EXPLORING THE DESIGN FACTORS OF PIXEL-SPACE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we introduce EiT by exploring the key design factors of pixel-space diffusion from three perspectives: model architecture, representation, and frequency domain. Inspired by Denoising Autoencoders (DAE) which performs denoising through a bottleneck, we construct a U-shaped pixel diffusion transformer that performs explicit compression along the channel dimension. In terms of representation, we find that both pixel and latent diffusion transformers exhibit a three-stage representation dynamics across layers and timesteps: denoising/local feature expansion, basis lift-up, and posterior projection. Moreover, earlier denoising and a longer basis lift-up stage are associated with better generation, which EiT naturally achieves through its channel bottleneck and increased depth. From the frequency perspective, we establish the equivalence between pixel-space and frequency-space diffusion, and show that the latter enables flexible frequency-band loss reweighting, yielding performance gains at no extra cost. On ImageNet 256×256 and 512×512, EiT achieves significantly better generation quality than JiT (Just image Transformers). Our code will be made publicly available.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2020) and flow matching (Lipman et al., 2022; Liu et al., 2022) have rapidly advanced visual generation. For high-resolution synthesis, the dominant paradigm has long been latent diffusion (Rombach et al., 2022; Peebles & Xie, 2023; Ma et al., 2024): a variational autoencoder (VAE) (Kingma & Welling, 2013) first compresses images into a compact latent space, where a diffusion model is then trained. However, this two-stage pipeline is expensive, since the VAE requires careful design and costly training, and sampling requires running both the diffusion model and the decoder. Consequently, end-to-end pixel diffusion has attracted growing attention and demonstrated strong competitiveness (Hoogeboom et al., 2023; Crowson et al., 2024; Ma et al., 2026; Chen et al., 2026; Yu et al., 2026). A representative example is JiT (Li & He, 2026), which shows that a plain diffusion transformer with large patches performs remarkably well under x-prediction. Nevertheless, pixel diffusion still lags behind state-of-the-art latent diffusion (Singh et al., 2026).

In this paper, we explore the design factors of pixel diffusion from three perspectives: model architecture, representation, and frequency domain. First, an x-prediction diffusion network (at least in the low-noise regime) resembles a Denoising Autoencoder (DAE) which maps noisy inputs back onto the data manifold (Vincent et al., 2008). Narrowing the hidden dimension forces DAE to keep only the dominant signal directions while discarding noise (Vincent et al., 2010; Baldi & Hornik, 1989). Motivated by DAE, we introduce compression into the pixel diffusion transformer, unifying representation compression and denoising in a single end-to-end network, EiT. Unlike prior U-shaped designs that merge tokens spatially (Crowson et al., 2024; Guo et al., 2026), EiT compresses the channel dimension while keeping the token grid intact. On ImageNet, EiT significantly outperforms the plain diffusion transformer JiT, typically with less computation. Second, by analyzing the effective rank and noise fraction, we reveal a three-stage representation dynamics in both pixel and latent diffusion transformers: denoising/local feature expansion, basis lift-up, and posterior projection. We interpret basis lift-up as the backbone expanding posterior-relevant information into a rich set of directions, from which the final layer can predict the target by linear projection. We further observe that deeper networks and earlier denoising generally lead to better generation. EiT naturally falls into this regime: its channel bottleneck suppresses noise more effectively, and

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.

the saved computation affords greater depth for basis lift-up. Finally, we prove that pixel-space and frequency-space diffusion are equivalent under orthonormal transforms such as the DCT and DWT. Frequency-space diffusion nevertheless offers additional flexibility: reweighting the loss across frequency bands consistently improves generation quality without extra parameters or FLOPs. To summarize, our contributions are:

- We propose EiT, a pixel diffusion transformer with channel compression, which outperforms JiT and PixelU under comparable budgets and also benefits latent diffusion.
- We uncover a three-stage representation dynamics shared by pixel and latent diffusion, where earlier denoising and longer basis lift-up are associated with better generation.
- We prove the equivalence of pixel-space and frequency-space diffusion, and show that frequency-band loss reweighting further improves generation at no extra cost.

2 Related Work

Pixel diffusion. Early pixel diffusion models make high-resolution training feasible through shifted noise schedules, sigmoid loss weighting, and hierarchical UNet or hourglass Transformer backbones (Hoogeboom et al., 2023; Crowson et al., 2024; Hoogeboom et al., 2025). A promising work is JiT (Li & He, 2026): under the manifold assumption, clean images lie on a low-dimensional manifold, so predicting the clean image (x-prediction) lets a plain large-patch Transformer model high-dimensional pixels even with limited capacity, where ϵ- and v-prediction fail catastrophically. Concurrently, DeCo, DiP, and PixelDiT (Ma et al., 2026; Chen et al., 2026; Yu et al., 2026) share a frequency-decoupled design: a DiT on large patches models low-frequency semantics, while a lightweight pixel-level module restores high-frequency details. Other VAE-free methods exploit the frequency structure of images more explicitly, including diffusion in the DCT space (Ning et al., 2025), coarse-to-fine resolution cascades (Chen et al., 2025), and band-specific noise schedules in the wavelet domain (Pavasovic et al., 2026). Most related to our architecture are HDiT (Crowson et al., 2024) and PixelU (Guo et al., 2026), which compress representations along the spatial dimension by merging neighboring tokens. In contrast, we explore compression along the channel dimension while keeping the token grid intact.

Diffusion representation. Research connecting diffusion models and representations follows two directions. The first treats diffusion models as representation learners. Diffusion Autoencoders pair a semantic encoder with a diffusion decoder to obtain decodable latent codes (Preechakul et al., 2022), and DiffMAE casts masked image modeling as conditional diffusion for visual pretraining (Wei et al., 2023). Pretrained diffusion features also transfer to understanding tasks: DIFT extracts intermediate activations for zero-shot correspondence (Tang et al., 2023), and RepFusion distills time-dependent features into recognition networks (Yang & Wang, 2023). The second direction uses representations to improve generation. REPA aligns noisy hidden states with clean-image features from pretrained encoders (Yu et al., 2024), while SRA aligns its shallow, high-noise features with its own deeper, low-noise ones (Jiang et al., 2026). RAE instead performs diffusion directly in the latent space of frozen representation encoders (Zheng et al., 2026), and RiT shows that a vanilla diffusion transformer with x-prediction suffices in this space (Zhang et al., 2026). However, these works either evaluate diffusion features by downstream utility or exploit representations as supervision and latent spaces. In this paper, we dive deep into the representation dynamics in diffusion transformers and reveal how they relate to the posterior target learning.

3 Analysis and Method

3.1 Background: Diffusion and Flow

Diffusion models and Flow matching models can be formulated from the perspective of ODEs (Ordinary Differential Equations) (Song et al., 2020; Lipman et al., 2022). For simplicity, we consider the flow matching formulation for the discussion in this paper. Given the data sample $\boldsymbol{x} \sim p_{data}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, a noisy sample is constructed by interpolation $\boldsymbol{x}_t = (1-t)\boldsymbol{x} + t\boldsymbol{\epsilon}$ under the linear schedule (Liu et al., 2022) in the time interval $t \in [0, 1]$. The conditional velocity is defined as $\boldsymbol{v} = \frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{\epsilon} - \boldsymbol{x}$ and is learned by a model $\boldsymbol{v}_\theta$. The flow matching objective is

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t} \left\| \boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v} \right\|_2^2 . \tag{1}$$

Given the learned flow velocity $\boldsymbol{v}_\theta$, the sampling is obtained by solving the ODE $\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ starting from $\boldsymbol{x}_1 \sim \mathcal{N}(0, \boldsymbol{I})$ and ending at $t = 0$. Following Li & He (2026), we adopt the x-prediction network to obtain the velocity function $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$.

3.2 U-SHAPED ARCHITECTURE AND CHANNEL COMPRESSION

Latent diffusion decomposes image generation into two stages: a VAE first maps an image to a compact latent space, and a diffusion model then learns the score or velocity in this compressed space (Rombach et al., 2022). In contrast, a diffusion model must learn its velocity field directly in the high-dimensional pixel space, and a standard isotropic Transformer processes representations of the same size throughout the network. Inspired by DAE that performs denoising through a bottleneck, we incorporate compression into the pixel diffusion Transformer. The resulting EiT model performs diffusion and representation compression jointly in one end-to-end network. Existing U-shape pixel diffusion Transformers compress representations along the spatial dimension. For example, HDiT repeatedly merges neighboring tokens (Crowson et al., 2024), while PixelU uses one spatial downsampling stage (Guo et al., 2026). Differ from these token merge methods, we explore another direction: compressing the **channel dimension** while retaining the token number unchanged (see Figure 1). In the experiments section, we will show that this channel-compression strategy yields a better generation quality than compressing the spatial dimension.

