FREPIX: FREQUENCY-HETEROGENEOUS FLOW MATCHING FOR PIXEL-SPACE IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel-space diffusion has re-emerged as a promising alternative to latent-space generation because it avoids the representation bottleneck introduced by VAEs. Yet most existing methods still treat image generation as a frequency-homogeneous process, overlooking the distinct roles and learning dynamics of low- and high-frequency components. To address this, we propose FREPix, a **FRE**quency-heterogeneous flow matching framework for **Pix**el-space image generation. FREPix explicitly decomposes generation into low- and high-frequency components, assigns them separate transport paths, predicts them with a factorized network, and trains them with a frequency-aware objective. In this way, coarse-to-fine generation becomes an explicit design principle rather than an implicit behavior. On ImageNet class-to-image generation, FREPix achieves competitive results among pixel-space generation models, reaching 1.91 FID at 256×256 and 2.38 FID at 512×512, with particularly strong performance in the early stages of training and in the low-NFE regime.

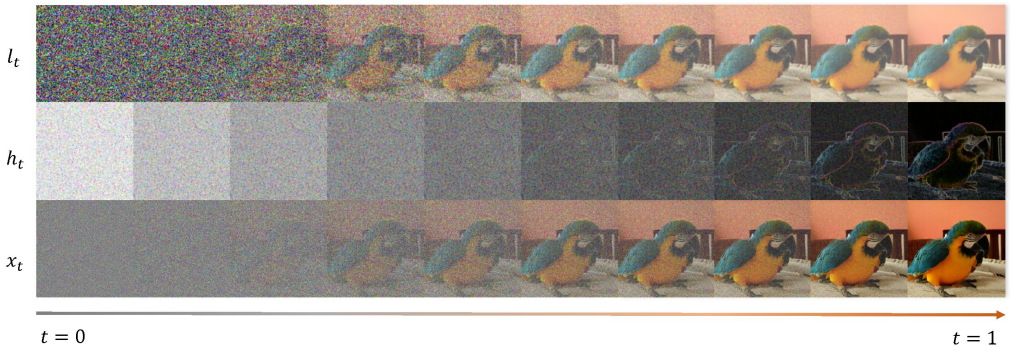


Figure 1: **Visualization of frequency decoupling in FREPix.** Evolution of low-frequency sub-state l_t (top), high-frequency sub-state h_t (middle), and final image x_t (bottom) over time $t \in [0, 1]$.

1 INTRODUCTION

Latent diffusion (Dhariwal & Nichol, 2021; Rombach et al., 2022; Peebles & Xie, 2023; Leng et al., 2025) has become the dominant paradigm for image generation by moving denoising from raw pixels to a compact latent space, which greatly reduces spatial complexity and makes large-scale training practical. But this efficiency comes with a structural cost. Generation is no longer performed in the original image domain, and image quality is inevitably tied to the representation and reconstruction fidelity of the VAEs (Yao et al., 2025; Shi et al., 2025; Hansen-Estruch et al., 2025). These limitations have renewed interest in pixel-space generation, where models operate directly on raw images and avoid the representational bottleneck introduced by latent space encodings.

Despite this appeal, pixel-space generation remains fundamentally difficult. Raw images are high-dimensional, spatially dense and entangle global semantics with local details in a single state space (Rahaman et al., 2019). Recent progress has made this paradigm increasingly viable through coarse-to-fine architectures (Ho et al., 2022; Teng et al., 2024) and stronger pixel-level modeling (Yu et al., 2026; Wang et al., 2025; Chen et al., 2025). Still, most existing methods treat image generation as a homogeneous process. They model the whole image with a single state and leave the separation between global structures and fine details to emerge implicitly during learning.

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.

Natural images are not organized uniformly across frequencies. Low-frequency components mainly determine layout, color composition, and semantic structure, while high-frequency components are more closely associated with edges, textures, and perceptual sharpness (Torralba & Oliva, 2003; Chen et al., 2019). More importantly, these two differ not only in visual role, but also in their underlying statistics and learning dynamics (Xu, 2020). As illustrated in Fig. 2, the stark divergence in their energy distributions provides an empirical evidence of this heterogeneity. Therefore, treating these components with a single state representation, a shared interpolation path, and a unified modeling strategy imposes an unnecessarily restrictive inductive bias on pixel-space generation.

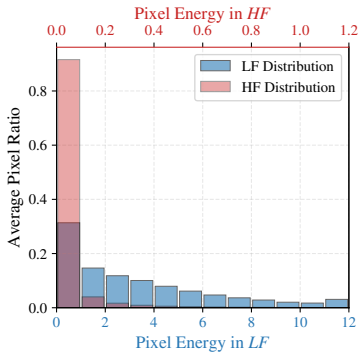


Figure 2: **Frequency heterogeneity in natural images.** The low-frequency component exhibits larger per-location energy (up to 12.0 vs. 1.2) and a broader distribution than the high-frequency component. The energy is measured by the squared ℓ_2 norm of the corresponding low-/high-frequency coefficients at each location.

In this paper, we propose FREPix, a **FRE**quency-heterogeneous flow matching framework for **Pix**el-space image generation. Natural images are frequency-heterogeneous, yet current pixel-space generation is still formulated largely as a frequency-homogeneous process. FREPix makes this heterogeneity explicit throughout generation. It decomposes the image into low- and high-frequency sub-states, assigns them heterogeneous interpolation paths, and explicitly decomposes the generation target: a low-frequency backbone first predicts the clean low-frequency component, while a high-frequency decoder then predicts the corresponding high-frequency component conditioned on the predicted low-frequency component. Training objective is further aligned with this factorization through a specifically designed frequency-aware flow matching objective. In this way, FREPix turns coarse-to-fine generation from an implicit behavior that the network is expected to discover into an explicit design principle for pixel-space flow matching.

Extensive experiments validate the effectiveness of FREPix in class-to-image generation. It achieves competitive results among pixel-space generation models on ImageNet, reaching 1.91 FID at 256×256 and 2.38 FID at 512×512, while also attaining competitive quality at an early stage of training and under low-NFE sampling. Together, these results show that explicitly modeling frequency heterogeneity provides a stronger inductive bias for end-to-end pixel-space generation.

2 Related Work

Latent-Space and Pixel-Space Image Generation. Modern image generation methods can be broadly divided into latent-space and pixel-space approaches. Latent diffusion models (Rombach et al., 2022) improve efficiency by performing denoising in a compressed representation, and have been further strengthened by transformer-based formulations such as DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024). However, their performance is inherently tied to the quality of the autoencoder, which may limit fine-detail reconstruction and introduce decoding artifacts. This has renewed interest in direct pixel-space generation (Ho et al., 2020; Nichol & Dhariwal, 2021), where recent models such as JiT (Li & He, 2025), PixelDiT (Yu et al., 2026), PixNerd (Wang et al., 2025), DeCo (Ma et al., 2026a) and PixelGen (Ma et al., 2026b) show that raw-pixel modeling can achieve competitive image quality. Most existing pixel-space methods, however, still model the image as a largely homogeneous state or introduce frequency cues mainly through architecture or auxiliary losses, without explicitly defining separate transport paths for different frequency components.

Frequency-Aware and Coarse-to-Fine Image Generation. A long line of work has explored the observation that images are naturally generated in a coarse-to-fine manner. Cascaded diffusion models (Ho et al., 2022) decompose generation across resolutions, while multi-scale pixel-space methods such as SiD2 (Hoogeboom et al., 2025) and PixelFlow (Chen et al., 2025) reduce the difficulty of raw-pixel modeling through structured resolution schedules. Frequency-aware methods further exploit the distinction between frequency. For example, WDM (Phung et al., 2023) applies wavelet-domain modeling, PixelDiT (Yu et al., 2026) separates global and local information through a dual-level design, and DeCo (Ma et al., 2026a) combines a low-frequency backbone with high-frequency

refinement. EqualSNR (Falck et al., 2025) study how training signals vary across frequencies, whereas our method focuses on defining frequency-specific flow-matching paths for low- and high-frequency sub-states. In contrast to prior frequency-aware architectures or losses, FREPix aligns the transport path, prediction target, architecture, and objective with the coarse-to-fine structure of images.

3 Frequency-Decoupled Flow Matching

3.1 Frequency-Decomposed State Space and Heterogeneous Interpolation

Standard pixel-space flow matching methods (Li & He, 2025; Ma et al., 2026a) represent the sample at time t by a single state $x_t \in \mathbb{R}^d$ and, under the standard linear path, apply the same interpolation schedule to all image components through a shared vector field. While convenient, this homogeneous formulation does not explicitly reflect the frequency heterogeneity of natural images.

Frequency-decomposed state space. To make this heterogeneity explicit without sacrificing exactness, we reparameterize the image state with an orthonormal discrete wavelet transform (DWT) $\mathcal{W} : \mathbb{R}^d \to \mathbb{R}^d$. For any sample x_t, we write

$$(l_t, h_t) = \mathcal{W}(x_t), \qquad x_t = \mathcal{W}^{-1}(l_t, h_t), \tag{1}$$

where $l_t \in \mathbb{R}^{d_l}$ denotes the low-frequency sub-state *(structure)* and $h_t \in \mathbb{R}^{d_h}$ denotes the high-frequency sub-state *(detail)*, with $d_l + d_h = d$. Since $\mathcal{W}$ is orthonormal, the factorization is exact and preserves signal energy by Parseval's identity (Percival, 1995). Thus, unlike latent compression, this frequency factorization is lossless and changes only the parameterization of the state space.

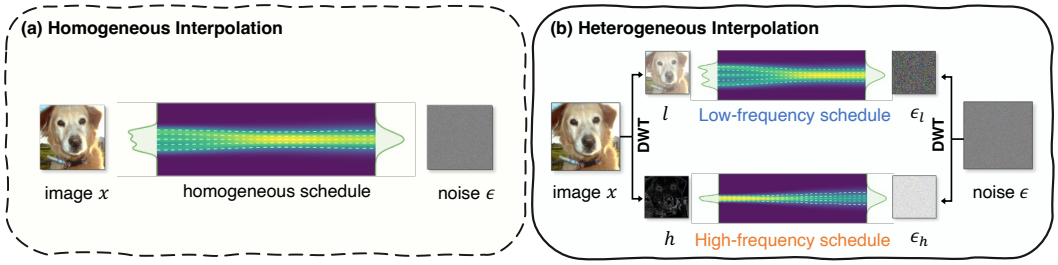


Figure 3: **Homogeneous vs. heterogeneous interpolation.** Standard pixel-space flow matching applies a shared interpolation schedule to all frequency components, treating the image as a homogeneous state during transport. In contrast, our method first decomposes the image into low- and high-frequency sub-states and then assigns them separate schedules $g_l(t)$ and $g_h(t)$.

From decomposed states to heterogeneous interpolation. Once the image state is explicitly decomposed into low- and high-frequency components, it is natural to decompose the interpolation path accordingly rather than transport all frequencies with a single shared schedule. Let $x \sim \rho_1$ be a clean image and $\epsilon \sim \rho_0 = \mathcal{N}(0, I_d)$ be a source noise, with $(l, h) = \mathcal{W}(x)$ and $(\epsilon_l, \epsilon_h) = \mathcal{W}(\epsilon)$. We define the heterogeneous interpolation path by

$$l_t = g_l(t)\, l + \big(1 - g_l(t)\big)\epsilon_l, \quad h_t = g_h(t)\, h + \big(1 - g_h(t)\big)\epsilon_h, \tag{2}$$

where $g_l, g_h \in C^1([0,1])$ are strictly increasing schedules that satisfy $g_l(0) = g_h(0) = 0$ and $g_l(1) = g_h(1) = 1$. This allows the two frequency sub-states to follow different transport dynamics.

For notational convenience, we further write the path in operator form as

$$\begin{aligned} x_t &= G(t)x + \big(I - G(t)\big)\epsilon, \quad \dot{x}_t = \dot{G}(t)(x - \epsilon) \\ G(t) &= \mathcal{W}^{-1} \begin{pmatrix} g_l(t) I_{d_l} & 0 \\ 0 & g_h(t) I_{d_h} \end{pmatrix} \mathcal{W}. \end{aligned} \tag{3}$$

where x_t denotes the pixel-space state and $\dot{x}_t$ is its time derivative, i.e., its conditional velocity. This formulation preserves linear operator interpolation between data and noise in pixel space, while generalizing the homogeneous scalar schedule of standard flow matching to a frequency-aware operator over explicitly decomposed sub-states. Fig. 3 further illustrates the difference.

