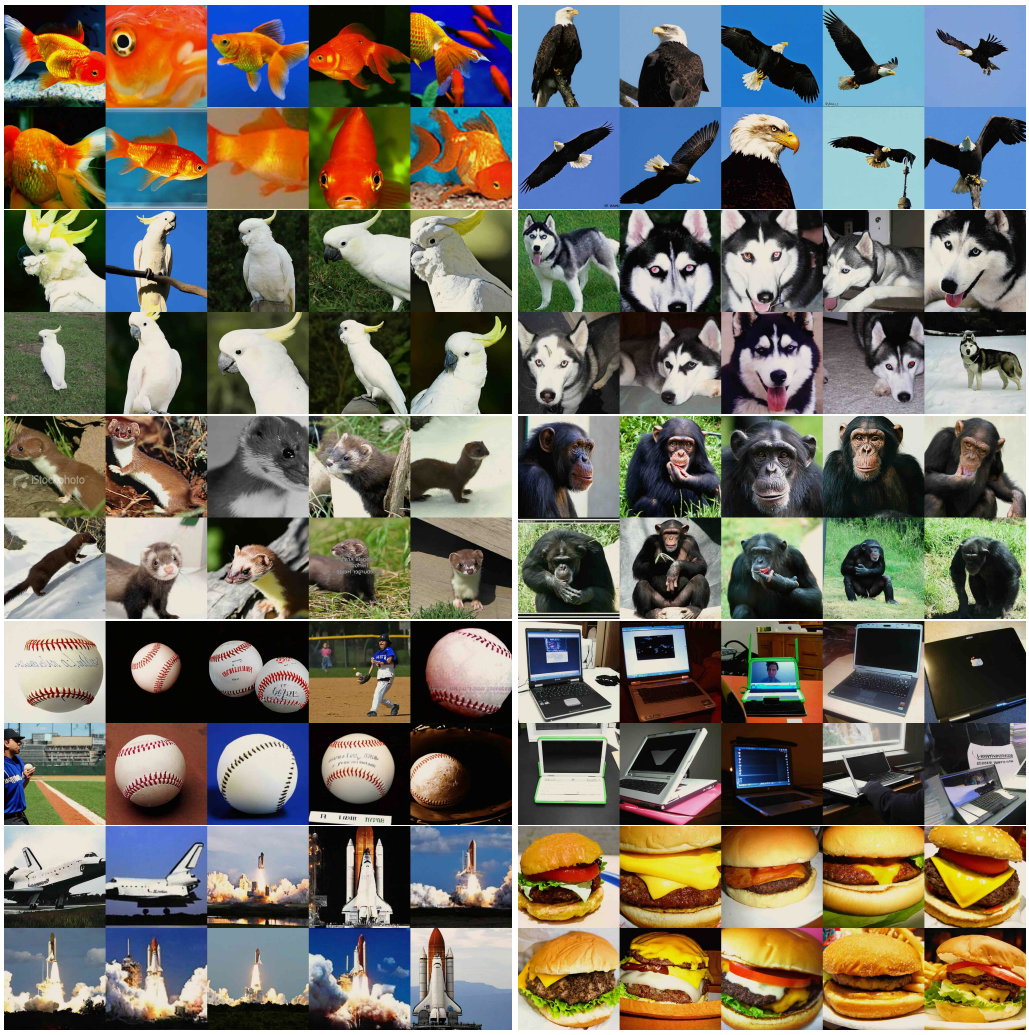


Figure 7: Uncurated samples. Class-to-image generation. Class: 1, 22, 89, 250, 356, 367, 429, 620, 812, 933. Setting: Heun 50 steps, CFG 4.0.

Without the $\frac{1}{\sqrt{2L}}$ factor, the branch variances C_A and C_M would be added directly to the main stream at each layer, resulting in $V_L = V_0 + L(C_A + C_M)$. The variance would scale linearly with depth $O(L)$. By scaling the output matrices by $\frac{1}{\sqrt{2L}}$, the depth factor L is analytically canceled out in the sum. The final variance V_L becomes strictly bounded by a constant $\mathcal{O}(1)$, regardless of how deep the network is scaled. When adding an adaGLU block to decoder layers, we zero-initialize its output matrix and use standard He initialization for the other linear layers. This ensures our prior analysis remains valid.