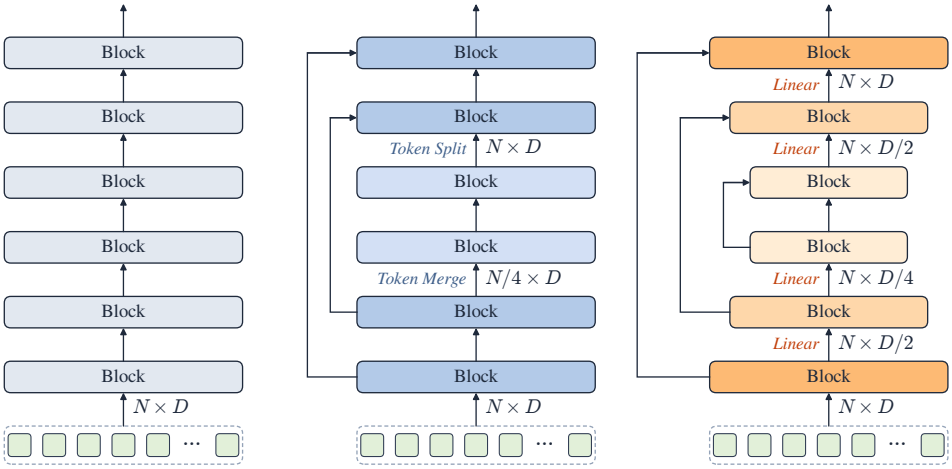

Figure 1: Comparison of pixel diffusion Transformer architectures. JiT keeps both the token count N and hidden dimension D fixed. PixelU constructs a spatial bottleneck by merging and splitting tokens. EiT preserves the full token grid and instead forms a channel bottleneck through linear projection; long skips connect mirrored encoder and decoder blocks.

To make our channel-compression a generic and simple method, we make the minimum changes to the original diffusion transformers (Ma et al., 2024; Li & He, 2026) by keeping the head dimension d fixed in self-attention and the number of tokens unchanged. **The channel compression is achieved by varying only the number of attention heads** h_ℓ, thus the hidden dimension is $D_\ell = h_\ell d$ at layer ℓ. Since the number of tokens and dimension of self-attention are unchanged, no effort is needed to modify the positional embedding (e.g., 2D RoPE (Su et al., 2024)). Moreover, the encoder and decoder are symmetric and blocks are uniformly distributed across stages. Therefore, the instantiation of EiT is easy and flexible. For instance, we can instantiate a 4-stage schedule 4-3-2-1, corresponding to encoder widths $(D, 3D/4, D/2, D/4)$, and the decoder reverses the selected schedule. Following U-ViT (Bao et al., 2023) and PixelU (Guo et al., 2026), mirrored encoder and decoder blocks are connected by long skips, fused through variance-preserving addition. The long skips usually carry the high-frequency information of input image from the shallow layers to the deep layers (Guo et al., 2026).

Channel compression is particularly efficient for image generation. It is known that much of the computation in Transformer (Vaswani et al., 2017) is spent in feed-forward network (FFN) whose cost scales as $\mathcal{O}(ND^2)$, rather than in attention with cost $\mathcal{O}(N^2D)$ in the small N regime. Reducing the stage width from D to αD therefore reduces the parameters and FLOPs of FFN to approximately α^2 of their original values. EiT can therefore use fewer parameters and FLOPs than JiT while obtaining better generation quality. Alternatively, the saved computation can be used to increase the depth under matched parameter and FLOP budgets with JiT. Furthermore, our analysis in Section 3.3 suggests two benefits of EiT architecture: the channel bottleneck suppresses noise better than JiT, while the increased depth provides a longer basis lift-up stage before the final posterior projection.

3.3 Diffusion Representation Dynamics

In the x-prediction network, the diffusion transformer is trained to directly predict the posterior mean $\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{x}_t]$ which does not contain noise at any timestep t. Therefore, the network should learn representations to perform denoising. However, is denoising the only behaviour we should expect from all layers and timesteps? To understand the internal representation dynamics in diffusion transformer, we apply the **Spatial Participation Ratio (similar to the entropy effective rank) to characterize the dimensionality of the spatial representation, and use Noise Fraction to measure the noise capacity in the representation** . Specifically, let $\boldsymbol{H}^{\ell,t}(\boldsymbol{x}_t, t) \in \mathbb{R}^{N\times D}$ denote the layer-ℓ activation whose rows are the N spatial tokens. We subtract the mean spatial token to obtain $\widetilde{\boldsymbol{H}}^{\ell,t}(\boldsymbol{x}_t, t)$ and let $\{\sigma_i\}$ be its singular values. Defining the normalized spectral energy as $p_i = \sigma_i^2 / \sum_j \sigma_j^2$, the Spatial Participation Ratio (SPR) is

$$R_{\mathrm{PR}} = \frac{1}{\sum_i p_i^2} = \frac{\left(\sum_i \sigma_i^2\right)^2}{\sum_i \sigma_i^4}. \tag{2}$$

Like entropy effective rank (Roy & Vetterli, 2007), SPR summarizes a singular-value spectrum as the effective number of active directions. Entropy effective rank normalizes the singular values as $q_i = \sigma_i / \sum_j \sigma_j$ and computes $R_{\mathrm{ER}} = \exp(-\sum_i q_i \log q_i)$, whereas SPR normalizes their squared values, $p_i = \sigma_i^2 / \sum_j \sigma_j^2$ and computes $R_{\mathrm{PR}} = 1/\sum_i p_i^2$ which emphasizes dominant modes and is less affected by a weak spectral tail. We choose SPR because it is also highly sensitive to noise: Gaussian noise flattens the singular-value spectrum and can substantially increase SPR.

In our analysis, we compute the SPR for each layer and timestep of JiT models on ImageNet 256 and plot the results in the first row of Figure 2. A high SPR, however, can indicate either much noise or a signal distributed across many dimensions. To disambiguate these cases, we introduce an intervention-based **Noise Fraction** to measure the noise capacity in the representation. Let $\phi_{\ell,t}(\boldsymbol{x}, \boldsymbol{\epsilon})$ denote the per-token normalized form of the representation $\boldsymbol{H}^{\ell,t}(\boldsymbol{x}, \boldsymbol{\epsilon})$. We measure the noise-induced and signal-induced variations as

$$N_{\ell,t} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}_a,\boldsymbol{\epsilon}_b} \left\|\phi_{\ell,t}(\boldsymbol{x}, \boldsymbol{\epsilon}_a) - \phi_{\ell,t}(\boldsymbol{x}, \boldsymbol{\epsilon}_b)\right\|_F^2, \tag{3}$$

$$S_{\ell,t} = \mathbb{E}_{\boldsymbol{\epsilon},\boldsymbol{x}_a,\boldsymbol{x}_b} \left\|\phi_{\ell,t}(\boldsymbol{x}_a, \boldsymbol{\epsilon}) - \phi_{\ell,t}(\boldsymbol{x}_b, \boldsymbol{\epsilon})\right\|_F^2, \tag{4}$$

where $N_{\ell,t}$ varies the noise while keeping the image fixed, whereas $S_{\ell,t}$ varies images while sharing the same noise. We then define

$$\rho_{\ell,t} = \frac{N_{\ell,t}}{N_{\ell,t} + S_{\ell,t}}. \tag{5}$$

A large $\rho_{\ell,t}$ indicates a noise-sensitive representation, while a small value indicates that its variation is dominated by the clean signal. Consequently, *high SPR together with high Noise Fraction identifies high-volume noise, whereas high SPR with low Noise Fraction indicates an expansion of signal directions* . The SPR and Noise Fraction results are plotted in Figure 2. To better summarize the behaviour of each layer, we also report the mean SPR and mean Noise Fraction of each layer in the middle rows of Figure 2, where SPR is averaged over $t \in [0.1, 0.9]$ and Noise Fraction over the high-noise range $t \in [0.5, 0.9]$, in which denoising is most demanding.