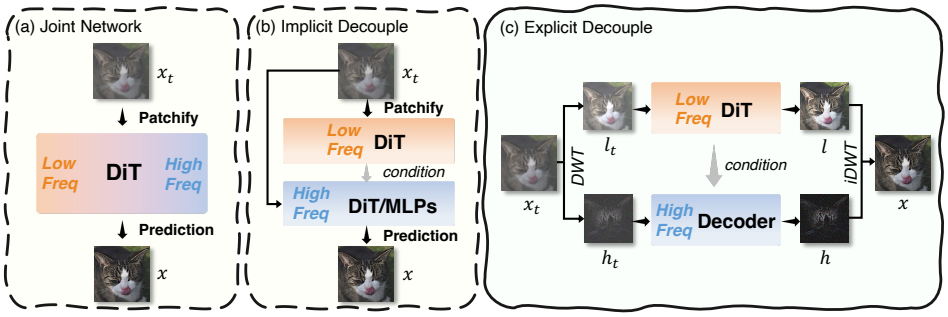


Figure 4: **Comparison of pixel-space methods.** (a) **Joint** (e.g., JiT) treats the image as a homogeneous state and predicts the clean target in one shot, leaving structure and detail entangled. (b) **Implicit decouple** (e.g., DeCo) introduces staged pathways that encourage specialization across scales, but does not explicitly assign prediction targets to different modules. (c) **Explicit decouple** (Ours) directly factorizes prediction responsibility with a structure predictor predicting the clean low-frequency component and a detail refiner predicting the clean high-frequency component.

Why use different schedules? Different schedules g_l and g_h let the transport process reflect the frequency heterogeneity of natural images. Under standard pixel-space flow matching, all frequency components follow the same schedule. Once the state space is decomposed, this homogeneous design becomes unnecessarily restrictive. We therefore adopt frequency-heterogeneous interpolation, allowing low- and high-frequency sub-states to evolve under different transport dynamics within a unified framework. We study the empirical instantiation of schedules in the experiments.

Proposition 1 (Validity of Heterogeneous Interpolation)**.** *Assume $\mathcal{W}$ is orthonormal, $x \sim \rho_1$ has finite second moment, $\epsilon \sim \mathcal{N}(0, I_d)$, and $g_l, g_h \in C^1([0,1])$ are strictly increasing with $g_l(0) = g_h(0) = 0$ and $g_l(1) = g_h(1) = 1$. Let x_t be defined by Eq. 3 and $\mathcal{B}$ denote the class of measurable vector fields $b(t,\cdot)$ such that $\int_0^1 \mathbb{E}\|b(t,x_t)\|^2\, dt < \infty$. Then:*

1. ***Smoothness:*** *The trajectory $t \mapsto x_t$ is almost surely continuously differentiable, with $\|\dot{x}_t\| \le L_g(\|x\| + \|\epsilon\|)$;*
2. ***Continuity Equation:*** *For every $t \in [0,1)$, the law of x_t admits a density p_t, and the marginal path satisfies the continuity equation $\partial_t p_t + \nabla \cdot (v_t p_t) = 0$ in the sense of distribution, where $v_t(x_t) = \mathbb{E}[\dot{x}_t | x_t]$ is the marginal velocity field;*
3. ***Learnability:*** *The population regression objective $\mathcal{L}(b) = \int_0^1 \mathbb{E}\big[\|b(t,x_t) - \dot{x}_t\|^2\big]\, dt$ is uniquely minimized (up to almost-everywhere equality) by the marginal velocity field $b^*(t,x_t) = \mathbb{E}[\dot{x}_t|x_t] = v_t(x_t)$.*

Proofs are in Appendix C. Proposition 1 shows that the heterogeneous interpolation is a principled extension of the standard flow matching path rather than a heuristic modification. This result is central to the remainder of our method: it justifies transport in the decomposed (l_t, h_t) state space and motivates the following network and the objective designs.

3.2 Factorized Generative Modeling via Explicit Architectural Decoupling

From decomposed transport to factorized generation. While we decompose the state space and transport path by frequency, the generator should preserve this structure rather than collapse it back into a unified prediction problem. Fig. 4 contrasts three architectural paradigms. In a joint design, one network operates on the mixed state x_t and predicts the clean target in one shot, leaving the separation between low-frequency structure and high-frequency detail entirely implicit. More recent designs, such as DeCo and PixelDiT, introduce staged pathways that can encourage specialization across scales. However, this specialization is defined primarily through architectural organization and feature routing, rather than by explicitly specifying which module should predict which frequency component.

To avoid collapsing the decomposed transport back and make the decomposition explicit at the prediction level, we design the generator to model the heterogeneous sub-state (l_t, h_t). Following JiT (Li & He, 2025), we adopt x-*prediction* parameterization. Specifically, we decouple the generation into two specialized modules: a *structure predictor* f_φ and a *detail refiner* g_ϕ:

$$\hat{l} = f_\varphi(l_t, t), \quad \hat{h} = g_\phi(h_t, \hat{l}, t), \quad \hat{x} = \mathcal{W}^{-1}(\hat{l}, \hat{h}). \tag{4}$$

The structure predictor f_φ is implemented as a Diffusion Transformer (DiT), which takes the noisy low-frequency sub-state l_t as input and predicts the clean low-frequency component $\hat{l}$, thereby capturing long-range dependencies and global structure. To enable efficient high-frequency modeling without the computational overhead of self-attention, the high-frequency predictor g_ϕ is implemented as the Decoder from DeCo. It takes the noisy high-frequency sub-state h_t as input to generate the clean high-frequency component $\hat{h}$ while using the predicted low-frequency structure $\hat{l}$ from the DiT as an explicit condition through AdaLN-Zero (Peebles & Xie, 2023).

Implicit vs. explicit decoupling. The key distinction of our architecture lies not only in modularization alone, but also in how the decomposition is specified. In implicitly decoupled designs, staged pathways can encourage different modules to specialize in different frequency roles, but this specialization remains emergent from the architecture. Our model instead makes the decomposition explicit at the prediction level by assigning different prediction targets to different modules and enforcing a low-frequency to high-frequency conditional dependency between them. This turns coarse-to-fine generation from an architectural tendency into a hard design principle, aligning the network itself with the decomposed transport.

To make this distinction more concrete, let the direct joint function class be $\mathcal{F}_{\text{dir}}$ and the explicit decoupled function class be $\mathcal{F}_{\text{dec}}$. We analyze a simplified statistical setting in which clean targets and predictions are bounded, the clean low-frequency component concentrates near a k_L-dimensional manifold, and the relevant loss classes admit covering-number growth in (Pollard, 1990). Proofs are in Appendix C.

Proposition 2 (Generalization comparison for explicit decoupling under simplified assumptions)**.** *Let the ambient dimension be $d := d_l + d_h$, and let $R_{\text{dir}}, \widehat{R}_{\text{dir}}$ and $R_{\text{dec}}, \widehat{R}_{\text{dec}}$ denote the corresponding true and empirical risks, respectively. The following bounds hold simultaneously for all $f \in \mathcal{F}_{\text{dir}}$ and $(g_l, g_h) \in \mathcal{F}_{\text{dec}}$ with probability at least $1 - \delta$:*

$$\begin{aligned} R_{\text{dir}}(f) &\le \widehat{R}_{\text{dir}}(f) + \frac{24A\sqrt{\pi d}}{\sqrt{N}} + 3\sqrt{\frac{2\log(8/\delta)}{N}}, \\ R_{\text{dec}}(g_L, g_H) &\le \widehat{R}_{\text{dec}}(g_l, g_h) + \frac{12A\sqrt{\pi}}{\sqrt{N}}\left(\sqrt{d_l} + \sqrt{k_l + d_h}\right) + 3\sqrt{\frac{2\log(8/\delta)}{N}}. \end{aligned} \tag{5}$$

where $k_l < d_l$ is the intrinsic dimension of the clean low-frequency component. Consequently, since $d = d_l + d_h$ and $k_l < d_l$, the decoupled complexity term is ***smaller*** *than the corresponding direct-model term.*

Proposition 2 should be interpreted as a comparison under a simplified statistical model rather than an exact characterization of modern Transformer-based architectures. Its role is to formalize the intuition that explicit decoupling can mitigate frequency entanglement by reducing the effective dimension and statistical complexity seen by each branch.

3.3 Frequency-Aligned Flow Matching Objective

With state, transport, and architecture all decomposed by frequency, the remaining question is how to align training with the same structure. In particular, the generator adopts an x-prediction parameterization, producing the clean reconstruction $x_\theta(x_t, t) = \text{net}_\theta(x_t, t)$ rather than regressing the velocity field directly. Following JiT, we preserve the optimization advantages of clean-data prediction, while recovering a flow matching training signal by analytically converting x_θ into the velocity v_θ induced by our heterogeneous path.

From x-prediction to induced velocity. Under the heterogeneous interpolation $x_t = G(t)x + \big(I - G(t)\big)\epsilon$, the conditional velocity induced by the path is obtained by differentiating with respect to t, which gives $v_t(x_t|x) = \dot{G}(t)(x - \epsilon)$. Further, we can rewrite v_t in terms of x_t which gives

$$v_t(x_t|x) = \dot{G}(t)\big(I - G(t)\big)^{-1}(x - x_t). \tag{6}$$

This makes it possible to convert clean-image prediction into velocity prediction. Specifically, by replacing the clean target x with its network prediction $\hat{x}$, we define the predicted velocity as

$$v_\theta(t, x_t) = \dot{G}(t)\big(I - G(t)\big)^{-1}(\hat{x} - x_t). \tag{7}$$

Frequency-aligned objective. Let $\mathcal{W}_l$ and $\mathcal{W}_h$ denote the low- and high-frequency projection operators of DWT where $(l, h) = (\mathcal{W}_l x, \mathcal{W}_h x) = \mathcal{W}(x)$ and $\mathcal{W}^{-1}(l, h) = \mathcal{W}_l^T l + \mathcal{W}_h^T h$, the conditional velocity induced by the heterogeneous interpolation can be naturally decomposed into low- and high-frequency components:

$$v_t(x_t|x) = \mathcal{W}_l^\top \dot{g}_l(t)(l - \epsilon_l) + \mathcal{W}_h^\top \dot{g}_h(t)(h - \epsilon_h). \tag{8}$$

This decomposition allows us to explicitly control the relative difficulty of low- and high-frequency learning during training. Let $\lambda_l(t), \lambda_h(t) > 0$ be time-dependent weights, we define the *frequency-aligned conditional flow matching objective* as

$$\mathcal{L}(\theta) := \mathbb{E}_{t,x,\epsilon}\Big[\lambda_l(t)\,\big\|\mathcal{W}_l\big(v_\theta(t, x_t) - v_t(x_t|x)\big)\big\|^2 + \lambda_h(t)\,\big\|\mathcal{W}_h\big(v_\theta(t, x_t) - v_t(x_t|x)\big)\big\|^2\Big]. \tag{9}$$

Frequency weighting preserves the target flow. The weights $\lambda_l(t)$ and $\lambda_h(t)$ allow us to rebalance optimization between the low- and high-frequency components over time. This gives a simple mechanism for rebalancing optimization across frequency components. We defer the discussion of specific weighting choices in the experiments. Importantly, this reweighting should improve training dynamics without changing the target flow field. To make this explicit, define the time-dependent weighting matrix

$$\mathbf{M}(t) := \lambda_l(t)\mathcal{W}_l^\top \mathcal{W}_l + \lambda_h(t)\mathcal{W}_h^\top \mathcal{W}_h. \tag{10}$$

Since $\mathcal{W}$ is orthonormal and $\lambda_l(t), \lambda_h(t) > 0$, $\mathbf{M}(t)$ is positive definite for all t, and the objective in Eq. 9 can be rewritten as

$$\mathcal{L}(\theta) = \mathbb{E}_{t,x}\Big[\Delta v_\theta(t, x_t)^\top \mathbf{M}(t)\Delta v_\theta(t, x_t)\Big]. \tag{11}$$

where $\Delta v_\theta(t, x_t) = v_\theta(t, x_t) - v_t(x_t|x)$.

Theorem 3 (Invariance of the Optimal Marginal Velocity under Frequency Weighting)**.** *Let $\lambda_l, \lambda_h \in C([0, 1])$ satisfy $\lambda_l(t), \lambda_h(t) > 0$ for all $t \in [0, 1]$, and let $\mathbf{M}(t)$ be defined by Eq. 10. Then the weighted objective in Eq. 11 admits*

$$\mathcal{L}(\theta) = \int_0^1 \mathbb{E}\Big[\Delta v_\theta(t, x_t)^\top \mathbf{M}(t)\Delta v_\theta(t, x_t)\Big]\, dt + C, \tag{12}$$

where $v_t(x_t) = \mathbb{E}[\dot{x}_t|x_t]$ is the marginal velocity, and C is a constant independent of θ. Consequently, the unique minimizer of $\mathcal{L}(\theta)$, up to almost-everywhere equality, is $v_\theta^*(t, x_t) = v_t(x_t)$.

Theorem 3 shows that $\lambda_l(t)$ and $\lambda_h(t)$ reweight the optimization geometry without changing the population-optimal marginal velocity field induced by our path. They can thus be used to rebalance learning dynamic across frequency and across time while preserving the same target flow.