E Additional Qualitative Results

We display more qualitative results of our PiT-G/32 model trained on ImageNet-512 in Figure 7 and qualitative results of the text-to-image generation model PiT-L/32 in 8. We compress each image grid to around 100KB to avoid excessive supplementary file size.



A quaint fairy cottage nestled in a lush garden, with delicate smoke wisps rising from its stone chimney. The cottage features a thatched roof and walls covered in climbing ivy. A curious squirrel peers out from a small, round window, framed by wooden shutters.



a tranquil lakeside setting where a herd of sauropods is seen gracefully traversing the water's edge. The gentle giants, with their long necks and tails, are silhouetted against the backdrop of a dense forest. The calm lake reflects the soft hues of the sky as the prehistoric creatures continue their age-old migration.



a close-up image of a sandy beach, where a bright red bucket lies on its side, surrounded by an array of colorful seashells scattered across the fine grains. there are no birds in sight, particularly no sandpipers, which often frequent such shores. the gentle waves in the background suggest the proximity to the water's edge.



a brightly colored storefront with large, bold letters spelling out 'AwesomePurchase' above the entrance. The shop's window displays are neatly arranged with an array of products, and a small, potted plant sits to the left of the door. The facade of the building is a clean, modern white, contrasting with the vibrant signage.



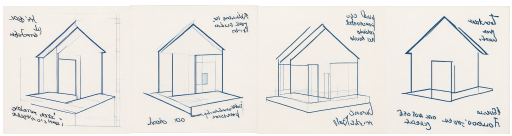
a concrete sidewalk running alongside a weathered wooden post, which has a bright blue '5' prominently painted on its flat top surface. The post is planted firmly in the ground, with patches of green grass sprouting around its base. To the side of the sidewalk, there's a neatly trimmed hedge that stretches out of view.



A vibrant mural on a red brick wall features the inspirational phrase "BE EXCELLENT TO EACH OTHER" in bold, black lettering. Next to the text, there's a whimsical graffiti depiction of a green alien donning a sleek black tuxedo, complete with a bow tie. In the foreground, a bright yellow fire hydrant stands out on the gray concrete sidewalk, adding a pop of color to the urban scene.



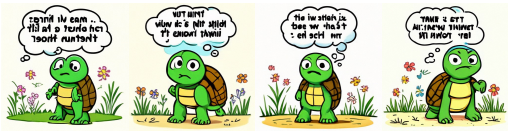
In the gentle light of the early morning, three red stuffed animals—two teddy bears and a plush fox—are propped against a soft pastel-colored wall within a peaceful nursery room. The wall itself is painted in a gradient of pastel hues, creating a calming backdrop for the vibrant toys. The toys' plush fabric appears soft to the touch, and they sit closely together as if in a huddled group, providing a cheerful contrast to the subtle tones of the room. Nearby, a white wooden crib with delicate bedding completes the serene setting, signifying the presence of a young child's space.



an architectural blueprint displaying a simple house design, with clean lines indicating a large triangular roof atop square walls, and a rectangular base representing the floor. The drawing is executed with precise, thin blue lines on a white background, giving it a technical and minimalist appearance. Alongside the schematics, there's handwritten text that reads 'this house is built on the principles of abstraction', suggesting an artistic or philosophical approach to the building's design.



a tranquil cityscape with high-rise buildings silhouetted against the evening sky. In the foreground, a large, fluffy, solitary cloud hovers subtly, its edges tinged with a golden hue from the setting sun. Below the cloud, in elegant, rounded cursive letters, the words 'contemplate the clouds' invite onlookers to pause and reflect amidst the urban environment.



an animated image depicting a green turtle with a perplexed expression on its face. The turtle is standing upright on two legs and has a large, transparent thought bubble above its head, filled with the paradoxical question, 'what if there was no such thing as a thought bubble?' surrounding the turtle, there are simplistic drawings of grass and flowers, emphasizing the cartoonish nature of the image.

Figure 8: Uncurated samples. Text-to-image generation.