Through the lens of SPR and Noise Fraction, we find that **diffusion transformer exhibits a 3-stage representation dynamics across layers and timesteps: denoising/local feature expansion - basis lift-up - posterior projection**. We summarize the model behavior as:

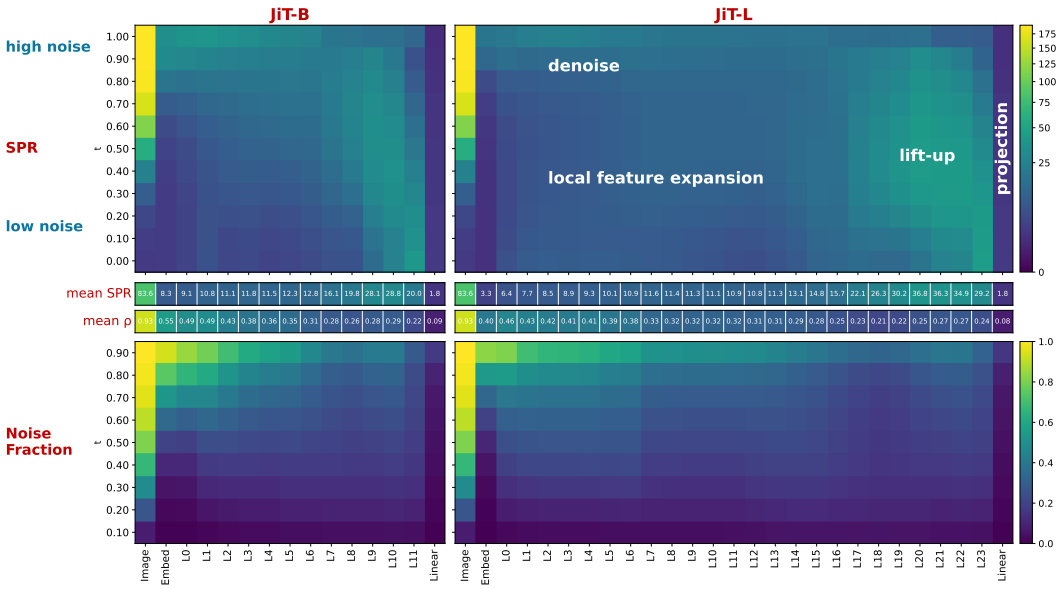

Figure 2: SPR and Noise Fraction of JiT-B (left) and JiT-L (right) on ImageNet 256×256. Columns follow the network: noisy input (Image), patch embedding (Embed), each Transformer block (L0, L1, ...), and the final linear layer (Linear); larger t means higher noise. Middle: mean SPR over $t \in [0.1, 0.9]$ and mean Noise Fraction ρ over $t \in [0.5, 0.9]$.

- denoising in shallow layers & high-noise $\boldsymbol{x}_t$: the Noise Fraction gradually reduces as the network depth increases
- local feature expansion in shallow layers & low-noise $\boldsymbol{x}_t$: the SPR increases after the FFN layer inside the transformer block while the Noise Fraction remains low.
- basis lift-up in deep layers: the SPR dramastically increases while the Noise Fraction is very low.
- Posterior projection in the final linear layer: low SPR aligning with the posterior target.

Clearly, denoising is not the only behaviour in the diffusion transformer. **But why does lift-up emerge?** Our interpretation is that the posterior mean $m_t(\boldsymbol{x}_t) = \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{x}_t]$ is generally a complex nonlinear function, whereas the final layer must predict it through a linear projection. We therefore interpret the sharp SPR increase in deep layers as *basis lift-up*: the nonlinear backbone expands posterior-relevant information into a richer, overcomplete set of basis directions in which the posterior becomes easier to approximate with a linear projection. Supporting this interpretation, our linear-probing experiments show that the clean $\boldsymbol{x}$ is already linearly decodable when $\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{x}_t] \approx \boldsymbol{x}$ from the early layers of lift-up, before the final projection. This interpretation is consistent with dimensional expansion followed by task-aligned compression in deep networks (Recanatesi et al., 2019), and echoes the feature-lifting principle behind universal approximation (Cybenko, 1989) and kernel methods (Rahimi & Recht, 2007). In Appendix A, we empirically verify that supressing the basis lift-up would degrade the generation quality in pixel diffusion. Interestingly, the depth of Transformer provides a plausible contribution to this lift-up effect: under residual updates $\boldsymbol{h}_{\ell+1} = \boldsymbol{h}_\ell + F_\ell(\boldsymbol{h}_\ell, t)$, each block can add or refine nonlinear features, yielding an iterative enrichment of the computational basis before the final projection. Consequently, a deeper Transformer has more opportunities to accumulate such target-aligned directions, which may explain why diffusion transformers usually scale well with depth (Peebles & Xie, 2023).

We further find that these representation dynamics are not exclusive to pixel-space diffusion but also arise in latent-space diffusion. Figure 3 shows the SPR and Noise Fraction of SiT-XL (Ma et al., 2024) and SiT-XL-REPA (Yu et al., 2024), both evaluated on ImageNet 256×256. Although both models are trained to

Table 1: Comparison of representation dynamics and generation quality.

Model	Depth	Denoising	Lift-up	FID
JiT-B	12	-	-	3.59
JiT-L	24	earlier	more layers	2.41
SiT-XL	28	-	-	2.54
SiT-XL-REPA	28	earlier	more layers	1.68

predict the noise-containing target $\mathbb{E}[\boldsymbol{\epsilon} - \boldsymbol{x} \mid \boldsymbol{x}_t]$, they still exhibit the same three-stage dynamics. The two comparisons in Table 1 suggest two potential characteristics of an effective diffusion representation: **denoising is completed as early as possible, leaving more subsequent layers to perform basis lift-up**. Specifically, increasing JiT's depth from 12 to 24 layers and adding REPA to SiT-XL both produce earlier denoising and a longer lift-up stage, together with improved FID. Interestingly, we find that EiT suppresses noise more effectively than JiT (Appendix B) because the U-shape bottleneck encourages the network to discard nuisance noise while retaining the principal component of the image signal (Baldi & Hornik, 1989), which resembles DAE (Vincent et al., 2008). Moreover, EiT can stack more layers under the same parameter budget (e.g., 24 layers in EiT-B vs. 12 in JiT-B), which gives the network more opportunities for basis lift-up.

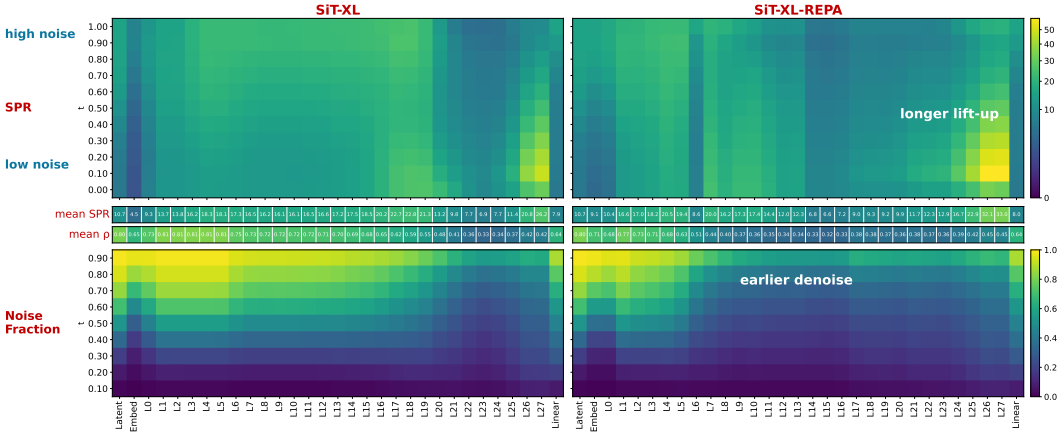

Figure 3: SPR and Noise Fraction of SiT-XL (left) and SiT-XL-REPA (right) on ImageNet 256×256, with the same layout as Figure 2 (Latent denotes the noisy VAE latent input). Compared with SiT-XL, SiT-XL-REPA denoises at earlier layers and sustains a high SPR over more deep layers.