Final training objective. Our primary objective is the frequency-aligned flow matching loss in Eq. 9. We further add a widely-used REPA loss (Yu et al., 2024) on intermediate features to representation alignment. In latent diffusion (Rombach et al., 2022) and pixel diffusion (Lu et al., 2026; Ma et al., 2026b), perceptual loss is widely used to improve VAE image reconstruction by supervising the decoded image $\hat{x}$. As we adopt x-prediction parameterization, the LPIPS perceptual loss (Zhang et al., 2018) is a natural auxiliary objective to encourage local pattern recovery. The final objective is $\mathcal{L}_{\text{total}} = \mathcal{L} + \mathcal{L}_{\text{REPA}} + \mathcal{L}_{\text{LPIPS}}$.

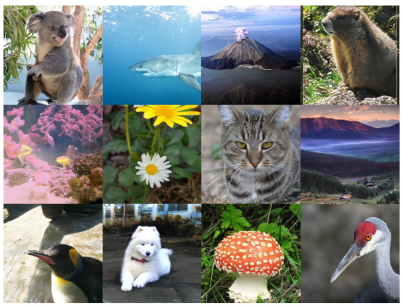


Figure 5: Qualitative results on ImageNet 256 × 256 using FREPix-XL.

Table 1: Baseline comparison results on ImageNet 256 after **200K** training steps. Inference uses 50 Euler steps without CFG. Text in gray: latent-space methods.

Method	Params	FID↓	IS↑	Pre.↑	Rec.↑
DiT-L/2	458M+86M	41.93	36.5	0.52	0.59
REPA-L/2	458M+86M	16.14	87.3	0.65	0.63
PixDDT	434M	46.37	36.2	0.45	0.63
PixNerd	458M	37.49	43.0	0.46	0.62
PixelFlow	459M	54.33	24.7	0.43	0.58
JiT-L	459M	23.25	67.7	0.55	0.65
DeCo	426M	31.35	48.4	0.51	0.65
FREPix	420M	13.85	105.6	0.67	0.54

4 Experiments

FREPix is evaluated through extensive class-to-image generation experiments on ImageNet at 256 and 512 resolutions. We report FID (gFID) (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016), Precision (Pre.) and Recall (Rec.) on 50K samples. For the frequency decomposition, we use a single-level orthonormal Haar DWT (Lepik & Hein, 2014). More details are in Appendix D.

4.1 Baseline Comparison

We further conduct a baseline comparison on ImageNet 256 under the same training budget. All models are trained for 200K steps and evaluated using 50 Euler steps without CFG. The compared methods include a joint pixel-space model, JiT, and several architectures that introduce coarse-to-fine or frequency-related inductive biases implicitly, including PixDDT (Wang et al., 2026), PixelFlow, PixNerd, and DeCo.

Table 1 shows that FREPix achieves the best FID, IS, and precision among all compared pixel-space baselines. In particular, FREPix obtains 13.85 FID, 105.6 IS, and 0.67 precision, substantially outperforming the joint JiT baseline, which achieves 23.25 FID, 67.7 IS, and 0.55 precision. Compared with DeCo, a strong frequency-aware baseline, FREPix improves FID by 17.50 and increases IS from 48.4 to 105.6. Similar advantages are observed over PixDDT, PixNerd, and PixelFlow. These results suggest that, under the same limited training budget, explicitly modeling with frequency-heterogeneity is more effective than relying on joint prediction or implicit architectural specialization. While FREPix has lower recall than some baselines, its stronger FID, IS, and precision indicate a clear improvement in sample fidelity and semantic quality.

4.2 Class-to-image Generation

The main experiments use FREPix-XL with 674M parameters. We sample with a 100-step Euler solver and CFG, train for 320 epochs on ImageNet 256, and further finetune for 10 epochs on ImageNet 512. Details are provided in Appendix D.

Table 2: Comparison results under early training steps of 80 epochs. For all models, we use the same settings.

Method	**FID↓**	**IS↑**	**Pre.↑**	**Rec.↑**
DeCo-XL/16	2.57	–	–	–
PixNerd-XL/16	2.68	257.3	0.78	0.56
PixelDiT	2.36	282.3	0.80	0.57
PixelGen	3.64	317.0	0.79	0.55
FREPix-XL	2.29	294.9	0.79	0.60

Table 3 reports the main class-conditional generation results. At 256 × 256, FREPix achieves competitive performance among recent pixel-space models, reaching strong FID, IS, precision, and recall with a relatively lightweight model and low computational cost. Notably, as shown in Table 2, FREPix-XL already performs well after only 80 training epochs, suggesting favorable optimization efficiency from the proposed frequency-factorized design. After training with 320 epochs, it further improves and remains competitive with strong pixel-space baselines, outperforming PixNerd and PixelFlow in FID while staying close to DeCo. This results highlight frequency heterogeneity as a useful design principle for pixel-space generation.

Table 3: Class-to-image generation on ImageNet 256×256 and 512×512 with CFG. Text in gray: latent-space methods. The Params. is reported with VAE.

	Method	Params	GFLOPs	Epochs	NFE	FID↓	IS↑	Pre.↑	Rec.↑
256×256	DiT-XL/2 (Peebles & Xie, 2023)	787M	238	1400	250×2	2.27	278.2	0.83	0.57
	REPA-XL/2 (Yu et al., 2024)	787M	238	800	250×2	1.42	305.7	0.80	0.64
	JetFormer (Tschannen et al., 2024)	2.8B	-	-	-	6.64	-	0.69	0.56
	FractalMAR-H (Li et al., 2025)	848M	-	600	-	6.15	348.9	0.81	0.46
	JiT-G/16 (Li & He, 2025)	2B	766	600	100×2	1.82	292.6	-	-
	PixelFlow-XL/4 (Chen et al., 2025)	677M	5818	320	120×2	1.98	282.1	0.81	0.60
	PixelDiT-XL (Yu et al., 2026)	797M	311	320	100×2	1.61	292.7	0.78	0.64
	DeCo-XL/16 (Ma et al., 2026a)	682M	237	320	100×2	1.90	303.0	0.80	0.61
	PixNerd-XL/16 (Wang et al., 2025)	700M	268	320	100×2	2.15	297.0	0.79	0.59
	FREPix-XL	674M	230	80	100×2	2.29	294.9	0.79	0.60
	FREPix-XL	674M	230	320	100×2	1.91	295.6	0.79	0.62
512×512	DiT-XL/2 (Peebles & Xie, 2023)	787M	1050	600	250×2	3.04	240.8	0.84	0.54
	SiT-XL/2 (Ma et al., 2024)	787M	1050	600	250×2	2.62	252.2	0.84	0.57
	ADM-G (Dhariwal & Nichol, 2021)	554M	3966	400	250	7.72	172.7	0.87	0.53
	VDM++ (Kingma & Gao, 2023)	2B	1110	800	250×2	2.65	278.1	-	-
	DeCo-XL/16 (Ma et al., 2026a)	682M	945	340	100×2	2.22	290.0	0.80	0.60
	PixelDiT-XL (Yu et al., 2026)	797M	1352	360	100×2	2.21	271.1	0.78	0.65
	JiT-H/32 (Li & He, 2025)	956M	-	600	100×2	1.94	309.1	-	-
	PixNerd-XL/16 (Wang et al., 2025)	700M	1042	340	100×2	2.84	245.6	0.80	0.59
	FREPix-XL	674M	922	330	100×2	2.38	334.7	0.80	0.59

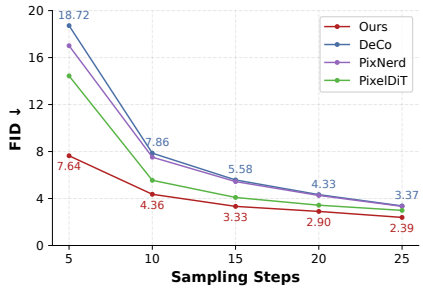


Figure 6: FID comparison in the low-NFE regime. We use the Euler sampler with CFG=3.0 from 5 to 25 sampling steps.

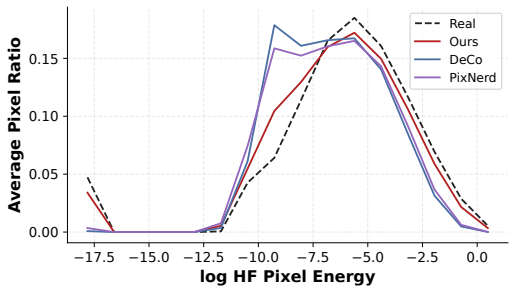


Figure 7: Visualization of the log-scale high-frequency pixel energy distributions for real images and generated samples.

At 512 × 512, FREPix-XL continues to show strong overall performance. Although its FID is not the best among all compared methods, it achieves the highest IS while maintaining competitive precision and recall. These results suggest that FREPix remains effective when scaling to higher resolution.

The advantage of FREPix becomes more pronounced under reduced computation. As shown in Fig. 6, FREPix-XL consistently achieves the lowest FID. This indicates that the frequency-heterogeneous design is particularly effective when limited computational resources. Together with its lower GFLOPs than most methods, these results suggest that FREPix provides a favorable trade-off between generation quality, sampling efficiency, and computational cost.

Furthermore, we visualize the log-scale pixel-energy distributions of the high-frequency components from ImageNet and generated samples in Fig. 7. Compared with baselines, FREPix is better aligned with the real distribution in the high-frequency subspaces, suggesting that FREPix more effectively captures the statistics of natural images.

4.3 Ablation Study

Ablation studies are conducted using FREPix-L at 256×256 resolution. For sampling, we take the Euler solver with 50 steps as the default choice **without** classifier-free guidance. The model is trained with 40 epochs (200K steps). More experimental details and results are provided in Appendix D.

Table 4: Ablation of power exponents γ_l and γ_h.

γ_l	γ_h	FID↓	IS↑	Pre.↑	Rec.↑
0.9	1.1	14.12	105.0	0.66	0.54
0.95	1.05	13.85	105.6	0.67	0.54
1.0	1.0	14.74	106.2	0.65	0.54
1.05	0.95	14.84	106.5	0.66	0.52
1.1	0.9	15.72	105.1	0.64	0.52

Table 5: Ablation of the strength ω.

ω	FID↓	IS↑	Pre.↑	Rec.↑
0	15.06	102.0	0.65	0.54
0.3	14.74	104.9	0.66	0.54
0.5	14.23	105.3	0.66	0.54
0.7	13.85	105.6	0.67	0.54
-0.7	15.49	99.7	0.65	0.54

Table 6: Ablation of REPA and LPIPS losses on FREPix-L.

Setting	FID↓	IS↑	Pre.↑	Rec.↑
Base	53.78	30.34	0.36	0.56
+LPIPS	25.70	65.25	0.57	0.53
+REPA	33.20	54.95	0.47	0.60
+REPA+LPIPS	13.85	105.6	0.67	0.54

Heterogeneous interpolation path. We instantiate the heterogeneous interpolation path with the low-frequency path slightly ahead of the high-frequency path, i.e., $g_l(t) > g_h(t)$ for $t \in (0, 1)$. Since larger $g(t)$ places the corresponding sub-state closer to its clean endpoint, this ordering exposes the model to cleaner structural information earlier, while leaving high-frequency details to be recovered later. The resulting trajectory is therefore explicitly coarse-to-fine: global structure approaches the data manifold before fine detail is fully formed. Concretely, we use smoothed power schedules

$$g_l(t) = \frac{(t+\varepsilon)^{\gamma_l} - \varepsilon^{\gamma_l}}{(1+\varepsilon)^{\gamma_l} - \varepsilon^{\gamma_l}}, \quad g_h(t) = \frac{(t+\varepsilon)^{\gamma_h} - \varepsilon^{\gamma_h}}{(1+\varepsilon)^{\gamma_h} - \varepsilon^{\gamma_h}}, \tag{13}$$

where $\gamma_l < \gamma_h$ and the offset $\varepsilon = 0.01$ is a small constant to regularize the derivatives near $t = 0$, while preserving the desired ordering between the two schedules.

Table 4 shows that placing the low-frequency path ahead is important. The heterogeneous path outperforms both the homogeneous schedule and the reversed ordering, with $(\gamma_l, \gamma_h) = (0.95, 1.05)$ giving the best overall FID–IS trade-off. In contrast, placing the high-frequency path ahead degrades performance. These results support our design principle that the interpolation path should reflect the asymmetric recovery process of natural images: structure should be established before high-frequency details are refined. Notably, we leave sophisticated heterogeneous path design to future work.

Frequency-aware reweighting. Motivated by the asymmetric recovery difficulty across frequency bands, we use a time-dependent cosine schedule for $\lambda_l(t)$ and $\lambda_h(t)$. Early in generation, low-frequency structure is easier and more reliable to recover, while high-frequency details become meaningful only when the sample approaches the data manifold. We therefore emphasize low-frequency errors at early times and high-frequency errors at later times:

$$\lambda_l(t) = 1 - \omega \cos\big(\pi(1-t)\big), \quad \lambda_h(t) = 1 + \omega \cos\big(\pi(1-t)\big), \tag{14}$$

where ω controls the reweighting strength. Larger ω yields a stronger asymmetry across frequency.

