PATCHAFR: NOISE-CORRECTED PATCH-ADAPTIVE FREQUENCY ROUTING FOR PIXEL-SPACE DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel space diffusion models generate images directly from raw pixels without relying on pretrained tokenizers, but their high dimensional input space makes training from scratch expensive. Existing frequency based approaches mainly characterize frequency recoverability at the global image level. We observe that this global treatment overlooks an important property of images: the same frequency band can have different local signal strength relative to the diffusion noise across spatial regions, resulting in spatially varying recoverability. Based on this observation, we introduce PatchAFR, a lightweight **Patch**-**A**daptive **F**requency **R**outing method that can be integrated into pixel space diffusion models. We formulate frequency recoverability through local spectral evidence and derive a timestep aware correction to estimate this evidence from noisy inputs. PatchAFR then jointly uses the corrected local evidence and diffusion timestep to adaptively route frequency bands for each spatial patch. Experiments on the pixel space diffusion model JiT demonstrate improvements throughout training. On JiT B/32, PatchAFR reduces FID by 8.4%, 7.7%, and 9.1% at epochs 60, 120, and 200, respectively, and requires 29% fewer training epochs to reach the same FID over 600 epochs.

1 INTRODUCTION

Pixel-space diffusion models (PDMs) perform diffusion directly in pixel space, avoiding information loss introduced by compression, and have recently attracted increasing attention for image generation. However, the distribution of image pixel space is sparse and highly non-uniform (Li & He, 2026; Shin et al., 2026), making PDMs expensive to optimize when trained from scratch. Recent studies have therefore increasingly focused on accelerating the training of PDMs.

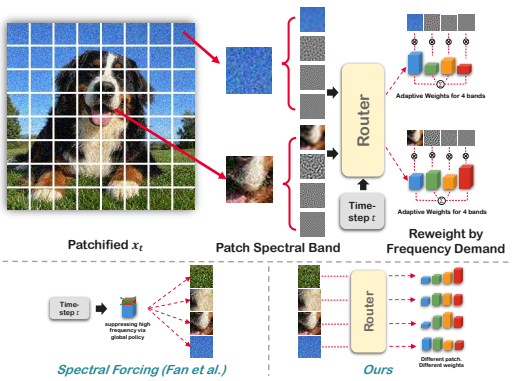

Figure 1: Global vs patch-adaptive frequency routing. A global policy uses the same frequency strategy for all patches, whereas PatchAFR predicts patch-specific weights for the four bands.

These approaches fall into two categories. The first introduces additional perceptual or representation supervision to accelerate convergence (Ma et al., 2026b; Shin et al., 2026). However, these methods require external vision foundation models (VFMs), and their effectiveness inherently depends on the quality of the representations provided by these models. The second line of work focuses instead on the learning process of PDMs themselves (Wang et al., 2026a; Lei et al., 2026; Tong et al., 2026). These methods are motivated by the observation that PDMs learn low-frequency information more readily. Based on that, suppressing high-frequency components to some extent can accelerate model training. For example, Spectral Forcing shows that the range of recoverable frequencies evolves with the diffusion

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.

timestep (Fan et al., 2026). It partitions the spectrum into global frequency bands and applies a time-dependent low-pass Discrete Cosine Transform (DCT) filter to the noisy input.

We argue that treating these frequency bands uniformly across the image is suboptimal. Although the bands are defined globally, their responses can be highly non-uniform across different spatial regions. As illustrated in Figure 1(top), we patchify a 256×256 noisy image x_t using a patch size of $p = 32$. Different patches can exhibit very different local frequency characteristics: a visually smooth sky region is dominated by low-frequency content, whereas a textured fur region contains substantially richer high-frequency details. Importantly, such differences persist even when the two regions are represented by the same globally defined DCT frequency bands.