3.4 Equivalence of Frequency-space Diffusion and Pixel-space Diffusion

After discussing the model architecture and representation dynamics, we now explore the third direction of pixel diffusion: frequency domain. Previous studies have shown that performing image generative modeling purely in the Discrete Cosine Transform (DCT) space (Ning et al., 2025) or Discrete Wavelet Transform (DWT) space (Pavasovic et al., 2026) can achieve superior generation quality than pixel-space diffusion under heuristic exploitation of frequencies and noise schedules. To reveal the role of frequency diffusion, we formally show that frequency-space and pixel-space diffusion are equivalent under an orthonormal transformation (Proposition 1).

Proposition 1 (**Equivalence under orthonormal transform and coupled randomness**). *Let* $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ *satisfy* $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}$, *and explicitly couple the pixel-space and frequency-space noises by* $\boldsymbol{\epsilon}_z := \boldsymbol{W}\boldsymbol{\epsilon}$ *with* $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. *Consider*

$$\boldsymbol{x}_t = (1-t)\boldsymbol{x} + t\boldsymbol{\epsilon}, \qquad \boldsymbol{z}_t = (1-t)\boldsymbol{W}\boldsymbol{x} + t\boldsymbol{\epsilon}_z,$$

where both spaces use the same linear schedule. Define $\widetilde{\boldsymbol{x}}_t = \boldsymbol{W}^\top \boldsymbol{z}_t$ *and the corresponding conditional velocity targets*

$$\boldsymbol{v} = \boldsymbol{\epsilon} - \boldsymbol{x}, \qquad \boldsymbol{v}_z = \boldsymbol{\epsilon}_z - \boldsymbol{W}\boldsymbol{x}.$$

For the same pixel-space predictor architecture $\boldsymbol{v}_\theta$, *define the per-sample losses*

$$L_{\text{pix}}(\theta) = \|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2, \qquad L_{\text{freq}}(\theta) = \|\boldsymbol{W}\boldsymbol{v}_\theta(\widetilde{\boldsymbol{x}}_t, t) - \boldsymbol{v}_z\|_2^2.$$

For every coupled training sample and every θ,

$$\widetilde{\boldsymbol{x}}_t = \boldsymbol{x}_t, \qquad \boldsymbol{v}_z = \boldsymbol{W}\boldsymbol{v}, \qquad L_{\text{freq}}(\theta) = L_{\text{pix}}(\theta),$$

and the two losses have identical gradients wherever differentiable. Hence their population objectives $\mathcal{L}_{\text{pix}}(\theta)$ *and* $\mathcal{L}_{\text{freq}}(\theta)$ *are equal.*

Suppose further that the two runs start from identical parameter–optimizer states and couple all stochastic training variables at every step, then pixel-space and frequency-space training follow the

same optimization trajectory:

$$\boxed{\theta_k^{\text{freq}} = \theta_k^{\text{pix}} \quad \textit{for every training step } k.}$$

Consequently, if training converges, the two models obtain exactly the same final parameters.

Corollary 1 (**Equivalence in expected loss under uncoupled randomness**). *Under the same architecture and optimizer configuration, suppose that all stochastic variables in the two runs are sampled independently according to the same laws. Then, at every training step k, the expected population losses over repeated training runs are equal:*

$$\mathbb{E}_{\text{train}}\left[\mathcal{L}_{\text{freq}}(\theta_k^{\text{freq}})\right] = \mathbb{E}_{\text{train}}\left[\mathcal{L}_{\text{pix}}(\theta_k^{\text{pix}})\right].$$

A complete proof of Proposition 1 and Corollary 1 is given in Appendix C.

Intuitively, Proposition 1 states that pixel and frequency diffusion training follow exactly the same optimization trajectory when all training randomness is shared. Corollary 1 relaxes this coupling requirement so that individual trajectories may differ, but their expected losses over repeated runs remain equal. The orthonormally normalized DCT and the critically sampled orthonormal Haar DWT are classical orthonormal transforms (Ahmed et al., 1974; Mallat, 1989) and are special cases of $\boldsymbol{W}$, so the two pipelines in Figure 4 are expected to achieve very close performance. Notably, our frequency pipeline applies $\boldsymbol{W}$ only outside the network and keeps the network operating in pixel space, so it inherits the inductive biases of pixel architectures (e.g., patch embedding and 2D RoPE), is exactly covered by Proposition 1, and adds negligible cost. Appendix D verifies this equivalence empirically and shows that feeding DWT coefficients directly to the network degrades FID.

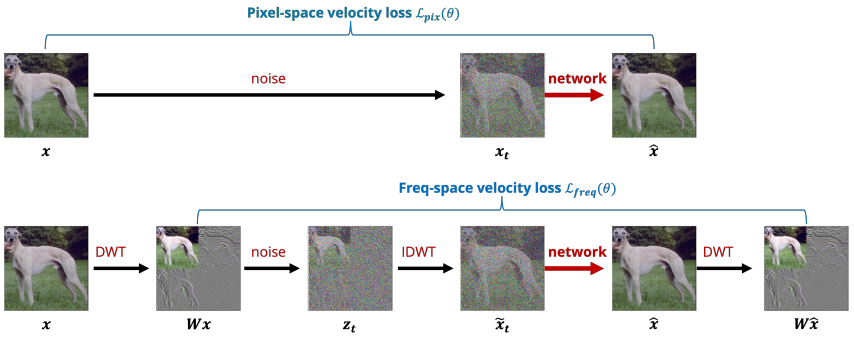

Figure 4: Pixel-space diffusion training (first row) and Frequency-space diffusion training (second row) on ImageNet 256. The orthonormal matrix $\boldsymbol{W}$ can be ethier DWT or DCT.

After clarifying the equivalence, then **what are the advantages of frequency diffusion over pixel diffusion?** Recall that the frequency diffusion loss $\mathcal{L}_{\text{freq}}(\theta)$ is derived by summing over all frequency bands. This allows frequency-band loss reweighting to reassign model capacity across frequencies, which pixel-level loss weighting in pixel diffusion cannot directly achieve, as each pixel mixes all frequencies. For example, using level-1 Haar DWT yields four frequency bands: LL, HH, HL, LH (Daubechies, 1992). We can assign higher weight to the LL component of $\mathcal{L}_{\text{freq}}(\theta)$ to enhance the generation of global structure or improve the sample quality of fine details by highlighting the HH/HL/LH components in $\mathcal{L}_{\text{freq}}(\theta)$. Let $\boldsymbol{r}_\theta = \boldsymbol{W}\boldsymbol{v}_\theta(\widetilde{\boldsymbol{x}}_t, t) - \boldsymbol{v}_z$ denote the DWT velocity residual, and let $\boldsymbol{r}_{\theta,b}$ denote its coefficients in frequency band b. We propose the DWT loss reweighting as:

$$\mathcal{L}_{\text{DWT}}(\theta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}_z,t}\left[\lambda\|\boldsymbol{r}_{\theta,\text{LL}}\|_2^2 + \gamma\left(\|\boldsymbol{r}_{\theta,\text{HH}}\|_2^2 + \|\boldsymbol{r}_{\theta,\text{HL}}\|_2^2 + \|\boldsymbol{r}_{\theta,\text{LH}}\|_2^2\right)\right]. \tag{6}$$

Empirically, we find that enhancing the low-frequency component, by setting $\lambda > 1$ and $\gamma = 1$, is beneficial on the ImageNet dataset. Thus we also propose an alternative low-frequency enhancement method for pixel diffusion: Low-Resolution Regularization (LRR). Its key idea is to construct an auxiliary flow-matching path at a lower spatial resolution. Downsampling acts as a low-pass filter, so learning the velocity along this path explicitly strengthens the modeling of low-frequency content. Let $\boldsymbol{x}_{\text{low}}$ denote the image obtained by downsampling $\boldsymbol{x}$. We independently draw $\boldsymbol{\epsilon}_{\text{low}} \sim \mathcal{N}(0, \boldsymbol{I}_{\text{low}})$

and construct $\boldsymbol{x}_{t,\text{low}} = (1-t)\boldsymbol{x}_{\text{low}} + t\boldsymbol{\epsilon}_{\text{low}}$ and $\boldsymbol{v}_{\text{low}} = \boldsymbol{\epsilon}_{\text{low}} - \boldsymbol{x}_{\text{low}}$. We augment the original velocity objective in Equation 1 with this low-resolution loss:

$$\mathcal{L}_{\text{LRR}}(\theta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},\boldsymbol{\epsilon}_{\text{low}},t}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{v}\|_2^2 + \eta\|\boldsymbol{v}_\theta(\boldsymbol{x}_{t,\text{low}},t) - \boldsymbol{v}_{\text{low}}\|_2^2\right], \tag{7}$$

Note that the network now perform twice the forward pass during training and $\boldsymbol{x}_{\text{low}}$ yields additional $1/4$ patch tokens. Thus, LRR brings extra training cost while DWT loss reweighting does not.