Table 5 shows that both the strength and direction are important. The proposed direction consistently outperforms its reversed counterpart, and $\omega = 0.7$ gives the best overall performance. These results suggest that effective supervision should reflect the time-varying recovery difficulty of low- and high-frequency components, rather than weighting them uniformly throughout the trajectory.

REPA and LPIPS loss. We further ablate the effects of the REPA and LPIPS losses in Table 6. Individually, both objectives substantially improve FID, IS, and precision over the base model. REPA additionally achieves the highest recall, indicating improved sample coverage, while LPIPS provides stronger gains in perceptual quality. Combining them yields the best overall performance, reducing FID from 53.78 to 13.85 and improving IS from 30.34 to 105.6, which suggests that representation alignment and perceptual supervision provide complementary training signals.

5 Conclusion

In this paper, we propose a frequency-heterogeneous flow matching framework for pixel-space image generation (FREPix). Our starting point is that natural images are inherently heterogeneous across frequencies, whereas existing pixel-space generation methods still largely formulate generation as a frequency-homogeneous process. FREPix makes this heterogeneity explicit throughout generation. Extensive experiments on ImageNet demonstrate that this formulation yields competitive performance among pixel-space generation models. We hope this work highlights frequency heterogeneity as a useful perspective for designing future pixel-space generative models.

AI USE STATEMENT

The authors determined the research questions, methodology, experimental design, analysis, and claims of this work. Generative AI tools were used to assist with manuscript preparation, including grammar checking, sentence-level editing, and polishing for clarity and readability. The authors take full responsibility for the final content of the paper, including its text, experiments, results, and conclusions.

ETHIC STATEMENT

Our work introduces a text-to-image generative models FREPix. At the same time, improved image generation can also increase risks associated with synthetic media. As with other generative image models, a stronger pixel-space generator may be misused to produce misleading, deceptive, or manipulative visual content. These concerns are not unique to our method, but advances in realism and controllability can make them more consequential in practice. Our work does not introduce mechanisms for safety, provenance, or misuse prevention, and we do not claim to address these broader challenges. We therefore believe that future progress in pixel-space generation should be accompanied by appropriate safeguards, including responsible deployment practices, provenance-aware tooling, and careful consideration of downstream use.

REPRODUCIBILITY STATEMENT

We provide details of the experimental setup, model configurations, evaluation metrics, and implementation procedures in the paper and appendix. We will release code and configuration files to facilitate reproduction of the main experiments and analyses.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

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

Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3435–3444, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Richard M Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

RM Dudley. Universal donsker classes and metric entropy. *The Annals of Probability*, 15(4): 1306–1326, 1987.

Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A fourier space perspective on diffusion models. *arXiv preprint arXiv:2505.11278*, 2025.

Philippe Hansen-Estruch, David Yan, Ching-Yao Chuang, Orr Zohar, Jialiang Wang, Tingbo Hou, Tao Xu, Sriram Vishwanath, Peter Vajda, and Xinlei Chen. Learnings from scaling visual tokenizers for reconstruction and generation. In *International Conference on Machine Learning*, pp. 22023–22043. PMLR, 2025.

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

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

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

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18062–18071, 2025.

Allan Jabri, David J Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *International Conference on Machine Learning*, pp. 14569–14589. PMLR, 2023.

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.

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

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025.

Ülo Lepik and Helle Hein. Haar wavelets. In *Haar wavelets: with applications*, pp. 7–20. Springer, 2014.

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

Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *Transactions on Machine Learning Research*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43600–43610, 2026a.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv preprint arXiv:2602.02493*, 2026b.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

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

Donald P Percival. On estimation of the wavelet variance. *Biometrika*, 82(3):619–631, 1995.

Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10199–10208, 2023.

David Pollard. *Empirical processes: theory and applications*. 1990.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

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

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

Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv preprint arXiv:2510.15301*, 2025.

Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: computation in neural systems*, 14(3):391, 2003.

Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. In *The Thirteenth International Conference on Learning Representations*, 2024.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40633–40642, 2026.

Zhi-Qin John Xu. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020.

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

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. In *The Thirteenth International Conference on Learning Representations*, 2024.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A Limitations

FREPix has several limitations. First, we only explore a limited family of heterogeneous interpolation schedules. While our results show that asymmetric low-/high-frequency transport is beneficial, the best schedule design remains underexplored, and richer or adaptive parameterizations may further improve performance. Second, FREPix is instantiated with a fixed orthonormal wavelet decomposition. This choice provides an exact and simple frequency factorization, but it is not the only way to expose heterogeneous image structure. More flexible multi-resolution or learned decompositions may better match the statistics of natural images and further improve the framework. Finally, our theoretical results are derived under simplified assumptions and are mainly intended to support the design intuition of explicit network decoupling, rather than to provide a complete characterization of modern large-scale architectures. We hope these limitations motivate future work on more flexible frequency decompositions, richer schedule designs, and broader empirical validation.

B Preliminaries

Flow-based generative models (Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2025) define sampling as simulating an ODE that pushes a prior distribution ρ_0 (typically $\mathcal{N}(0, I_d)$) forward to the data distribution ρ_1. During training, a noisy sample x_t is typically constructed using a simple linear interpolation path:

$$x_t = t\,x + (1-t)\,\epsilon, \qquad t \in [0,1], \tag{15}$$

where $x \sim \rho_1$ and $\epsilon \sim \rho_0$ denote the clean data and noise. Here, $t \in [0,1]$ dictates the generative trajectory from the initial noise state ($t=0$) to the clean data ($t=1$). This interpolation path induces the conditional velocity field $v_t(x_t \mid x) = x - \epsilon$. Conditional Flow Matching (CFM) (Lipman et al., 2023) learns a time-dependent network v_θ via L^2-regression against this target:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\, x\sim\rho_1,\, \epsilon\sim\rho_0}\Big[\big\|v_\theta(t,x_t) - v_t(x_t \mid x)\big\|^2\Big]. \tag{16}$$

Once trained, new samples are obtained by integrating the ODE

$$\frac{d}{dt}x_t = v_\theta(t,x_t), \qquad t \in [0,1], \tag{17}$$

starting from $t=0$ and ending at $t=1$. In practice, this ODE can be approximately solved using numerical solvers (e.g., Euler- and Heun-based solvers (Karras et al., 2022)).

C Proofs

C.1 Proofs of Proposition 1

Proof. Recall the heterogeneous interpolation path in Eq. 3

$$G(t) := \mathcal{W}^{-1}\begin{pmatrix} g_l(t)I_{d_l} & 0 \\ 0 & g_h(t)I_{d_h}\end{pmatrix}\mathcal{W}, \qquad x_t = G(t)x + (I - G(t))\epsilon. \tag{18}$$

By the orthonormality of $\mathcal{W}$ and the regularity of $g_l, g_h \in C^1([0,1])$, the matrix-valued map $t \mapsto G(t)$ is continuously differentiable. Define

$$L_g := \max\Big\{\sup_{t\in[0,1]}|\dot g_l(t)|,\ \sup_{t\in[0,1]}|\dot g_h(t)|\Big\} < \infty. \tag{19}$$

Since orthonormal changes of coordinates preserve operator norms,

$$\|\dot G(t)\|_{\mathrm{op}} = \max\big\{|\dot g_l(t)|, |\dot g_h(t)|\big\} \le L_g, \qquad \|G(t)\|_{\mathrm{op}} = \max\big\{|g_l(t)|, |g_h(t)|\big\} \le 1. \tag{20}$$

Step 1: Smoothness. For each realization of (x, ϵ), the path $t \mapsto x_t$ is C^1, with $\dot{x}_t = \dot{G}(t)(x - \epsilon)$, yielding

$$\|\dot{x}_t\| \le \|\dot{G}(t)\|_{\mathrm{op}} \, \|x - \epsilon\| \le L_g(\|x\| + \|\epsilon\|), \tag{21}$$

which establishes the claimed bound. Furthermore, given that x has finite second moment and $\epsilon \sim \mathcal{N}(0, I_d)$,

$$\int_0^1 \mathbb{E}\|\dot{x}_t\|^2 \, dt \le 2L_g^2\big(\mathbb{E}\|x\|^2 + \mathbb{E}\|\epsilon\|^2\big) < \infty. \tag{22}$$

Step 2: Density and continuity equation. Fix $t \in [0, 1)$. The strict monotonicity of g_l and g_h together with the boundary conditions $g_l(1) = g_h(1) = 1$ yields $0 \le g_l(t) < 1$ and $0 \le g_h(t) < 1$. Therefore,

$$I - G(t) = \mathcal{W}^{-1} \begin{pmatrix} (1 - g_l(t))I_{d_l} & 0 \\ 0 & (1 - g_h(t))I_{d_h} \end{pmatrix} \mathcal{W} \tag{23}$$

is invertible, and the conditional law of x_t given x is Gaussian:

$$x_t \mid x \sim \mathcal{N}\big(G(t)x, \, \Sigma_t\big), \qquad \Sigma_t := (I - G(t))(I - G(t))^\top. \tag{24}$$

By the positive definiteness of Σ_t for every $t < 1$, the law of x_t has the density

$$p_t(z) = \int_{\mathbb{R}^d} \phi_{\Sigma_t}\big(z - G(t)x\big) \, \rho_1(dx), \tag{25}$$

where ϕ_{Σ_t} denotes the Gaussian density with covariance Σ_t. In particular, p_t is well-defined for every $t \in [0, 1)$.

As $t \mapsto x_t$ is almost surely C^1, the chain rule gives

$$\frac{d}{dt}\varphi(x_t) = \nabla\varphi(x_t) \cdot \dot{x}_t. \tag{26}$$

Moreover,

$$\left|\frac{d}{dt}\varphi(x_t)\right| \le \|\nabla\varphi\|_\infty \, \|\dot{x}_t\|, \tag{27}$$

and the right-hand side is integrable by Step 1. Hence, by dominated convergence,

$$\frac{d}{dt}\mathbb{E}[\varphi(x_t)] = \mathbb{E}[\nabla\varphi(x_t) \cdot \dot{x}_t] \, . \tag{28}$$

Define the marginal velocity field $v_t(x_t) := \mathbb{E}[\dot{x}_t \mid x_t]$. Using the tower property,

$$\mathbb{E}[\nabla\varphi(x_t) \cdot \dot{x}_t] = \mathbb{E}[\nabla\varphi(x_t) \cdot v_t(x_t)] = \int_{\mathbb{R}^d} \nabla\varphi(z) \cdot v_t(z) \, p_t(z) \, dz. \tag{29}$$

Direct computation yields

$$\mathbb{E}[\varphi(x_t)] = \int_{\mathbb{R}^d} \varphi(z) \, p_t(z) \, dz. \tag{30}$$

Therefore,

$$\frac{d}{dt}\int_{\mathbb{R}^d} \varphi(z) \, p_t(z) \, dz = \int_{\mathbb{R}^d} \nabla\varphi(z) \cdot v_t(z) \, p_t(z) \, dz = -\int_{\mathbb{R}^d} \varphi(z) \, \nabla\cdot\big(v_t(z)p_t(z)\big) \, dz. \tag{31}$$

As this identity holds for every $\varphi \in C_c^\infty(\mathbb{R}^d)$, it follows that

$$\partial_t p_t + \nabla \cdot (v_t p_t) = 0 \tag{32}$$

in the sense of distributions on $\mathbb{R}^d$.

Step 3: Learnability. Let

$$\mathcal{B} = \Big\{ b : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d \text{ measurable } : \int_0^1 \mathbb{E}\|b(t,x_t)\|^2 \, dt < \infty \Big\}. \tag{33}$$

By Jensen's inequality for conditional expectations and Step 1,

$$\int_0^1 \mathbb{E}\|v_t(x_t)\|^2 \, dt = \int_0^1 \mathbb{E}\big\|\mathbb{E}[\dot{x}_t \mid x_t]\big\|^2 \, dt \le \int_0^1 \mathbb{E}\|\dot{x}_t\|^2 \, dt < \infty, \tag{34}$$

which implies $v \in \mathcal{B}$.

Now fix any $b \in \mathcal{B}$. Expansion of the squared norm yields

$$\|b(t,x_t) - \dot{x}_t\|^2 = \|b(t,x_t) - v_t(x_t)\|^2 + \|v_t(x_t) - \dot{x}_t\|^2 + 2\langle b(t,x_t) - v_t(x_t),\, v_t(x_t) - \dot{x}_t \rangle. \tag{35}$$

Taking expectations, the cross term vanishes:

$$\begin{aligned} &\mathbb{E}[\langle b(t,x_t) - v_t(x_t),\, v_t(x_t) - \dot{x}_t \rangle] \\ &\quad = \mathbb{E}[\mathbb{E}[\langle b(t,x_t) - v_t(x_t),\, v_t(x_t) - \dot{x}_t \rangle \mid x_t]] \\ &\quad = \mathbb{E}[\langle b(t,x_t) - v_t(x_t),\, \mathbb{E}[v_t(x_t) - \dot{x}_t \mid x_t] \rangle] = 0, \end{aligned} \tag{36}$$

since $b(t,x_t) - v_t(x_t)$ is $\sigma(x_t)$-measurable and $\mathbb{E}[v_t(x_t) - \dot{x}_t \mid x_t] = v_t(x_t) - \mathbb{E}[\dot{x}_t \mid x_t] = 0$.