This observation raises a simple question: *Can globally defined frequency components be used differently across spatial regions?*

To answer this question, we show that **the frequency components needed by a PDM vary across spatial regions**. In particular, this need is determined by the relative strength of the local image signal with respect to the corresponding diffusion noise floor. However, diffusion noise itself contaminates the measured raw band energy. We therefore derive a timestep-aware correction that substantially reduces this interference and provides a more reliable estimate of local spectral evidence under diffusion noise.

Motivated by this analysis, we propose PatchAFR, illustrated in Figure 1(bottom). PatchAFR builds on JiT with three simple components. First, we decompose the noisy input into B radial frequency bands using a global DCT and reconstruct each band as a full-resolution image. Second, for each patch and frequency band, we correct the observed local energy for diffusion noise to estimate the underlying signal energy. Third, a lightweight two-layer MLP router jointly uses these noise-corrected energies and the diffusion timestep to predict patch-wise routing preferences over frequency bands. The resulting logits are projected into soft gates under a timestep-dependent frequency budget, and the gated bands are recombined before entering the JiT backbone. Different from Spectral Forcing (Fan et al., 2026) which suppresses high frequency signals according to the timestep, our method learns a router to adaptively adjust the combination of spectral bands.

At matched training epochs, our method generally achieves lower FID than vanilla JiT (Li & He, 2026) across JiT model configurations. On JiT B/32, our method reduces FID by 8.4%, 7.7%, and 9.1% at epochs 60, 120, and 200, respectively. At epoch 200, it reaches an FID of 11.38, compared with 12.52 for JiT. Across epochs up to 600, our method achieves lower FID than JiT and requires 29% fewer epochs to reach the same FID.

Our main contributions are summarized as follows:

- We introduce PatchAFR, a patch-wise frequency routing approach for JiT, together with theoretical grounding in local spectral recoverability and diffusion-noise correction.
- We develop a patch-level spectral analysis for pixel-space diffusion. Our analysis reveals that globally defined frequency bands can have substantially different reliability across spatial regions. We further derive a timestep-aware correction for estimating local spectral evidence in the presence of diffusion noise.
- Guided by this analysis, we develop a lightweight patch-wise frequency router for JiT. Using noise-corrected local spectral evidence together with the diffusion timestep, the router adaptively modulates globally defined frequency-band responses for each spatial region. Across different JiT model configurations, our method generally improves FID at matched training epochs, reducing FID by 9.1% on JiT-B/32 on epoch 200.

2 Related Work

Pixel-Space Diffusion and Training Efficiency. Pixel-space diffusion models (PDMs) (Hoogeboom et al., 2025; Chen et al., 2025; Wang et al., 2026b; Li & He, 2026; Chen et al., 2026; Ma et al., 2026a; Lei et al., 2026; Yu et al., 2026; Guo et al., 2026) directly model raw images, avoiding the pretrained tokenizer and reconstruction bottleneck of latent diffusion (Vahdat et al., 2021; Rombach et al., 2022; Peebles & Xie, 2023; Podell et al., 2024; Pernias et al., 2024; Chen et al., 2024; Esser et al., 2024; Xie et al., 2025; Ma et al., 2024). JiT (Li & He, 2026) shows that clean-data prediction

with large-patch image Transformers enables competitive pixel-space generation without tokenizers, pretraining, or auxiliary losses. However, the high dimensionality of pixel space makes optimization difficult and convergence slow, motivating recent efforts to improve PDM training. EPG (Lei et al., 2026) introduces self-supervised pretraining followed by end-to-end generative fine-tuning, while PixelREPA (Shin et al., 2026) adapts representation alignment (Yu et al., 2025) to pixel-space diffusion after showing that direct semantic alignment can be ineffective for JiT. In contrast, our work improves PDM training by exploiting the spectral structure of the noisy pixel input.