4 Experimental Results

Considering the simplicity, we build our pixel diffusion model EiT from JiT for pixel generation tasks and compare our models against JiT and PixelU. REPA is not used for all pixel models in our experiments. To ensure fairness, we always use the same patch size and training recipe for EiT and the base model (unless otherwise noted). The full training parameters and model details are provided in the Appendix E. We report the generation results by Frechet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) on ImageNet 256×256 and 512×512 (Russakovsky et al., 2015). We leave all ablation studies to the Appendix F.

4.1 Results of channel compression - EiT

We first evaluate the performance of EiT (channel compression) and compare it with JiT and PixelU (spatial compression). In detail, we train all models for 600 epochs following the training parameters recommended by JiT (Li & He, 2026) and split the experiments into two groups according to the model size: base and large. The generation quality is then evaluated by FID-50k and IS using 50-step Heun sampler (Karras et al., 2022). For a detailed comparison, we perform conditional generation and classifier-free guidance generation (CFG) (Ho & Salimans, 2022) separately. As summarized in Table 2, EiT consistently achieves the best FID among models of comparable size on ImageNet 256×256. Notably, despite using fewer parameters and less computation, EiT-L attains an FID of 2.22, still outperforming JiT-L (2.41) and PixelU-L (2.23). Conditional generation usually provides a more direct measure of the quality of the learned score function. Remarkably, after only 200 epochs, both EiT-B and EiT-L already outperform their JiT counterparts trained for 600 epochs in conditional FID (see Figure 5), demonstrating substantially greater training efficiency.

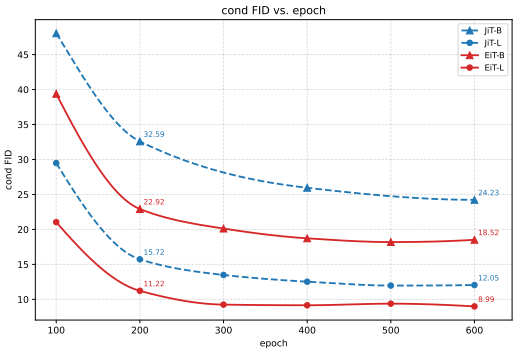

Figure 5: Conditional FID of Pixel Diffusion Models on ImageNet 256×256.

Table 2: Generation Comparison of Pixel Diffusion Models on ImageNet 256×256.

Models	depth	hidden dim	# para	GFLOPs	cond		CFG	
					FID ↓	IS ↑	FID ↓	IS ↑
JiT-B	12	768	131M	25.2	24.23	65.5	3.59	262.6
PixelU-B	12	768	158M	22.4	21.27	73.6	3.53	269.8
EiT-B	24	≤768	130M	24.3	18.52	76.9	3.10	275.3
EiT-B (DWT)	24	≤768	130M	24.3	**18.44**	**77.7**	**3.02**	**277.2**
JiT-L	24	1024	459M	87.9	12.05	111.2	2.41	290.1
PixelU-L	24	1024	516M	75.3	9.66	112.2	2.23	291.6
EiT-L	30	≤1024	363M	67.1	9.53	120.8	2.22	303.3
EiT-L	30	≤1152	460M	84.4	8.99	121.1	2.06	300.2
EiT-L (DWT)	30	≤1152	460M	84.4	**8.63**	**127.0**	**2.03**	**312.1**

Table 3: Generation Comparison of Pixel Diffusion Models on ImageNet 512×512.

Models	depth	hidden dim	# para	GFLOPs	cond		CFG	
					FID ↓	IS ↑	FID ↓	IS ↑
JiT-B	12	768	133M	25.8	27.96	63.2	4.18	271.2
EiT-B	24	≤768	132M	24.7	19.56	82.8	3.50	288.9
EiT-B (DWT)	24	≤768	132M	24.7	19.81	81.3	**3.44**	287.8
JiT-L	24	1024	462M	88.6	13.32	108.9	2.67	299.1
EiT-L	30	≤1152	463M	85.1	10.83	123.9	2.43	321.5
EiT-L (DWT)	30	≤1152	463M	85.1	**9.56**	**131.3**	**2.40**	**329.9**

4.2 Results of Frequency-Space Diffusion: EiT (DWT)

In Section 3.4, we established the equivalence between frequency-space and pixel-space diffusion and showed that the frequency-space formulation enables flexible frequency-band loss reweighting at negligible additional cost. To validate the effectiveness of DWT and its loss reweighting, we train EiT (DWT) using Equation 6 with $\lambda = 1.2$ and $\gamma = 1$ (namely highlighting the training of low-frequency components). As shown in Tables 2 and 3, EiT (DWT) consistently improves generation quality on both ImageNet 256×256 and 512×512 without increasing the parameter count or model GFLOPs. We also find that Low-Resolution Regularization (LRR) yields similar improvements for pixel-space diffusion (Appendix F.2), but incurs approximately 28% additional training GFLOPs. These results demonstrate that DWT provides an effective and computationally efficient realization of frequency-space diffusion.

4.3 Extension to Latent-Space Diffusion

To assess the generality of channel compression, we extend EiT to latent-space diffusion by incorporating its channel-compression architecture into SiT (Ma et al., 2024). We evaluate the resulting models both with and without REPA (Yu et al., 2024) on ImageNet 256×256, training all Base models for 80 epochs and all XL models for 200 epochs. As shown in Table 4, EiT consistently outperforms its SiT counterpart under comparable parameter and computation budgets. Without REPA, EiT-B reduces CFG FID from 6.39 to 4.99. The gains remain consistent when combined with REPA: EiT-B-REPA improves the FID scores from 3.39 to 3.09. These results demonstrate that channel compression is effective not only for pixel-space diffusion but also for latent-space diffusion. Because the spectrum of SD-VAE latents does not have a semantics correspondance to pixel-space spectrum (Ning et al., 2026), we therefore do not apply DWT-based loss reweighting in latent diffusions.

Table 4: Generation Comparison of Latent Diffusion Models on ImageNet 256×256.

Models	# para	GFLOPs	Epoch	FID	
				cond	CFG
SiT-B	130M	23.0	80	39.81	6.39
EiT-B	129M	22.3	80	**29.17**	**4.99**
SiT-B-REPA	138M	24.9	80	24.55	3.39
EiT-B-REPA	136M	24.1	80	**19.18**	**3.09**
SiT-XL-REPA	683M	120.7	200	7.40	1.68
EiT-XL-REPA	685M	117.2	200	**7.13**	**1.64**

5 Conclusion

In this paper, we explored the design factors of pixel-space diffusion from three perspectives. For model architecture, we proposed EiT, which introduces channel compression into the pixel diffusion transformer and outperforms JiT and PixelU under comparable budgets, while also benefiting latent diffusion. For representation, we revealed a three-stage dynamics of denoising/local feature expansion, basis lift-up, and posterior projection shared by pixel and latent diffusion, and found that earlier denoising and a longer lift-up stage are associated with better generation. For the frequency domain, we proved the equivalence between pixel-space and frequency-space diffusion and showed that frequency-band loss reweighting further improves generation at no extra cost. A limitation of this work is that our experiments are restricted to class-conditional ImageNet generation. We have not yet validated our method on text-to-image or video generation, which we leave as future work.