Integrating over t yields the orthogonal decomposition

$$\mathcal{L}(b) = \mathcal{L}(v) + \int_0^1 \mathbb{E}\|b(t,x_t) - v_t(x_t)\|^2 \, dt. \tag{37}$$

Hence $\mathcal{L}(b) \ge \mathcal{L}(v)$ for every $b \in \mathcal{B}$, with equality if and only if

$$b(t,x_t) = v_t(x_t) \qquad \text{for Lebesgue-a.e. } t \in [0,1] \text{ and } p_t\text{-a.e. } x_t \in \mathbb{R}^d. \tag{38}$$

Therefore, the population regression objective is uniquely minimized, up to almost-everywhere equality, by

$$b^*(t,x_t) = v_t(x_t) = \mathbb{E}[\dot{x}_t \mid x_t]. \tag{39}$$

This proves the proposition. □

C.2 Proofs of Proposition 2

Let $\mathcal{W} : \mathbb{R}^d \to \mathbb{R}^{d_L} \times \mathbb{R}^{d_H}$ be an orthonormal discrete wavelet transform, where $d = d_L + d_H$. For any sample (x_t, x), write its wavelet decomposition as

$$\mathcal{W}(x_t) = (l_t, h_t), \qquad \mathcal{W}(x) = (l, h). \tag{40}$$

Given N i.i.d. samples $S = \{(l_t^{(i)}, h_t^{(i)}, l^{(i)}, h^{(i)})\}_{i=1}^N$, we compare direct modeling with explicit decoupling under a simplified analysis, and then extend the result to the practical architecture.

Definition 1 (Function classes)**.** *The direct modeling class takes the full noisy state $(l_t, h_t) \in \mathbb{R}^d$ and jointly predicts the clean low- and high-frequency components:*

$$\mathcal{F}_{\text{dir}} := \mathcal{F}_{\text{dir}}^L \times \mathcal{F}_{\text{dir}}^H, \qquad \mathcal{F}_{\text{dir}}^L \subset \{f_L : \mathbb{R}^d \to \mathbb{R}^{d_L}\}, \qquad \mathcal{F}_{\text{dir}}^H \subset \{f_H : \mathbb{R}^d \to \mathbb{R}^{d_H}\}. \tag{41}$$

The decoupled function class separates the prediction responsibilities: the low-frequency branch only takes l_t, while the high-frequency branch is analyzed under teacher forcing and takes (l, h_t):

$$\mathcal{F}_{\text{dec}} := \mathcal{F}_{\text{dec}}^L \times \mathcal{F}_{\text{dec}}^H, \qquad \mathcal{F}_{\text{dec}}^L \subset \{g_L : \mathbb{R}^{d_L} \to \mathbb{R}^{d_L}\}, \qquad \mathcal{F}_{\text{dec}}^H \subset \{g_H : \mathbb{R}^{d_L} \times \mathbb{R}^{d_H} \to \mathbb{R}^{d_H}\}. \tag{42}$$

The practical function class feeds the predicted low-frequency component into the high-frequency branch:

$$\mathcal{F}_{\text{real}} := \Big\{ (g_L, g_H) : g_L \in \mathcal{F}_{\text{dec}}^L,\ g_H \in \mathcal{F}_{\text{dec}}^H \Big\}, \tag{43}$$

with prediction rule $\hat{l} = g_L(l_t), \hat{h} = g_H(\hat{l}, h_t)$.

Assumption 1 (Boundedness). *There exists $B > 0$ such that almost surely $\|l\|_\infty,\ \|h\|_\infty \le B$, and all candidate predictors satisfy $\|\hat{l}\|_\infty,\ \|\hat{h}\|_\infty \le B$. Define the normalized squared losses*

$$\ell_L(\hat{l}, l) := \frac{\|\hat{l} - l\|_2^2}{4B^2 d_L}, \qquad \ell_H(\hat{h}, h) := \frac{\|\hat{h} - h\|_2^2}{4B^2 d_H}. \tag{44}$$

Then $0 \le \ell_L, \ell_H \le 1$.

Assumption 2 (Low-dimensional structural manifold). *The clean low-frequency component l is concentrated near a low-dimensional manifold*

$$\mathcal{M}_L \subset \mathbb{R}^{d_L} \tag{45}$$

with intrinsic dimension $k_L < d_L$.

Assumption 3 (Covering-number growth under a finite-dimensional proxy analysis). *Let the loss classes induced by the normalized squared losses be*

$$\begin{aligned} \mathcal{G}^L_{\text{dir}} &:= \{z \mapsto \ell_L(f_L(l_t, h_t), l) : f_L \in \mathcal{F}^L_{\text{dir}}\}, \quad \mathcal{G}^H_{\text{dir}} := \{z \mapsto \ell_H(f_H(l_t, h_t), h) : f_H \in \mathcal{F}^H_{\text{dir}}\}, \\ \mathcal{G}^L_{\text{dec}} &:= \{z \mapsto \ell_L(g_L(l_t), l) : g_L \in \mathcal{F}^L_{\text{dec}}\}, \qquad \mathcal{G}^H_{\text{dec}} := \{z \mapsto \ell_H(g_H(l, h_t), h) : g_H \in \mathcal{F}^H_{\text{dec}}\}, \end{aligned} \tag{46}$$

where $z = (l_t, h_t, l, h)$.

We assume a simplified finite-dimensional proxy analysis in which each loss class $\mathcal{G}$ admits an effective parameterization of dimension $m_{\mathcal{G}}$ over a bounded parameter set, and the induced loss is uniformly Lipschitz with respect to that parameterization. Under this proxy, the metric entropy satisfies the Pollard-type growth condition (Pollard, 1990): there exists a constant $A \ge 1$ such that, for every $\varepsilon \in (0, 1]$ and every $\mathcal{G} \in \{\mathcal{G}^L_{\text{dir}}, \mathcal{G}^H_{\text{dir}}, \mathcal{G}^L_{\text{dec}}, \mathcal{G}^H_{\text{dec}}\}$,

$$\log \mathcal{N}(\varepsilon, \mathcal{G}, L_2(P_N)) \le m_{\mathcal{G}} \log(A/\varepsilon). \tag{47}$$

In the simplified linear proxy considered here, the effective dimensions scale with the corresponding input degrees of freedom:

$$m_{\mathcal{G}^L_{\text{dir}}} = m_{\mathcal{G}^H_{\text{dir}}} = d, \qquad m_{\mathcal{G}^L_{\text{dec}}} = d_L, \qquad m_{\mathcal{G}^H_{\text{dec}}} = k_L + d_H, \tag{48}$$

where the last relation reflects that the high-frequency branch is conditioned on (l, h_t) and the clean low-frequency component l is assumed to lie near a k_L-dimensional structural manifold.

Assumption 4 (Conditional Lipschitz property). *There exists $L_{\text{cond}} > 0$ such that for every $z, z' \in \mathbb{R}^{d_L}$ and every $h_t \in \mathbb{R}^{d_H}$,*

$$\|g_H(z, h_t) - g_H(z', h_t)\|_2 \le L_{\text{cond}} \|z - z'\|_2, \qquad \forall g_H \in \mathcal{F}^H_{\text{dec}}.$$

This assumption quantifies the error propagation induced by replacing the clean low-frequency input l with its prediction $\hat{l} = g_L(l_t)$ in the practical architecture.

C.2.1 From covering numbers to Rademacher complexity

Lemma 4 (Entropy integral bound). *Let $\mathcal{G} \subset [0, 1]^{\mathcal{X}}$ be a function class satisfying*

$$\log \mathcal{N}(\varepsilon, \mathcal{G}, L_2(P_N)) \le m \log(A/\varepsilon), \qquad \forall \varepsilon \in (0, 1], \tag{49}$$

for a certain constant $A \ge 1$. Then the empirical Rademacher complexity of $\mathcal{G}$ on the sample S satisfies

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \le \frac{6A\sqrt{\pi m}}{\sqrt{N}}. \tag{50}$$

Proof. By Dudley's entropy integral bound (Dudley, 1967; 1987),

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \le \inf_{\alpha > 0} \left[4\alpha + \frac{12}{\sqrt{N}} \int_\alpha^1 \sqrt{\log \mathcal{N}(\varepsilon, \mathcal{G}, L_2(P_N))}\, d\varepsilon \right]. \tag{51}$$

Letting $\alpha \downarrow 0$ yields

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \le \frac{12}{\sqrt{N}} \int_0^1 \sqrt{\log \mathcal{N}(\varepsilon, \mathcal{G}, L_2(P_N))}\, d\varepsilon. \tag{52}$$

Substituting the covering-number assumption gives

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \le \frac{12\sqrt{m}}{\sqrt{N}} \int_0^1 \sqrt{\log(A/\varepsilon)}\, d\varepsilon. \tag{53}$$

By the change of variables $u = \log(A/\varepsilon)$, which yields $\varepsilon = Ae^{-u}$ and $d\varepsilon = -Ae^{-u}du$, we obtain

$$\begin{aligned} \int_0^1 \sqrt{\log(A/\varepsilon)}\, d\varepsilon &= A \int_{\log A}^{\infty} \sqrt{u}\, e^{-u}\, du \\ &\le A \int_0^{\infty} \sqrt{u}\, e^{-u}\, du = A\,\Gamma\left(\frac{3}{2}\right) = \frac{A\sqrt{\pi}}{2}. \end{aligned} \tag{54}$$

Therefore,

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \le \frac{12\sqrt{m}}{\sqrt{N}} \cdot \frac{A\sqrt{\pi}}{2} = \frac{6A\sqrt{\pi m}}{\sqrt{N}}. \tag{55}$$

□

C.2.2 Risks and generalization comparison

Definition 2 (Risks). *Define the branch-wise risks*

$$\begin{aligned} R^L_{\mathrm{dir}}(f_L) &:= \mathbb{E}\big[\ell_L(f_L(l_t, h_t), l)\big], \quad & R^H_{\mathrm{dir}}(f_H) &:= \mathbb{E}\big[\ell_H(f_H(l_t, h_t), h)\big], \\ R^L_{\mathrm{dec}}(g_L) &:= \mathbb{E}\big[\ell_L(g_L(l_t), l)\big], & R^H_{\mathrm{dec}}(g_H) &:= \mathbb{E}\big[\ell_H(g_H(l, h_t), h)\big]. \end{aligned} \tag{56}$$

The total risks are

$$R_{\mathrm{dir}}(f) := R^L_{\mathrm{dir}}(f_L) + R^H_{\mathrm{dir}}(f_H), \qquad R_{\mathrm{dec}}(g_L, g_H) := R^L_{\mathrm{dec}}(g_L) + R^H_{\mathrm{dec}}(g_H). \tag{57}$$

Their empirical counterparts, denoted by $\widehat{R}_{\mathrm{dir}}$ and $\widehat{R}_{\mathrm{dec}}$, are defined analogously.

Proposition 5 (Generalization comparison for explicit decoupling under simplified assumptions). *Let the ambient dimension be $d := d_L + d_H$, and let $R_{\mathrm{dir}}, \widehat{R}_{\mathrm{dir}}$ and $R_{\mathrm{dec}}, \widehat{R}_{\mathrm{dec}}$ denote the corresponding true and empirical risks, respectively (see Definition 2). The following bounds hold simultaneously for all $f \in \mathcal{F}_{\mathrm{dir}}$ and $(g_L, g_H) \in \mathcal{F}_{\mathrm{dec}}$ with probability at least $1 - \delta$:*

$$R_{\mathrm{dir}}(f) \le \widehat{R}_{\mathrm{dir}}(f) + \frac{24A\sqrt{\pi d}}{\sqrt{N}} + 3\sqrt{\frac{2\log(8/\delta)}{N}}, \tag{58}$$

$$R_{\mathrm{dec}}(g_L, g_H) \le \widehat{R}_{\mathrm{dec}}(g_L, g_H) + \frac{12A\sqrt{\pi}}{\sqrt{N}}\big(\sqrt{d_L} + \sqrt{k_L + d_H}\big) + 3\sqrt{\frac{2\log(8/\delta)}{N}}, \tag{59}$$

where $k_L < d_L$ is the intrinsic dimension of the clean low-frequency component. Consequently, since $d = d_L + d_H$ and $k_L < d_L$, the decoupled complexity term is smaller than the corresponding direct-model term.