Frequency-Aware Diffusion Models. Frequency-dependent learning dynamics have been studied from several perspectives in diffusion models. DPM-Slim (Yang et al., 2023) observes timestep-dependent frequency recovery and high-frequency degradation in compact latent diffusion models, motivating wavelet gating and spectrum-aware distillation. FreqWarm (Lai et al., 2026) studies the frequency mismatch between RGB images and high-dimensional latent representations and introduces a frequency warm-up curriculum, while LWD (Sigillo et al., 2025) uses frequency-sensitive spatial supervision to emphasize detail-rich regions. FaPS (Wang et al., 2026c) exploits spectral bias across both training iterations and diffusion timesteps to adaptively select image patches for efficient training. In pixel space, DeCo (Ma et al., 2026a) separates low-frequency semantic modeling from high-frequency detail generation, while Spectral Forcing (Fan et al., 2026) derives a timestep-dependent global recoverability boundary and applies DCT-based low-pass filtering to the noisy input. Most closely related to our work, Spectral Forcing applies the same frequency rule across the image at each timestep; we instead model spatially varying spectral evidence and perform patch-adaptive frequency routing.

3 Motivation: Local Spectral Recoverability

Spectral Forcing shows that frequency recoverability in diffusion models evolves with the diffusion timestep: as the noisy state approaches the data endpoint, progressively higher frequency components become distinguishable from diffusion noise (Fan et al., 2026). This observation motivates a timestep dependent global recoverability boundary, which characterizes which frequency components are recoverable at a given noise level.

However, such a global characterization implicitly aggregates spectral evidence over the entire image. Natural images are spatially heterogeneous. A globally defined frequency band may contain strong image signal in one region but very little signal in another. For example, a high frequency band can carry substantial structural information in a textured region while containing almost no clean image energy in a smooth region, as illustrated in Figure 1. Therefore, although the frequency coordinate system is global, the reliability of a given frequency band need not be spatially uniform.

We formalize this distinction by keeping the frequency decomposition global while measuring its recoverability locally. Let $x \in \mathbb{R}^n$ denote a clean image and let $\epsilon \sim \mathcal{N}(0, I_n)$ denote white Gaussian noise. Following rectified flow, the noisy input is

$$z_t = tx + (1-t)\epsilon, \qquad t \in (0,1). \tag{1}$$

Let D denote the orthonormal channel wise DCT, M_b the mask corresponding to global frequency band b, and R_p the spatial extractor for patch p. We define

$$\Pi_b = D^\top M_b D, \qquad A_{p,b} = R_p \Pi_b. \tag{2}$$

Here, Π_b first projects the full image onto the globally defined frequency band b, while R_p subsequently extracts its spatial response inside patch p. Thus, all patches share exactly the same frequency definition, and only the local evidence associated with that frequency band is allowed to vary spatially.

Applying this operator to the noisy input gives

$$A_{p,b} z_t = t A_{p,b} x + (1-t) A_{p,b} \epsilon. \tag{3}$$

For patch p and band b, let d denote the number of scalar entries in the patch. We define the clean local signal energy and the expected unit noise energy as

$$S_{p,b}(x) = \frac{1}{d} \left\| A_{p,b} x \right\|_2^2, \qquad N_{p,b} = \frac{1}{d} \mathbb{E}_\epsilon \left[\left\| A_{p,b} \epsilon \right\|_2^2 \right]. \tag{4}$$