AI USE STATEMENT

In this work, we used generative AI tools (the Cursor editor with GPT-5.6) to implement our methods in code, and to assist in converting the proofs of Proposition 1 and Corollary 1 from Markdown to LaTeX and to check the derivations. The statements and all proof ideas were developed by the authors. We have not used generative AI tools to develop the core ideas or theoretical framework of this work (e.g., the three-stage representation dynamics and the pixel–frequency equivalence), formulate mathematical claims, provide ingredients for the proofs, propose hypotheses, design the algorithms, or interpret the experimental results, and translation, generating synthetic datasets, cleaning or reformatting datasets, and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to suggest experimental hyperparameters, to create the scientific figures, to draft parts of the paper in English, to edit the paper for readability, and to search for related literature. We did not use generative AI tools to generate research ideas or to design the structure of the paper. We have reviewed all AI-assisted work. All AI-generated code was reviewed and tested by the authors, and all reported results were obtained from our own training and evaluation runs; final hyperparameters were selected by the authors based on these runs. The proofs were checked step by step by the authors. All AI-drafted and AI-edited text was revised by the authors to ensure that it accurately reflects our ideas and results, and every reference found with AI assistance was checked against the original paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We believe that our results can be readily reproduced. EiT is built on the publicly available code of JiT (Li & He, 2026) with minimal changes: channel compression is realized only by varying the number of attention heads across stages, while the tokenization, positional embedding, training recipe, and sampler of JiT are kept unchanged (Section 3.2). Likewise, EiT (DWT) only applies an orthonormal Haar DWT outside the network and reweights the frequency-band losses (Figure 4 and Equation 6). The complete architecture configurations, training hyperparameters, and sampling settings of all pixel diffusion models are listed in Appendix E and Table 6, the u-type designs are detailed in Appendix F.1, and the settings of the latent diffusion experiments are given in Section 4.3. The SPR and Noise Fraction used in our representation analysis are precisely defined in Section 3.3. For the theoretical results, all assumptions are stated in Proposition 1 and Corollary 1, and complete proofs are provided in Appendix C. All experiments are conducted on the public ImageNet dataset (Russakovsky et al., 2015) and evaluated with the standard FID-50k and IS metrics. Finally, we will publicly release our code and pretrained model weights to facilitate the reproduction of all results in this paper.

REFERENCES

Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

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. In *CVPR*, pp. 22669–22679. IEEE, 2023.

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

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. In *CVPR*, pp. 36136–36146, 2026.

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

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.

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. In *ECCV*, pp. 425–442. Springer, 2026.

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. *NeurIPS*, 30, 2017.

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

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

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

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *CVPR*, pp. 18062–18071. IEEE, 2025.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. Representation alignment for diffusion transformers without external components. In *International Conference on Learning Representations*, volume 2026, pp. 18884–18915, 2026.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

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

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. *NeurIPS*, 37:122458–122483, 2024.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *CVPR*, pp. 36115–36125, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ICLR*, 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, pp. 23–40. Springer, 2024.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. In *CVPR*, pp. 43600–43610, 2026.

Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

Mang Ning, Mingxiao Li, Jianlin Su, Haozhe Jia, Lanmiao Liu, Martin Beneš, Wenshuo Chen, Albert Ali Salah, and Itir Onal Ertugrul. Dctdiff: Intriguing properties of image generative modeling in the dct space. *ICML*, 2025.

Mang Ning, Mingxiao Li, Le Zhang, Lanmiao Liu, Matthew B Blaschko, Albert Ali Salah, and Itir Onal Ertugrul. Spectrum matching: a unified perspective for superior diffusability in latent diffusion. *arXiv preprint arXiv:2603.14645*, 2026.

Krunoslav Lehman Pavasovic, Théophane Vallaeys, Stéphane Mallat, Giulio Biroli, Luke Zettlemoyer, Brian Karrer, and Jakob Verbeek. Wait for the signal: Simple frequency-aware flow-matching. *arXiv preprint arXiv:2607.28760*, 2026.

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

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *CVPR*, pp. 10619–10629. IEEE, 2022.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *NeurIPS*, 20, 2007.

Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

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

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

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

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *NeurIPS*, 36:1363–1389, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pp. 1096–1103, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(110):3371–3408, 2010.

Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *ICCV*, pp. 16284–16294. IEEE, 2023.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *ICCV*, pp. 18938–18949. IEEE, 2023.

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. *ICLR*, 2024.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *CVPR*, pp. 14273–14282, 2026.

Le Zhang, Ning Mang, and Aishwarya Agrawal. Rit: Vanilla diffusion transformers suffice in representation space. *arXiv preprint arXiv:2605.21981*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, volume 2026, pp. 35791–35820, 2026.

A Effect of supressing the lift-up

To verify whether basis lift-up benefits generation quality, we train a variant, JiT-B (SPR), which explicitly suppresses lift-up by adding an SPR penalty to the training loss. Specifically, the SPR of any layer exceeding 16 is penalized with a loss weight of 0.1. Both JiT-B (SPR) and JiT-B are trained for 600 epochs on ImageNet 256×256. Suppressing the lift-up degrades generation quality, increasing the FID from 3.59 to 3.79 (with CFG), suggesting that basis lift-up is beneficial for pixel diffusion.

B Diffusion Representation Dynamics

Figure 6 compares the SPR and Noise Fraction of JiT-B (12 blocks) and EiT-B (24 blocks, u-type 4-3-2-1) on ImageNet 256×256, following the protocol in Section 3.3. The mean rows average SPR over $t \in [0.1, 0.9]$ and Noise Fraction over the high-noise range $t \in [0.5, 0.9]$, where denoising is most demanding. EiT-B exhibits the same three-stage dynamics as JiT-B, with two notable differences.

Earlier and stronger denoising. The mean Noise Fraction of EiT-B drops to 0.28 at L6, about one third of its depth, whereas JiT-B reaches the same level only at L7, about two thirds of its depth. EiT-B then keeps the Noise Fraction at or below 0.25 over 13 consecutive blocks (L7–L19) and reaches 0.18 at L16, lower than the minimum of JiT-B (0.22). This supports our interpretation that the channel bottleneck, like a DAE, discards noise while retaining the dominant image signal.

Longer basis lift-up. With noise removed early, EiT-B leaves more deep layers for basis lift-up. JiT-B exhibits an SPR above 24 only at L9 and L10, with a peak of 28.8, whereas EiT-B maintains an SPR above 24 over six consecutive blocks (L17–L22), with a peak of 54.0 at L20. Note that the SPR spikes in shallow layers (e.g., L2 and L5) are accompanied by a relatively high Noise Fraction (0.58 and 0.37), and thus belong to the denoising/local feature expansion stage rather than lift-up. Overall, EiT falls into the regime of an effective diffusion representation identified in Section 3.3: early denoising followed by a longer lift-up stage.

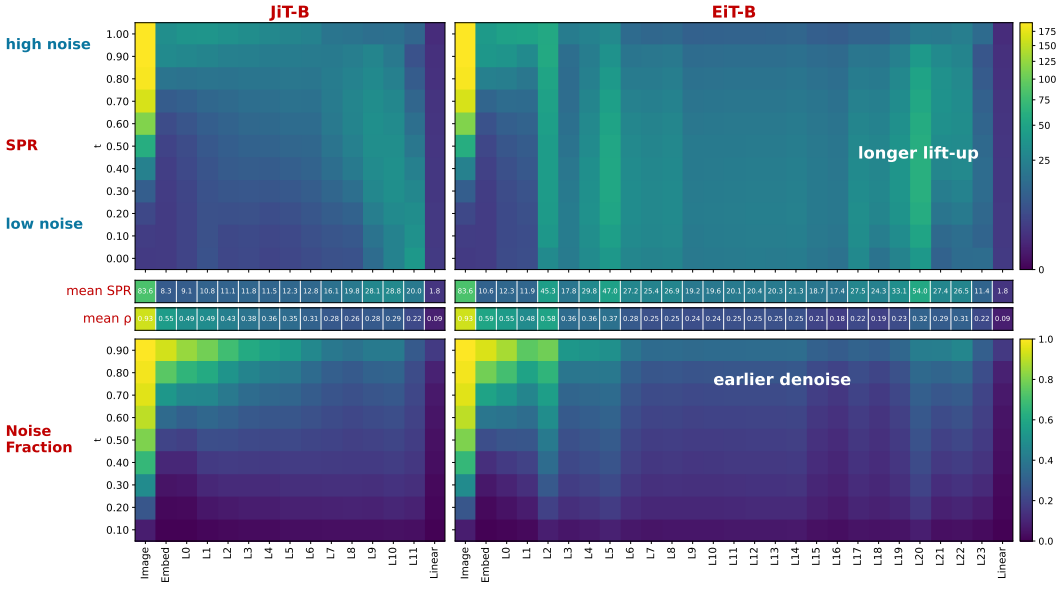

Figure 6: SPR and Noise Fraction of JiT-B (12 blocks, left) and EiT-B (24 blocks, u-type 4-3-2-1, right) on ImageNet 256×256, with the same layout as Figure 2. Compared with JiT-B, EiT-B reduces the Noise Fraction at an earlier relative depth (earlier denoising) and maintains a high SPR over more deep blocks (longer lift-up).