Proof. We first bound the Rademacher complexities of the four loss classes. By Assumption 3 and Lemma 4,

$$\begin{aligned} \widehat{\mathfrak{R}}_S(\mathcal{G}^L_{\mathrm{dir}}) &\le \frac{6A\sqrt{\pi d}}{\sqrt{N}}, \qquad & \widehat{\mathfrak{R}}_S(\mathcal{G}^H_{\mathrm{dir}}) &\le \frac{6A\sqrt{\pi d}}{\sqrt{N}}, \\ \widehat{\mathfrak{R}}_S(\mathcal{G}^L_{\mathrm{dec}}) &\le \frac{6A\sqrt{\pi d_L}}{\sqrt{N}}, & \widehat{\mathfrak{R}}_S(\mathcal{G}^H_{\mathrm{dec}}) &\le \frac{6A\sqrt{\pi(k_L + d_H)}}{\sqrt{N}}. \end{aligned} \tag{60}$$

In view of Assumption 1, all losses are $[0, 1]$-valued; thus, the standard uniform Rademacher generalization bound (Mohri et al., 2018) implies that for any class $\mathcal{G}$ of $[0, 1]$-valued losses, with probability at least $1 - \eta$,

$$R(g) \le \widehat{R}(g) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log(2/\eta)}{2N}} \qquad \forall g \in \mathcal{G}. \tag{61}$$

Direct model. Applying the above bound to $\mathcal{G}_{\mathrm{dir}}^{L}$ and $\mathcal{G}_{\mathrm{dir}}^{H}$ with confidence level $\eta = \delta/4$ for each class, a union bound yields that, with probability at least $1-\delta/2$, both inequalities hold simultaneously for all $f = (f_L, f_H) \in \mathcal{F}_{\mathrm{dir}}$:

$$\begin{aligned} R_{\mathrm{dir}}^{L}(f_L) &\le \widehat{R}_{\mathrm{dir}}^{L}(f_L) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}_{\mathrm{dir}}^{L}) + 3\sqrt{\frac{\log(8/\delta)}{2N}}, \\ R_{\mathrm{dir}}^{H}(f_H) &\le \widehat{R}_{\mathrm{dir}}^{H}(f_H) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}_{\mathrm{dir}}^{H}) + 3\sqrt{\frac{\log(8/\delta)}{2N}}. \end{aligned} \tag{62}$$

Using Eq. 60 and summing the two branch-wise bounds yields

$$R_{\mathrm{dir}}(f) \le \widehat{R}_{\mathrm{dir}}(f) + \frac{24A\sqrt{\pi d}}{\sqrt{N}} + 3\sqrt{\frac{2\log(8/\delta)}{N}}. \tag{63}$$

Decoupled model. Applying the same argument to $\mathcal{G}_{\mathrm{dec}}^{L}$ and $\mathcal{G}_{\mathrm{dec}}^{H}$, again with confidence level $\eta = \delta/4$ for each class, a union bound yields that, with probability at least $1-\delta/2$, the following hold simultaneously for all $(g_L, g_H) \in \mathcal{F}_{\mathrm{dec}}$:

$$\begin{aligned} R_{\mathrm{dec}}^{L}(g_L) &\le \widehat{R}_{\mathrm{dec}}^{L}(g_L) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}_{\mathrm{dec}}^{L}) + 3\sqrt{\frac{\log(8/\delta)}{2N}}, \\ R_{\mathrm{dec}}^{H}(g_H) &\le \widehat{R}_{\mathrm{dec}}^{H}(g_H) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}_{\mathrm{dec}}^{H}) + 3\sqrt{\frac{\log(8/\delta)}{2N}}. \end{aligned} \tag{64}$$

Using Eq. 60 and summing gives

$$R_{\mathrm{dec}}(g_L, g_H) \le \widehat{R}_{\mathrm{dec}}(g_L, g_H) + \frac{12A\sqrt{\pi}}{\sqrt{N}}\big(\sqrt{d_L} + \sqrt{k_L + d_H}\big) + 3\sqrt{\frac{2\log(8/\delta)}{N}}. \tag{65}$$

Complexity comparison. Each of the two events above occurs with probability at least $1-\delta/2$. A final union bound implies that both inequalities hold simultaneously with probability at least $1-\delta$. From $k_L < d_L$, it follows that

$$\sqrt{d_L} + \sqrt{k_L + d_H} < \sqrt{d_L} + \sqrt{d_L + d_H} < 2\sqrt{d_L + d_H} = 2\sqrt{d}. \tag{66}$$

Multiplying both sides by $12A\sqrt{\pi}/\sqrt{N}$ establishes that the decoupled complexity term is strictly smaller than the corresponding direct-model term. □

C.2.3 Error propagation and the practical architecture

Lemma 6 (Conditional error propagation)**.** *Under Assumptions 1 and 4, for any $g_H \in \mathcal{F}_{\mathrm{dec}}^{H}$ and any $(l, \hat{l}, h_t, h)$, one has*

$$\ell_H\big(g_H(\hat{l}, h_t), h\big) \le \ell_H\big(g_H(l, h_t), h\big) + \frac{L_{\mathrm{cond}}}{B\sqrt{d_H}}\|\hat{l} - l\|_2. \tag{67}$$

Proof. Fix $u := g_H(\hat{l}, h_t)$ and $v := g_H(l, h_t)$. By the definition of the normalized high-frequency loss,

$$\ell_H(u, h) - \ell_H(v, h) = \frac{1}{4B^2 d_H}\Big(\|u - h\|_2^2 - \|v - h\|_2^2\Big). \tag{68}$$

Using the identity $\|a\|_2^2 - \|b\|_2^2 = (a-b)^\top(a+b)$ with $a = u - h$ and $b = v - h$, we obtain

$$\ell_H(u, h) - \ell_H(v, h) = \frac{1}{4B^2 d_H}(u - v)^\top(u + v - 2h). \tag{69}$$

Taking absolute values and applying the Cauchy–Schwarz inequality yields

$$\big|\ell_H(u, h) - \ell_H(v, h)\big| \le \frac{1}{4B^2 d_H}\|u - v\|_2\,\|u + v - 2h\|_2. \tag{70}$$

By Assumption 1, $\|u\|_\infty,\ \|v\|_\infty,\ \|h\|_\infty \le B$, so each coordinate of $u+v-2h$ has magnitude at most $4B$, which implies $\|u+v-2h\|_2 \le 4B\sqrt{d_H}$.

Moreover, by Assumption 4,

$$\|u-v\|_2 = \|g_H(\hat{l},h_t) - g_H(l,h_t)\|_2 \le L_{\text{cond}}\|\hat{l}-l\|_2. \tag{71}$$

Combining the two estimates yields

$$\left|\ell_H(u,h) - \ell_H(v,h)\right| \le \frac{L_{\text{cond}}}{B\sqrt{d_H}}\|\hat{l}-l\|_2. \tag{72}$$

The desired one-sided inequality follows immediately. □

Corollary 7 (Generalization bound for the practical decoupled model)**.** *Fix any low-frequency predictor $g_L \in \mathcal{F}^L_{\text{dec}}$ independent of the randomness of the current sample, and let $(g_L, g_H) \in \mathcal{F}_{\text{real}}$. Define the practical risk by $R_{\text{real}}(g_L,g_H) := \mathbb{E}\big[\ell_L\big(g_L(l_t),l\big) + \ell_H\big(g_H(g_L(l_t),h_t),h\big)\big]$. Then, under Assumptions 1–4, with probability at least $1-\delta$, the following holds simultaneously for all $g_H \in \mathcal{F}^H_{\text{dec}}$:*

$$\begin{aligned} R_{\text{real}}(g_L,g_H) \le \widehat{R}_{\text{dec}}(g_L,g_H) &+ \frac{12A\sqrt{\pi}}{\sqrt{N}}\big(\sqrt{d_L}+\sqrt{k_L+d_H}\big) \\ &+ \frac{L_{\text{cond}}}{B\sqrt{d_H}}\mathbb{E}\|g_L(l_t)-l\|_2 + 3\sqrt{\frac{2\log(8/\delta)}{N}}. \end{aligned} \tag{73}$$

Proof. Define the practical high-frequency risk

$$R^H_{\text{real}}(g_L,g_H) := \mathbb{E}\big[\ell_H\big(g_H(g_L(l_t),h_t),h\big)\big]. \tag{74}$$

By Lemma 6, applied pointwise with $\hat{l} = g_L(l_t)$ and then averaged over the data distribution,

$$R^H_{\text{real}}(g_L,g_H) \le R^H_{\text{dec}}(g_H) + \frac{L_{\text{cond}}}{B\sqrt{d_H}}\mathbb{E}\|g_L(l_t)-l\|_2. \tag{75}$$

Therefore,

$$\begin{aligned} R_{\text{real}}(g_L,g_H) &= R^L_{\text{dec}}(g_L) + R^H_{\text{real}}(g_L,g_H) \\ &\le R^L_{\text{dec}}(g_L) + R^H_{\text{dec}}(g_H) + \frac{L_{\text{cond}}}{B\sqrt{d_H}}\mathbb{E}\|g_L(l_t)-l\|_2 \\ &= R_{\text{dec}}(g_L,g_H) + \frac{L_{\text{cond}}}{B\sqrt{d_H}}\mathbb{E}\|g_L(l_t)-l\|_2. \end{aligned} \tag{76}$$

Substituting the bound for $R_{\text{dec}}(g_L,g_H)$ from Proposition 2 completes the proof. □

Remark 1. *In the practical architecture, the high-frequency predictor conditions on the predicted $\hat{l}$ rather than the ground-truth l. Corollary 7 establishes that this modification introduces only an additional term controlled by the low-frequency prediction error. Consequently, the complexity advantage of explicit decoupling is retained provided that the structure predictor is sufficiently accurate.*

C.3 Proofs of Theorem 3

Proof. We prove the two claims in turn.

Recall from Eq. 10 that

$$\mathbf{M}(t) = \lambda_l(t)\,\mathcal{W}_l^\top\mathcal{W}_l + \lambda_h(t)\,\mathcal{W}_h^\top\mathcal{W}_h.$$

By the orthonormality of $\mathcal{W}$, we have $\mathcal{W}_l^\top\mathcal{W}_l + \mathcal{W}_h^\top\mathcal{W}_h = I_d$, which implies that for any nonzero $a \in \mathbb{R}^d$,

$$a^\top\mathbf{M}(t)a = \lambda_l(t)\|\mathcal{W}_l a\|_2^2 + \lambda_h(t)\|\mathcal{W}_h a\|_2^2 \ge \min\{\lambda_l(t),\lambda_h(t)\}\,\|a\|_2^2 > 0, \tag{77}$$

where the equality uses $\|\mathcal{W}_l a\|_2^2 + \|\mathcal{W}_h a\|_2^2 = \|a\|_2^2$ and the last inequality follows from $\lambda_l(t),\lambda_h(t) > 0$. Thus $\mathbf{M}(t)$ is symmetric positive definite for every $t \in [0,1]$.

Consider the weighted objective in Eq. 11. For fixed t, ϵ, and z, define $\delta := v_\theta(t,z) - v_t(z \mid x)$. Since $\mathbf{M}(t)$ is symmetric, expanding the quadratic form and collecting the x-independent term into C_1 gives

$$\begin{aligned}\mathcal{L}_{\mathrm{FA}}(\theta) &= \int_0^1 \mathbb{E}_{t,x,\epsilon}\Big[v_\theta(t,x_t)^\top \mathbf{M}(t) v_\theta(t,x_t) - 2\, v_\theta(t,x_t)^\top \mathbf{M}(t) v_t(x_t \mid x) \\ &\qquad + v_t(x_t \mid x)^\top \mathbf{M}(t) v_t(x_t \mid x)\Big] dt \\ &= \int_0^1 \mathbb{E}_{t,x,\epsilon}\Big[v_\theta(t,x_t)^\top \mathbf{M}(t) v_\theta(t,x_t) - 2\, v_\theta(t,x_t)^\top \mathbf{M}(t) v_t(x_t \mid x)\Big] dt + C_1,\end{aligned} \tag{78}$$

where $C_1 := \int_0^1 \mathbb{E}_{t,x,\epsilon}\big[v_t(x_t \mid x)^\top \mathbf{M}(t) v_t(x_t \mid x)\big] dt$ is independent of θ.

We now simplify the cross term. For each fixed t, expanding the expectation and using the fact that neither $v_\theta(t,z)$ nor $\mathbf{M}(t)$ depends on the conditioning variable x yields

$$\begin{aligned}&\mathbb{E}_{t,x,\epsilon}\big[v_\theta(t,x_t)^\top \mathbf{M}(t) v_t(x_t \mid x)\big] \\ &\quad = \int_{\mathbb{R}^d} v_\theta(t,z)^\top \mathbf{M}(t) \Big(\int_{\mathbb{R}^d} v_t(z \mid x)\, p_t(z \mid x)\, \rho_1(x)\, dx \Big) dz \\ &\quad = \int_{\mathbb{R}^d} v_\theta(t,z)^\top \mathbf{M}(t)\, v_t(z)\, p_t(z)\, dz = \mathbb{E}_{z\sim p_t}\big[v_\theta(t,z)^\top \mathbf{M}(t) v_t(z)\big],\end{aligned} \tag{79}$$

where the second equality follows from the definition of the marginal velocity field $v_t(z) = \int v_t(z \mid x) p_t(z \mid x) \rho_1(x)\, dx$.