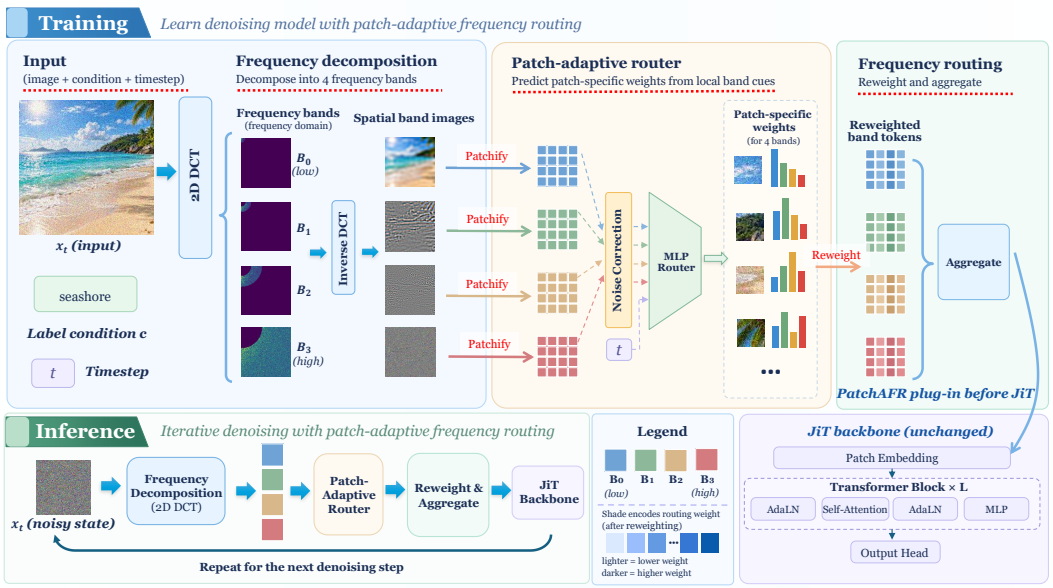

Figure 2: Overview of PatchAFR. Given a noisy input and its timestep, PatchAFR follows the three stages from left to right: frequency decomposition, noise-corrected local evidence estimation, and patch-adaptive routing and recombination. The routed input is then processed by the unchanged JiT backbone during both training and inference.

Spectral Forcing (Fan et al., 2026) formulates per-band data-to-noise ratio as $\frac{t^2 P_x(k)}{(1-t)^2}$ to motivate global recoverability, where $P_x(k)$ is approximated by $P_x(k) \propto k^{-\alpha}$. However, we define the local recoverability score with patch-wise relative energy strength, notated as

$$\rho_{p,b}(t;x) = \underbrace{\frac{t^2}{(1-t)^2}}_{\text{global noise state}} \quad \underbrace{\frac{S_{p,b}(x)}{N_{p,b}}}_{\text{local spectral evidence}} . \tag{5}$$

This factorization makes the two sources of recoverability explicit. The first term is determined solely by the diffusion timestep and therefore describes the global signal to noise state shared by the entire image. The second term measures how strongly the globally defined frequency band b is supported within patch p. Since this local spectral evidence depends on the image content through $S_{p,b}(x)$, patches at the same timestep can have substantially different recoverability for the same frequency band.

Therefore, the timestep only captures the overall trend of frequency recoverability. To use recoverable frequency information more accurately, we also need to consider the local spectral evidence in each spatial region.

4 PatchAFR: Patch-Adaptive Frequency Routing

Motivated by the local spectral recoverability discussed in Section 3, we introduce PatchAFR, a patch-adaptive frequency routing module placed before the JiT backbone. Given the noisy input z_t and timestep t, it produces a frequency-routed input $\widetilde{z}_t$ with the same spatial resolution as z_t, which is then directly processed by JiT. The routing decision is patch-adaptive and is determined from the local spectral evidence of the noisy input together with the diffusion timestep.

Figure 2 presents the PatchAFR pipeline. The upper row shows PatchAFR training, where PatchAFR is inserted as an independent component before the unchanged JiT backbone. The lower row illustrates inference, where PatchAFR is also used to make routing decisions at each denoising step. The complete PatchAFR pipeline consists of three stages: frequency decomposition and patch

tokenization, noise-corrected local spectral evidence estimation, and patch-adaptive frequency routing and band recombination. We describe each stage below.

4.1 Frequency Decomposition and Patch Tokenization

As shown in the frequency decomposition block of Figure 2, we first decompose the noisy input z_t into $B = 4$ frequency bands. Specifically, we apply a channel wise 2D DCT to the full image and partition the resulting spectrum using four disjoint radial masks $\{M_b\}_{b=1}^{B}$, where $\sum_{b=1}^{B} M_b = I$. For each band, we apply the corresponding mask and transform it back to the spatial domain using the inverse DCT, giving $z_t^{(b)} = D^\top M_b D z_t$.