C Proofs for Orthonormal Transform Equivalence

Proof of Proposition 1. The explicit coupling $\boldsymbol{\epsilon}_z = \boldsymbol{W}\boldsymbol{\epsilon}$ gives

$$\boldsymbol{z}_t = (1-t)\boldsymbol{W}\boldsymbol{x} + t\boldsymbol{W}\boldsymbol{\epsilon} = \boldsymbol{W}\boldsymbol{x}_t,$$

and therefore $\widetilde{\boldsymbol{x}}_t = \boldsymbol{W}^\top \boldsymbol{z}_t = \boldsymbol{x}_t$. The frequency-space velocity target satisfies

$$\boldsymbol{v}_z = \boldsymbol{W}\boldsymbol{\epsilon} - \boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{v}.$$

Thus, for every realization of all coupled training randomness and every θ, orthonormality yields

$$\begin{aligned} L_{\text{freq}}(\theta) &= \|\boldsymbol{W}\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{W}\boldsymbol{v}\|_2^2 \\ &= \|\boldsymbol{W}(\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v})\|_2^2 \\ &= \|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2 = L_{\text{pix}}(\theta). \end{aligned}$$

Because the two sides are the same scalar function of θ,

$$\nabla_\theta L_{\text{freq}}(\theta) = \nabla_\theta L_{\text{pix}}(\theta)$$

wherever differentiable. Averaging over a coupled minibatch preserves this gradient identity, as does taking the expectation that defines the population objectives.

Let s_k^{pix} and s_k^{freq} denote the optimizer states. By assumption,

$$(\theta_0^{\text{pix}}, s_0^{\text{pix}}) = (\theta_0^{\text{freq}}, s_0^{\text{freq}}).$$

Assume the two parameter–optimizer-state pairs are equal at step k. Since all training randomness at this step is coupled, the two runs evaluate equal minibatch gradients. Applying the same optimizer update to equal parameters, states, and gradients gives

$$(\theta_{k+1}^{\text{pix}}, s_{k+1}^{\text{pix}}) = (\theta_{k+1}^{\text{freq}}, s_{k+1}^{\text{freq}}).$$

Induction proves equality at every step. Any converged parameter vectors are therefore identical. □

Proof of Corollary 1. Let Ξ_k^{pix} and Ξ_k^{freq} collect all stochastic inputs at step k, including the minibatch, noise, timestep, conditioning variables, data augmentation, and model randomness. The two collections are sampled independently across runs but, by assumption, their corresponding components follow the same joint sampling laws. For the frequency-space noise, define $\boldsymbol{\eta} = \boldsymbol{W}^\top \boldsymbol{\epsilon}_z$. Orthonormal invariance gives

$$\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{W}^\top \boldsymbol{I}\boldsymbol{W}) = \mathcal{N}(0, \boldsymbol{I}).$$

After mapping the frequency-space quantities back to pixel space,

$$\widetilde{\boldsymbol{x}}_t = (1-t)\boldsymbol{x} + t\boldsymbol{\eta}, \qquad \boldsymbol{W}^\top \boldsymbol{v}_z = \boldsymbol{\eta} - \boldsymbol{x}.$$

Together with the matching laws of all other components of Ξ_k, this shows that the complete transformed frequency-space minibatch has the same distribution as the pixel-space minibatch. Norm preservation then implies, for every fixed θ,

$$\big(L_{\text{freq}}(\theta), \nabla_\theta L_{\text{freq}}(\theta)\big) \overset{d}{=} \big(L_{\text{pix}}(\theta), \nabla_\theta L_{\text{pix}}(\theta)\big).$$

The same identity holds for the corresponding stochastic minibatch losses and gradients.

Let $S_k^{\text{pix}} = (\theta_k^{\text{pix}}, s_k^{\text{pix}})$ and $S_k^{\text{freq}} = (\theta_k^{\text{freq}}, s_k^{\text{freq}})$ denote the parameter–optimizer states. The initial states are sampled from the same distribution; identical deterministic initialization is a special case. Because both runs use the same optimizer configuration, the result above implies that, conditional on any fixed state, their next-state updates have the same transition distribution. If $S_k^{\text{freq}} \overset{d}{=} S_k^{\text{pix}}$, applying these identical transition distributions gives $S_{k+1}^{\text{freq}} \overset{d}{=} S_{k+1}^{\text{pix}}$. Induction from the initial states therefore yields

$$(\theta_k^{\text{freq}}, s_k^{\text{freq}}) \overset{d}{=} (\theta_k^{\text{pix}}, s_k^{\text{pix}}) \quad \text{for every } k.$$

Moreover, the same input–target distribution and norm preservation give $\mathcal{L}_{\text{freq}}(\theta) = \mathcal{L}_{\text{pix}}(\theta)$ for every fixed θ. Combining this pointwise identity with equality in distribution of the parameter iterates yields

$$\mathbb{E}_{\text{train}}\left[\mathcal{L}_{\text{freq}}(\theta_k^{\text{freq}})\right] = \mathbb{E}_{\text{train}}\left[\mathcal{L}_{\text{pix}}(\theta_k^{\text{pix}})\right].$$

The statement for converged iterates follows whenever the terminal iterates and corresponding expected losses are well defined. Since the stochastic variables are not coupled across runs, this argument does not imply equality of individual optimization trajectories. □

D Empirical Verification of Equivalence

To empirically verify the equivalence in Section 3.4, we train JiT-B in pixel space and JiT-B (DWT) in the Haar DWT space following Figure 4, without loss reweighting ($\lambda = \gamma = 1$). The training randomness of the two runs is sampled independently, matching the uncoupled setting of Corollary 1. As shown in Table 5, the two models achieve nearly identical FID after 600 epochs on ImageNet 256×256: 24.23 vs. 24.31 for conditional generation and 3.59 vs. 3.60 with CFG. Moreover, since the DWT is applied outside the network, JiT-B (DWT) keeps the same parameters and GFLOPs as JiT-B, incurring negligible extra training cost.

Table 5: Generation Comparison of Pixel Diffusion and Frequency Diffusion on ImageNet 256×256.

Models	# para	GFLOPs	Epoch	FID	
				cond	CFG
JiT-B	131M	25.2	600	24.23	3.59
JiT-B (DWT)	131M	25.2	600	24.31	3.60
JiT-B (vanilla DWT)	131M	25.2	600	25.11	3.71

Feeding DWT coefficients directly to the network. We further train JiT-B (vanilla DWT), which feeds the noisy DWT coefficients $\boldsymbol{z}_t$ directly into the network, instead of first mapping them back to the pixel-space input $\widetilde{\boldsymbol{x}}_t = \boldsymbol{W}^\top \boldsymbol{z}_t$ as in Figure 4. It is trained with the same DWT velocity loss ($\lambda = \gamma = 1$), architecture, and training recipe as JiT-B (DWT), so the two models differ only in the input space of the network. As shown in Table 5, JiT-B (vanilla DWT) degrades the FID from 24.31 to 25.11 for conditional generation and from 3.60 to 3.71 with CFG. This gap is consistent with our theory, since Proposition 1 requires the same pixel-space predictor to be applied to $\boldsymbol{W}^\top \boldsymbol{z}_t$. Once the network directly receives $\boldsymbol{z}_t$, the equivalence no longer holds: the architecture is designed for the spatial structure of RGB images rather than for DWT coefficients, and adaptive optimizers such as Adam are not invariant to an orthonormal change of the input coordinates. In addition, the DWT sub-bands have very different dynamic ranges, with large LL coefficients and sparse, near-zero high-frequency coefficients, which may make training numerically more sensitive. In contrast, the design in Figure 4 applies the DWT only outside the network and keeps the network operating in pixel space. It therefore inherits the inductive biases of pixel-space architectures, is exactly covered by Proposition 1, and still enables frequency-band loss reweighting. We thus adopt this design for all frequency-space models in this paper.

E Training Parameters and Model Details

Table 6 lists the configurations of JiT and EiT. EiT is built on JiT and shares its patch size, 128-dimensional bottleneck patch embedding, 32 in-context class tokens, training recipe, and sampler, so that the two models differ only in architecture.