Substituting Eq. 79 into Eq. 78 and completing the square yields

$$\begin{aligned}\mathcal{L}_{\mathrm{FA}}(\theta) &= \int_0^1 \mathbb{E}_{z\sim p_t}\Big[\big(v_\theta(t,z) - v_t(z)\big)^\top \mathbf{M}(t)\big(v_\theta(t,z) - v_t(z)\big) - v_t(z)^\top \mathbf{M}(t) v_t(z)\Big] dt + C_1 \\ &= \int_0^1 \mathbb{E}_{z\sim p_t}\Big[\big(v_\theta(t,z) - v_t(z)\big)^\top \mathbf{M}(t)\big(v_\theta(t,z) - v_t(z)\big)\Big] dt + C,\end{aligned} \tag{80}$$

where $C := C_1 - \int_0^1 \mathbb{E}_{z\sim p_t}\big[v_t(z)^\top \mathbf{M}(t) v_t(z)\big] dt$ is independent of θ. Since the law of $z \sim p_t$ coincides with that of x_t, the preceding display is equivalent to Eq. 12.

By Step 1, $\mathbf{M}(t)$ is positive definite for every t. Hence, for any vector $a \in \mathbb{R}^d$, $a^\top \mathbf{M}(t) a \geq 0$, with equality if and only if $a = 0$. Therefore, the integrand in Eq. 12 is nonnegative almost surely, and it vanishes if and only if

$$v_\theta(t,x_t) = v_t(x_t). \tag{81}$$

It follows that the weighted objective is minimized if and only if

$$v_\theta(t,x_t) = v_t(x_t) \qquad \text{for Lebesgue-a.e. } t \in [0,1] \text{ and } p_t\text{-a.e. } x_t \in \mathbb{R}^d. \tag{82}$$

Thus the unique minimizer of $\mathcal{L}_{\mathrm{FA}}(\theta)$, up to almost-everywhere equality, is

$$v_\theta^*(t,x_t) = v_t(x_t). \tag{83}$$

This completes the proof. □

D Experimental Details

D.1 Model Configuration

To start, all experiments are conducted on a node with 8×A800 GPUs. The experiment configurations of our model are summarized in Table 7. In practice, we follow the training setups from previous works such as DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024). Notably, existing methods utilize a patch size of 16. In our framework, the low- and high-frequency predictors operate on sub-states derived via DWT, which possess spatial dimensions of $H/2 \times W/2 \times C$. To maintain scale consistency with these approaches, we consequently employ a patch size of 8. For the frequency decomposition, we use a single-level orthonormal Haar DWT. For an input of shape $H \times W \times C$, the low-frequency component (LL) has shape $H/2 \times W/2 \times C$, while the high-frequency component is formed by concatenating the three detail sub-bands (LH, HL, HH) and has shape $H/2 \times W/2 \times 3C$.

Table 7: Configurations of experiments.

	FREPix-L	**FREPix-XL**
architecture		
DiT depth	22	28
hidden dim	1024	1152
heads	16	16
params	420M	674M
decoder depth	3	
decoder hidden dim	32	
patch size	8	
dropout	0.1	0.2
image size	256 (other settings: 512)	
representation alignment (Yu et al., 2024)		
alignment depth	8-th layer	
loss weight	0.5	
alignment encoder	Frozen DINOv2 (Oquab et al., 2024)	
perceptual supervision (Zhang et al., 2018)		
loss weight	0.5	
perceptual encoder	Frozen VGG	
training		
optimizer	AdamW (Loshchilov & Hutter, 2018), $\beta_1, \beta_2 = 0.9, 0.999$	
batch size	256	
learning rate	1e-4	
lr schedule	constant	
weight decay	0	
ema decay	0.9999	
time sampler	$\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2), \mu = -0.8, \sigma = 0.8$	
noise scale	1.0	
path smooth constant ε	0.01	
sampling		
ODE solver	Euler	Heun
ODE steps	50	100
timeshift	1.0	2.0
CFG scale	3.0 (256×256), 4.5 (512×512)	
CFG interval [28]	[0.1, 1]	

D.2 Detailed Architecture of FREPix

In this section, we provide a more detailed formulation of FREPix. Recall that at time t, the image state is decomposed by the orthonormal wavelet transform as

$$(l_t, h_t) = \mathcal{W}(x_t), \qquad x_t = \mathcal{W}^{-1}(l_t, h_t), \tag{84}$$

where l_t denotes the low-frequency sub-state and h_t denotes the high-frequency sub-state. For a single-level 2D DWT applied to an input image of shape $H \times W \times C$, the low-frequency component has shape $H/2 \times W/2 \times C$, while the high-frequency component is obtained by concatenating the three detail sub-bands and has shape $H/2 \times W/2 \times 3C$.

Low-frequency DiT. Firstly, the low-frequency branch (DiT) tokenizes l_t using non-overlapping patches of size $P \times P$. These patch vectors are projected into the DiT hidden space by a linear embedding layer $E_s(\cdot)$:

$$l_t^{\text{tok}} = \text{Unfold}(l_t) \in \mathbb{R}^{B \times L \times 3P^2}, \tag{85}$$

$$s_0 = E_s(l_t^{\text{tok}}) \in \mathbb{R}^{B \times L \times D}, \tag{86}$$

where $L = \frac{H/2}{P}\frac{W/2}{P}$ is the number of low-frequency patches. The condition vector c combines the timestep embedding and the class embedding:

$$c = \text{SiLU}(E_t(t) + E_y(y)) \in \mathbb{R}^{B \times 1 \times D}, \tag{87}$$

where $E_t(\cdot)$ denotes the timestep embedder and $E_y(\cdot)$ denotes the label embedding layer. The low-frequency tokens are then processed by K DiT blocks with 2D RoPE:

$$s_k = \text{DiTBlock}_k(s_{k-1}, c, \text{RoPE}), \qquad k = 1, \ldots, K. \tag{88}$$

After the final block, the low-frequency tokens are projected back to the patch domain:

$$l^{\text{tok}} = W_l(s_k) \in \mathbb{R}^{B \times L \times 3P^2}. \tag{89}$$

Finally, the clean low-frequency prediction is reconstructed by reshaping and folding these tokens back to the spatial grid:

$$\hat{l} = \text{Fold}\big(\text{Reshape}(l^{\text{tok}})\big) \in \mathbb{R}^{B \times 3 \times H/2 \times W/2}. \tag{90}$$

High-frequency decoder. The high-frequency branch follows a lightweight attention-free decoder from DeCo (Ma et al., 2026a). We first patchify the high-frequency component h_t and embed each patch with a linear layer $E_q(\cdot)$:

$$h_t^{\text{tok}} = \text{Unfold}(h_t) \in \mathbb{R}^{B \times L \times 9P^2}, \tag{91}$$

$$q_0 = E_q(h_t^{\text{tok}}) \in \mathbb{R}^{(BL) \times P^2 \times 9}. \tag{92}$$

The decoder condition is constructed from both the final low-frequency semantic token s_K and the predicted low-frequency patch token:

$$c' = \text{Reshape}\Big(s_K + W_{\text{s}}\big(\text{sg}(l^{\text{tok}})\big)\Big) \in \mathbb{R}^{(BL) \times D}. \tag{93}$$

The decoder itself is a stack of patch-local residual MLP blocks:

$$q_m = q_{m-1} + \alpha_m(c') \odot \text{MLP}_m\Big(\gamma_m(c') \odot \text{RMSNorm}(q_{m-1}) + \beta_m(c')\Big), \qquad m = 1, \ldots, M, \tag{94}$$

where $\alpha_m(\cdot)$, $\beta_m(\cdot)$, and $\gamma_m(\cdot)$ are AdaLN-Zero (Peebles & Xie, 2023) modulation parameters produced from the condition c', respectively. After the final block, the decoder predicts the clean high-frequency patch tokens:

$$h^{\text{tok}} = W_h(q_M) \in \mathbb{R}^{(BL) \times P^2 \times 9}. \tag{95}$$

The clean high-frequency prediction is reconstructed by reshaping and folding back to the spatial grid:

$$\hat{h} = \text{Fold}\big(\text{Reshape}(h^{\text{tok}})\big) \in \mathbb{R}^{B \times 9 \times H/2 \times W/2}. \tag{96}$$

Overall pipeline. The two predicted components are finally merged back into pixel space by the inverse DWT:

$$\hat{x} = \mathcal{W}^{-1}(\hat{l}, \hat{h}). \tag{97}$$

Therefore, the full generator can be written as

$$x_t \xrightarrow{\mathcal{W}} (l_t, h_t) \xrightarrow{f_\varphi,\, g_\phi} (\hat{l}, \hat{h}) \xrightarrow{\mathcal{W}^{-1}} \hat{x}. \tag{98}$$

This architecture explicitly factorizes the prediction targets: the DiT predicts clean low-frequency structure first, and the decoder then predicts clean high-frequency detail conditioned on that structure. Since FREPix adopts an x-prediction parameterization, the reconstructed clean image $\hat{x}$ is subsequently converted into the induced velocity for flow-matching training, as described in Sec. 3.3.

D.3 Class-to-Image Generation

This subsection provides further implementation details for class-to-image generation. For ImageNet class-to-image experiments, we initially train the XL-sized model (FREPix-XL) at 256×256 resolution for 320 epochs (1.6M steps), followed by fine-tuning at 512×512 resolution for an additional 10 epochs (50k steps). During inference, we use 100 steps Euler solver incorporated with Classifier-Free Guidance (CFG) and a guidance interval. The batch size and learning rate follow the default settings in Table 7. We utilize a global batch size of 256 and the AdamW optimizer with a constant learning rate of 1×10^{-4}. The time sampler utilizes a logit-normal distribution over t: $\text{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$, which aligns with JiT (Li & He, 2025). We set the CFG scale to 3.0 for 256×256 resolution (320 epochs) and 4.5 for 512×512 resolution (totaling 330 epochs). We use the CFG guidance interval of $[0.15, 1]$ for the default configuration.

D.4 Ablation Study

This subsection provides additional implementation details for the ablation studies. All ablation experiments are conducted using the L-sized model (FREPix-L). For computational efficiency, we train the models at 256×256 resolution for 40 epochs (200k steps). During inference, we utilize a 50-step Euler solver **without** CFG. The batch size and learning rate follow the default settings described previously. We use a global batch size of 256 and the AdamW optimizer with a constant learning rate of 1×10^{-4}. The time sampler employs a logit-normal distribution over t: $\text{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$. For power exponent ablation studies, we set the reweighting strength to $\omega = 0.7$ (our final configuration). For ablation studies of reweighting strength, we employ power exponents of $\gamma_l = 0.95$ and $\gamma_h = 1.05$ (our final settings). For ablation studies of decoupling strategies, our model maintains the same parameter settings as the main experiments ($\omega = 0.7$, $\gamma_l = 0.95$, and $\gamma_h = 1.05$). To ensure a fair comparison, all models are trained with Large-sized and sampled using the same steps.

E PSEUDO-CODES FOR TRAINING AND SAMPLING

In this section, we provide the detailed pseudo-codes for the training and sampling procedures of our proposed framework.

Algorithm 1 Training step

Require: f_φ: low-frequency predictor; g_ϕ: high-frequency predictor; x: training batch; $\lambda_l(t), \lambda_h(t)$: time-dependent weights.

$t = \text{sample_t}()$
$\epsilon = \text{randn_like}(x)$ ▷ Sample Gaussian noise
$(l, h) = \mathcal{W}(x), \quad (\epsilon_l, \epsilon_h) = \mathcal{W}(\epsilon)$ ▷ DWT
$l_t = g_l(t)\, l + \big(1 - g_l(t)\big)\epsilon_l, \quad h_t = g_h(t)\, h + \big(1 - g_h(t)\big)\epsilon_h$ ▷ Heterogeneous interpolation
$v_t^l = \dot{g}_l(t)(l - \epsilon_l), \quad v_t^h = \dot{g}_h(t)(h - \epsilon_h)$ ▷ Target velocity
$\hat{l} = f_\varphi(l_t, t)$ ▷ Predicted clean low-freq
$\hat{h} = g_\phi(h_t, \hat{l}, t)$ ▷ Predicted clean high-freq
$\hat{x} = \mathcal{W}^{-1}(\hat{l}, \hat{h})$ ▷ Predicted clean image
$v_\theta^l = \frac{\dot{g}_l(t)}{1 - g_l(t)}(\hat{l} - l_t), \qquad v_\theta^h = \frac{\dot{g}_h(t)}{1 - g_h(t)}(\hat{h} - h_t)$ ▷ Predicted velocity
$\mathcal{L}_{\text{FA}} = \lambda_l(t)||v_\theta^l - v_t^l||^2 + \lambda_h(t)||v_\theta^h - v_t^h||^2$ ▷ Compute reweighted v-loss
loss $\leftarrow \mathcal{L}_{\text{FA}} + \mathcal{L}_{\text{REPA}} + \mathcal{L}_{\text{LPIPS}}$ ▷ Loss

Algorithm 2 Sampling step (Euler)

Require: x_t: current samples at t; t, t_{next}.