The four resulting band images have the same spatial resolution and remain aligned with the original noisy input. We then apply the same patch partition used by JiT to each band image. For patch p and band b, the resulting band token is $u_{t,p,b} = R_p z_t^{(b)} = A_{p,b} z_t$. Thus, each spatial location is represented by four aligned band tokens $\{u_{t,p,b}\}_{b=1}^{B}$, which provide the local frequency observations used by the subsequent router.

Since these band tokens are extracted directly from the noisy state z_t, their responses contain both image signal and diffusion noise. We therefore estimate the local spectral evidence before using them for patch adaptive routing.

4.2 Noise Corrected Local Spectral Evidence

Recall from Eq. 5 that the local recoverability of patch p and frequency band b depends on the clean local spectral energy $S_{p,b}(x)$. Although the clean image x is available when constructing training samples, PatchAFR is designed to make its routing decision from the current noisy state z_t at both training and inference. Therefore, we estimate $S_{p,b}(x)$ only from the noisy band token $u_{t,p,b} = A_{p,b} z_t$ obtained in the previous stage.

Let d denote the number of scalar entries in one band token. We first measure its raw local energy as $E_{p,b}^{\mathrm{raw}}(t) = \frac{1}{d}\|u_{t,p,b}\|_2^2$. We then define the noise corrected local spectral evidence as

$$\widehat{S}_{p,b}(t) = \frac{\left[E_{p,b}^{\mathrm{raw}}(t) - (1-t)^2 N_{p,b}\right]_+}{t^2 + \eta}, \qquad [u]_+ = \max(u, 0), \tag{6}$$

where $N_{p,b}$ is the expected unit noise energy defined in Eq. 5, and $\eta > 0$ is a small constant used for numerical stabilization. The term $(1-t)^2 N_{p,b}$ represents the expected noise energy contributed by the diffusion process, while t^2 accounts for the attenuation of the underlying image signal.

The form of this correction follows directly from the rectified flow observation $z_t = tx + (1-t)\epsilon$. Since $u_{t,p,b} = A_{p,b} z_t$, its expected raw energy satisfies

$$\mathbb{E}_\epsilon\left[E_{p,b}^{\mathrm{raw}}(t) \mid x\right] = \underbrace{t^2 S_{p,b}(x)}_{\text{local image signal}} + \underbrace{(1-t)^2 N_{p,b}}_{\text{diffusion noise}}. \tag{7}$$

Therefore, the raw energy $E_{p,b}^{\mathrm{raw}}(t)$ cannot be directly interpreted as the local spectral evidence required by Eq. 5, because it contains an additive noise contribution that varies with the timestep. Subtracting $(1-t)^2 N_{p,b}$ removes this contribution in expectation, and dividing by t^2 compensates for the attenuation of the clean signal. In particular, the corresponding unstabilized estimator $\widetilde{S}_{p,b}(t) = \big(E_{p,b}^{\mathrm{raw}}(t) - (1-t)^2 N_{p,b}\big)/t^2$ satisfies $\mathbb{E}_\epsilon[\widetilde{S}_{p,b}(t) \mid x] = S_{p,b}(x)$.

This correction removes the systematic noise contribution in expectation, but the division by t^2 in $\widehat{S}_{p,b}(t)$ becomes numerically unstable near the noise endpoint. We therefore provide the router with the noise-subtracted residual $E_{p,b}^{\mathrm{raw}}(t) - (1-t)^2 N_{p,b}$ as a more stable local spectral feature. In addition, inspired by the signal-to-noise balance, we include the auxiliary feature $\frac{t\widehat{S}_{p,b}(t) - (1-t)N_{p,b}}{t\widehat{S}_{p,b}(t) + (1-t)N_{p,b} + \eta}$, together with the timestep t, to characterize the diffusion-dependent reliability of each patch-band observation.