Architecture. Following Section 3.2, EiT keeps the head dimension fixed ($d = 64$ for EiT-B and $d = 72$ for EiT-L) and compresses channels by reducing the number of heads. The u-type specifies the encoder stage widths in units of $D/4$, where D is the maximum hidden dimension, and the decoder mirrors this schedule. Thus, 4-3-2-1 in EiT-B corresponds to widths $(768, 576, 384, 192)$ with $(12, 9, 6, 3)$ heads, and its 24 blocks are evenly distributed over the eight encoder and decoder stages (3 blocks per stage). Similarly, 4-3-2 in EiT-L corresponds to widths $(1152, 864, 576)$ with $(16, 12, 8)$ heads, and its 30 blocks form six stages of 5 blocks each. Hence, the hidden dimension and number of heads of EiT are marked with $\leq$ in the table. Adjacent stages are connected by linear projections, and mirrored encoder and decoder blocks by long skips. Since narrow stages are cheap,

EiT stacks more blocks than JiT under a comparable budget (24 vs. 12 for Base and 30 vs. 24 for Large). For all models, the in-context class tokens are inserted at one third of the depth.

Training and sampling. All models are trained for 600 epochs with Adam (Kingma & Ba, 2014), a batch size of 2048, and a constant learning rate of 4×10^{-4} after 5 warmup epochs. We keep two EMA models with decays 0.9999 and 0.9996, and drop class labels with probability 0.1 to enable CFG. We use patch sizes 16 and 32 for ImageNet 256×256 and 512×512, respectively, yielding the same 256 tokens, and increase the noise scale from 1.0 to 2.0 at the higher resolution. For sampling, we use a 50-step Heun solver with linearly spaced timesteps, apply CFG within the interval $t \in [0.1, 1.0]$ (Kynkäänniemi et al., 2024), and sweep the CFG scale over $[2.0, 4.0]$.

Table 6: Configurations of experiments.

	ImageNet 256				ImageNet 512			
	JiT-B	**EiT-B**	**JiT-L**	**EiT-L**	**JiT-B**	**EiT-B**	**JiT-L**	**EiT-L**
architecture								
depth	12	24	24	30	12	24	24	30
hidden dim	768	≤768	1024	≤1152	768	≤768	1024	≤1152
u-type	-	4-3-2-1	-	4-3-2	-	4-3-2-1	-	4-3-2
heads	12	≤12	16	≤16	12	≤12	16	≤16
parameters	131M	130M	459M	460M	133M	132M	462M	463M
patch size	16				32			
bottleneck dim	128				128			
dropout	0				0			
in-context class tokens	32				32			
in-context start block	4	8	8	10	4	8	8	10
training								
epochs	600				600			
warmup epochs	5				5			
optimizer	Adam				Adam			
batch size	2048				2048			
learning rate	4e-4				4e-4			
learning rate schedule	constant				constant			
ema decay	{0.9999, 0.9996}				{0.9999, 0.9996}			
noise scale	1.0				2.0			
class token drop	0.1				0.1			
sampling								
ODE solver	Heun				Heun			
ODE steps	50				50			
time steps	linear in [0.0, 1.0]				linear in [0.0, 1.0]			
CFG scale sweep range	[2.0, 4.0]				[2.0, 4.0]			
CFG interval	[0.1, 1.0]				[0.1, 1.0]			

F Ablation Studies

F.1 Design of U-type in EiT

The u-type determines how EiT compresses channels. It lists the encoder stage widths in units of $D/4$, where D is the maximum hidden dimension, and the decoder mirrors this schedule. Since the number of blocks per stage is fixed (3 for EiT-B and 5 for EiT-L), the u-type controls both the depth and the bottleneck width: 4-3-2-1 has eight stages with a bottleneck of $D/4$, whereas 4-3-2 has six stages with a wider bottleneck of $D/2$. For example, EiT-B has 24 blocks with 4-3-2-1 and 18 blocks with 4-3-2. Adding a narrower stage therefore deepens the network at the cost of a narrower bottleneck.

Table 7 compares the two schedules, and all EiT variants outperform their JiT counterparts. For the Base models, 4-3-2-1 (24 blocks, bottleneck 192) achieves a conditional FID comparable to 4-3-2 (18 blocks, bottleneck 384) but a clearly better CFG FID (3.10 vs. 3.34). For the Large models, the trend reverses. At the same maximum width of 1024, 4-3-2 (30 blocks, bottleneck 512) outperforms 4-3-2-1 (40 blocks, bottleneck 256) in conditional FID (9.53 vs. 10.08) and remains comparable in CFG FID (2.22 vs. 2.19), while using fewer parameters and FLOPs. Increasing the maximum width to 1152 further widens the bottleneck to 576 and yields the best results (8.99 and 2.06). These results suggest that, under the EiT architecture, depth should be scaled first when the network is shallow, as it provides more layers for basis lift-up (Section 3.3). Once the depth is sufficient, a large enough bottleneck dimension becomes more important for preserving the image signal.

Table 7: Ablation study of the u-type design in EiT on ImageNet 256×256.

	u-type	depth	hidden dim	# para	GFLOPs	epochs	FID (cond)	FID (CFG)
JiT-B	-	12	768	131M	25.2	600	24.23	3.59
EiT-B	4-3-2-1	24	≤768	130M	24.3	600	**18.52**	**3.10**
EiT-B	4-3-2	18	≤768	125M	23.3	600	18.55	3.34
JiT-L	-	24	1024	459M	87.9	600	12.05	2.41
EiT-L	4-3-2-1	40	≤1024	378M	69.8	600	10.08	2.19
EiT-L	4-3-2	30	≤1024	363M	67.1	600	9.53	2.22
EiT-L	4-3-2	30	≤1152	460M	84.4	600	**8.99**	**2.06**

F.2 Loss Reweighting of DWT and Low-Resolution Regularization

Section 3.4 introduces two ways to strengthen the modeling of low-frequency content: DWT loss reweighting (Equation 6) and Low-Resolution Regularization (LRR, Equation 7). Table 8 compares the two methods on EiT-B (4-3-2-1) on ImageNet 256×256, where all models are trained for 600 epochs with the same recipe, and EiT-B (DWT) uses different λ and $\gamma = 1$ as in Section 4.2. For EiT-B (LRR), the low-resolution image $\boldsymbol{x}_{\text{low}}$ is obtained by bicubic downsampling, and the weight of the low-resolution loss in Equation 7 is set to $\eta = 1.2$. By Proposition 1, the pixel-space EiT-B is equivalent to its DWT counterpart with $\lambda = \gamma = 1$, so the comparison between EiT-B and EiT-B (DWT) isolates the effect of up-weighting the LL band.

Similar gains. Both methods improve the CFG FID of EiT-B by a similar margin: DWT loss reweighting reduces it from 3.10 to 3.02, and LRR reduces it to 3.00. This agreement suggests that the two methods benefit from the same source, namely emphasizing low-frequency content during training, which is beneficial for generation on ImageNet.

Different costs. The two methods differ substantially in training cost. Since the orthonormal DWT is applied outside the network, EiT-B (DWT) keeps exactly the same parameters (130M) and GFLOPs (24.3) as EiT-B. In contrast, LRR requires an additional forward pass on the low-resolution input at every training iteration, which raises the training cost from 24.3 to 31.0 GFLOPs (about 28% more) while yielding a nearly identical FID (3.00 vs. 3.02). We therefore adopt DWT loss reweighting as the default low-frequency enhancement in our main experiments, as it achieves comparable improvements at no extra cost.

Table 8: Ablation study of the DWT loss reweighting and Low-Resolution Regularization in EiT on ImageNet 256×256.

	u-type	depth	hidden dim	# para	GFLOPs	epochs	FID (CFG)
EiT-B	4-3-2-1	24	≤768	130M	24.3	600	3.10
EiT-B (DWT, $\lambda = 1.1$)	4-3-2-1	24	≤768	130M	24.3	600	3.05
EiT-B (DWT, $\lambda = 1.2$)	4-3-2-1	24	≤768	130M	24.3	600	**3.02**
EiT-B (DWT, $\lambda = 1.3$)	4-3-2-1	24	≤768	130M	24.3	600	3.07
EiT-B (LRR, $\eta = 1.2$)	4-3-2-1	24	≤768	131M	31.0	600	**3.00**