$x_{pred} \leftarrow \text{net}_\theta(x_t, t)$ ▷ Network prediction
$v_{pred} = \dot{G}(t)\big(I - G(t)\big)^{-1}(\hat{x} - x_t)$ ▷ Estimate velocity, $G(t)$ is in Eq. 3
$x_{next} \leftarrow x_{pred} + (t_{next} - t) \cdot v_{pred}$ ▷ Euler update
return x_{next}

F ADDITIONAL EXPERIMENTS AND RESULTS

This section provides more experimental results and qualitative results.

F.1 ADDITIONAL EXPERIMENTS

Comparison on early training steps. We compare the optimization efficiency of different pixel-space generation models at an early training stage. As shown in Table 8, after only 80 training epochs, FREPix achieves the best FID, IS and recall among all compared methods. These results suggest that FREPix performs favorably in the limited training budget regime.

Table 8: Comparison results using 100-step Euler sampling in pixel-space diffusion models. All models are trained with 80 epochs.

Method	FID↓	IS↑	Pre.↑	Rec.↑
DeCo-XL/16	2.57	-	-	-
PixelDiT	2.36	282.3	0.80	0.57
FREPix-XL	2.29	294.9	0.79	0.60

Computational comparison. To quantify the computational resources of latent-space and pixel-space generation models, we report the number of parameters, training epochs, GFLOPs, FID and Inception Score (IS) for each model. FLOPs are measured for a single forward pass at resolution (256×256), excluding sampling steps and CFG duplication. For prior works, we use the results reported in (Yu et al., 2026) and convert them to a unified convention where one multiply-add is counted as two FLOPs.

Table 9 compares latent-space and pixel-space generation models in terms of parameters, GFLOPs, FID and IS. Latent models achieve strong FID scores with approximately 240–290 GFLOPs, benefiting from generation in a compact latent space. In contrast, prior pixel-space models typically require several hundred to several thousand GFLOPs to approach a comparable quality regime, reflecting the substantially higher cost of modeling full-resolution pixels directly. Notably, **FREPix-XL** achieves an FID of **1.91** and an IS of **295.6** with only **230 GFLOPs**. This not only obtains competitive image quality results in existing pixel-space models, but also closes much of the gap to the strongest latent-space models, while using slightly less computation than common latent baselines and substantially less computation than most prior pixel-space methods. These results suggest that explicit frequency-heterogeneous modeling significantly improves the computation–quality trade-off of pixel-space generation, narrowing the efficiency gap between pixel-space and latent-space generative models.

Table 9: Computation comparison on latent-space and pixel-space generation models at resolution 256×256. Text in gray: latent diffusion models that require VAE.

Method	Params	Epochs	GFLOPs	FID↓	IS↑
DiT-XL/2 (Peebles & Xie, 2023)	675M + 86M	1400	238	2.27	278.2
SiT-XL/2 (Ma et al., 2024)	675M + 86M	1400	238	2.06	277.5
REPA-XL/2 (Yu et al., 2024)	675M + 86M	800	238	1.42	305.7
ADM (Dhariwal & Nichol, 2021)	554M	400	2240	4.59	186.7
RIN (Jabri et al., 2023)	410M	480	668	3.42	182.0
SiD, UViT/2 (Hoogeboom et al., 2023)	2B	-	1110	2.44	256.3
VDM++, UViT/2 (Kingma & Gao, 2023)	2B	-	1110	2.12	267.7
JiT-G/16 (Li & He, 2025)	2B	600	766	1.82	292.6
PixelFlow-XL/4 (Chen et al., 2025)	677M	320	5818	1.98	282.1
DeCo-XL/16 (Ma et al., 2026a)	682M	320	237	1.90	303.0
PixelDiT-XL (Yu et al., 2026)	797M	320	311	1.61	292.7
PixNerd-XL/16 (Wang et al., 2025)	700M	320	268	2.15	297.0
FREPix-XL	674M	320	230	1.91	295.6

CFG guidance scale and interval. We report the classifier-free guidance (CFG) settings used for FREPix-XL on ImageNet 256×256. Table 10 lists the CFG scale, guidance interval, and the resulting gFID, Inception Score (IS), precision, and recall for models trained for 80 and 320 epochs. For the 80 epochs, the best FID of 2.29 is achieved with a relatively higher CFG value of 3.0 and an interval of $[0.15,\ 1]$. For the 320 epochs, the best FID of 1.91 is achieved with a relatively higher CFG scale of 3.0 and an interval of $[0.15,\ 1]$. Compared with other settings, this configuration slightly reduces IS and precision while improving recall.

Table 10: CFG settings and results for FREPix-XL at resolution 256×256. For sampling, we use Euler solver with 100 steps.

Training Steps	Epochs	CFG value	CFG interval	FID↓	IS↑	Pre.↑	Rec.↑
400k	80	2.75	$[0.1,\ 1.0]$	2.61	294.3	0.80	0.59
400k	80	2.75	$[0.15,\ 1.0]$	2.35	283.8	0.79	0.60
400k	80	3.00	$[0.1,\ 1.0]$	2.62	313.4	0.80	0.59
400k	80	3.00	$[0.15,\ 1.0]$	2.29	294.9	0.79	0.60
1600k	320	2.75	$[0.1,\ 1.0]$	2.09	310.0	0.80	0.61
1600k	320	2.75	$[0.15,\ 1.0]$	1.94	300.4	0.79	0.61
1600k	320	3.00	$[0.1,\ 1.0]$	2.07	317.6	0.81	0.61
1600k	320	3.00	$[0.15,\ 1.0]$	1.91	295.6	0.79	0.62

Compatibility with REPA. Table 11 shows that REPA consistently improves all three methods across the evaluated metrics. FREPix-L particularly reduces FID from 25.70 to 13.85 and increases IS from 65.25 to 105.6. These results demonstrate that FREPix is fully compatible with and benefits substantially from REPA-based representation alignment.

Table 11: Comparison results on compatibility with REPA.

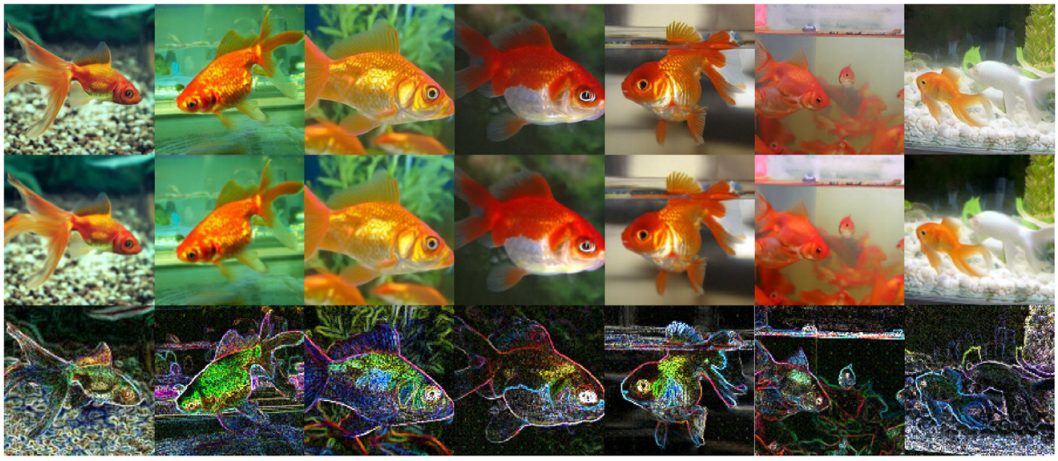
Method	REPA	Params	FID↓	IS↑	Pre.↑	Rec.↑
JiT-L	✗	459M	34.85	42.45	0.50	0.61
JiT-L	✓	459M	23.25	67.7	0.55	0.65
DeCo-L	✗	426M	67.55	19.10	0.47	0.56
DeCo-L	✓	426M	31.35	48.4	0.51	0.65
FREPix-L	✗	420M	25.70	65.25	0.57	0.53
FREPix-L	✓	420M	13.85	105.6	0.67	0.54

F.2 Additional Qualitative Results

We provide additional qualitative results to further assess the visual fidelity and frequency-decoupled generation behavior of FREPix. Fig. 8 to Fig. 15 shows uncurated ImageNet 256×256 samples generated by FREPix-XL (settings: train epochs: 320, cfg value: 3.0, cfg guidance interval: $[0.1,\ 1.0]$). In addition to the final generated images, we visualize the corresponding low- and high-frequency components obtained by the same wavelet decomposition used in our method. These results demonstrate that FREPix produces coherent global structures in the low-frequency branch while preserving localized details and textures in the high-frequency branch.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 8: Uncurated samples generated by FREPix-XL conditioned on class 1: *goldfish*.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 9: Uncurated samples generated by FREPix-XL conditioned on class 19: *chickadee*.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 10: Uncurated samples generated by FREPix-XL conditioned on class 22: *bald eagle*.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 11: Uncurated samples generated by FREPix-XL conditioned on class 88: *macaw*.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 12: Uncurated samples generated by FREPix-XL conditioned on class 107: *jellyfish*.

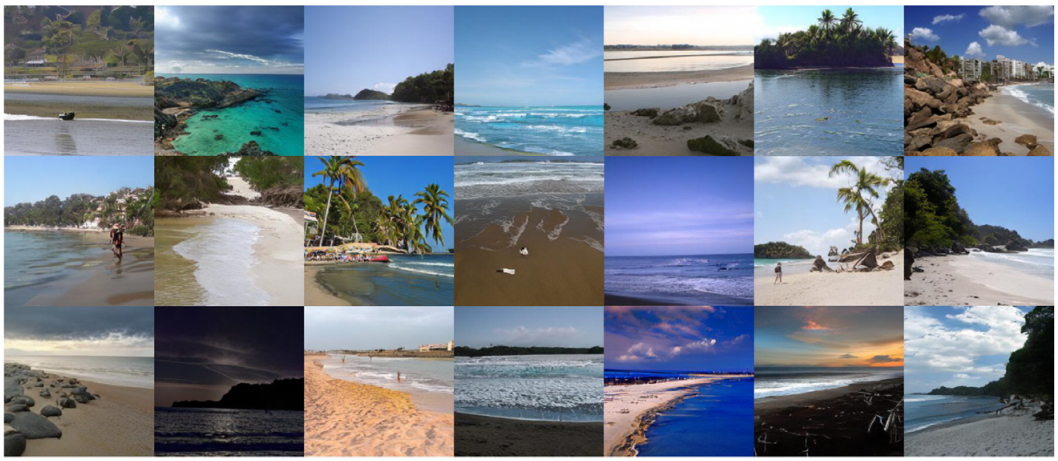


(a) Uncurated class-conditional samples generated by FREPix-XL.

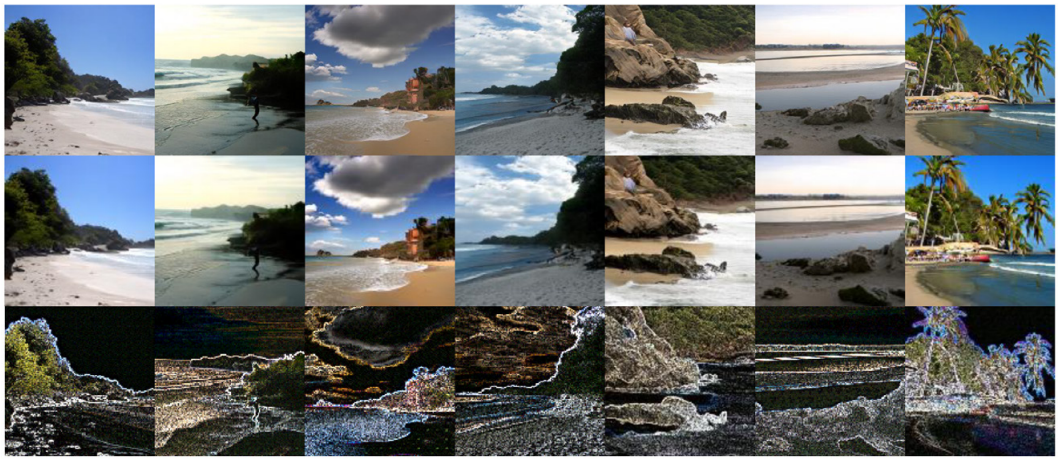


(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 13: Uncurated samples generated by FREPix-XL conditioned on class 108: *sea anemone*.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 14: Uncurated samples generated by FREPix-XL conditioned on class 978: *seashore*.



(a) Uncurated class-conditional samples generated by FREPix-XL.



(b) Frequency-decoupled visualization of the generated samples. From top to bottom: generated image, low-frequency component (LF), and high-frequency component (HF). For clarity, HF is displayed using coefficient magnitudes with logarithmic compression, and LF/HF are normalized separately.

Figure 15: Uncurated samples generated by FREPix-XL conditioned on class 979: *valley*.