4.3 Patch Adaptive Routing and Band Recombination

Given the band tokens $\{u_{t,p,b}\}_{b=1}^{B}$ and corresponding noise corrected routing features, PatchAFR predicts how much each frequency band should contribute to every spatial patch. For each patch p, the router takes the local band information together with the timestep t and produces one routing logit $\ell_{p,b}$ for each frequency band b. The router is a lightweight trainable network whose parameters are optimized jointly with the JiT backbone under the original diffusion training objective.

The routing logits are converted into soft gates

$$g_{p,b} = \sigma\left(\frac{\ell_{p,b} + \delta}{T}\right), \tag{8}$$

where T is the routing temperature and δ is a sample specific offset used to match the timestep dependent global frequency budget. In particular, δ is chosen such that the average retained frequency content follows the global recoverability schedule, while the relative allocation across patches and bands is determined by the trainable router.

The routed representation of patch p is then obtained by reweighting and summing its aligned band tokens:

$$\widetilde{z}_{t,p} = \sum_{b=1}^{B} g_{p,b} u_{t,p,b}, \qquad \widetilde{z}_t = \mathrm{Unpatchify}\left(\{\widetilde{z}_{t,p}\}_p\right). \tag{9}$$

The resulting $\widetilde{z}_t$ has the same spatial resolution as the original noisy input and is directly passed to the unchanged JiT backbone.

Importantly, PatchAFR performs routing online. During training, the router is jointly learned with JiT and its routing weights are recomputed from the current noisy sample at every sampled timestep. During inference, the learned router parameters remain fixed, but the gates $\{g_{p,b}\}$ are recomputed at every denoising step from the current noisy state and timestep. The frequency allocation can therefore evolve throughout the diffusion trajectory and adapt simultaneously to the current noise level and local image content.

5 Experiments

5.1 Implementation Details

We build on the official JiT (Li & He, 2026) implementation and follow its training setup. We train all models on ImageNet (Deng et al., 2009) at 256×256 resolution. We use four radial frequency bands and a lightweight two-layer MLP router. Unless otherwise specified, we report FID-50K using the EMA model with 50-step Heun sampling, a CFG (Ho & Salimans, 2022) scale of 4.0 over the interval $(0.1, 1.0)$, and sampling seed 17. Results across different CFG scales and sampling seeds, together with further implementation details, are provided in the appendix.

5.2 Comparison with other methods.

In Tab. 1, we first compare PatchAFR with methods developed for pixel-space diffusion. Compared with the JiT baseline, PatchAFR shows consistent gains throughout training. It also outperforms Spectral Forcing (Fan et al., 2026) at all evaluated stages, with a further reduction in FID, indicating stronger improvements from the early and middle stages that persist later in training. In Figure 3, we compare the long-run training dynamics of JiT B/32, Spectral Forcing, and PatchAFR (uniform band partition and noise corrected energy). PatchAFR consistently achieves lower FID after epoch 60, and the advantage is maintained throughout training. Up to epoch 600, it reaches comparable FID with 29% fewer training epochs. It confirms the advantages of patch-wise frequency routing.

We also adapt several frequency methods originally developed for LDMs to JiT. These methods often rely on spectral characteristics specific to latent representations. For example, FreqWarm (Lai et al., 2026) models the frequency shift from RGB images to latent features, and LWD (Sigillo et al., 2025) derives wavelet saliency from the latent space. Such assumptions do not directly hold for JiT, which operates on raw noisy pixels without a latent representation. In contrast, PatchAFR is designed for pixel-space diffusion: it estimates noise-corrected local spectral evidence from the current noisy input

Table 1: Quantitative comparison on JiT B/32. LDM-oriented methods are adapted to the JiT framework for comparison. FID is reported at different training epochs; lower is better. Best results are bolded.

Method	60	120	200
LDM-oriented methods			
FreqWarm (Lai et al., 2026)	26.87	14.76	11.92
FaPS (Wang et al., 2026c)	36.82	22.23	17.41
LWD (Sigillo et al., 2025)	44.74	29.27	23.46
FDS (Ren et al., 2025)	30.94	18.66	15.39
DPM-Slim (WG) (Yang et al., 2023)	26.13	14.99	12.15
PDM-oriented methods			
JiT (Li & He, 2026)	27.18	15.72	12.52
SpectralForcing (Fan et al., 2026)	26.09	15.38	12.03
PatchAFR	**24.89**	**14.51**	**11.38**

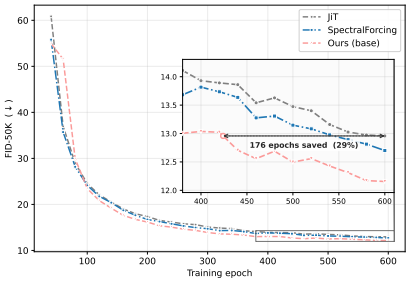

Figure 3: FID curves of JiT B/32, Spectral Forcing, and PatchAFR. Results are evaluated at CFG 2.9. PatchAFR outperforms the baselines after epoch 60 and reaches the same FID with 29% fewer training epochs.

and performs patch-adaptive routing accordingly. As shown in Table 1, this design yields the best FID among all compared methods.

5.3 Comparison with JiT baselines across model scales

As shown in Table 2, our method consistently improves the JiT baseline on both B/32 and L/32. On B/32, it reduces FID by 8.4%, 7.7%, and 9.1% at epochs 60, 120, and 200, respectively. L/32 also shows improvements throughout training. We provide a detailed discussion of the B/16 results and budget sensitivity in the appendix. Moreover, PatchAFR can be further combined with PixelREPA (Shin et al., 2026), leading to additional improvements on B/16. We report this combination only for B/16, as PixelREPA (Shin et al., 2026) relies on DINO features and is not readily applicable to the patch-32 setting. We visualize generated samples in Figure 8.

Table 2: Quantitative comparison across model scales. FID is reported at different training epochs.

Method	60	120	200
JiT B/32	27.18	15.72	12.52
PatchAFR	24.89	14.51	11.38
JiT L/32	13.19	7.62	6.88
PatchAFR	12.32	7.47	6.78
JiT B/16	10.59	5.38	4.77
PatchAFR	10.24	5.64	5.35
PatchAFR +PixelREPA	8.46	4.71	4.14

5.4 Ablation Study

Local Spectral Evidence Improves Routing. Sec. 3 shows that frequency recoverability depends not only on the global noise state determined by the timestep, but also on local spectral evidence. To evaluate the contribution of local information, we compare a global router using only the timestep embedding with our patch-wise router using local spectral evidence. As shown in Table 3, the patch-wise router achieves lower FID at all evaluated stages. This indicates that timestep information alone is insufficient to characterize spatially varying frequency recoverability, while local spectral evidence provides more effective routing cues.

Table 3: Global (timestep-only) vs. patch-wise routing on L/32. Patch-wise routing consistently improves FID across reported epochs.

	60	120	200
Timestep Global Router	13.32	7.84	6.98
Patch-wise Router	12.32	7.47	6.78

Table 4: Ablation of router inputs on L/32. FID-50K (↓) at different training epochs.

Router Feature	60	120	200
Noise Corrected Energy	13.74	7.85	7.08
Residual Feature	13.30	7.82	7.10
Residual + Aux	12.32	7.47	6.78

Noise Correction Is Useful despite Early Instability. To evaluate the effect of noise correction, we compare raw band energy with our corrected spectral evidence. Raw energy is contaminated by timestep-dependent diffusion noise, whereas our correction removes the expected noise contribution and compensates for signal attenuation. As shown in Table 5, noise correction improves JiT. PatchAFR further improves performance across all reported checkpoints. The results confirm noise correction provides more effective routing.

Table 5: Ablation of noise correction on JiT B/32. Simple noise-corrected energy improves over raw energy despite an early spike, while PatchAFR further improves FID across the reported checkpoints.

Router Feature	60	120	200
JiT B/32	27.18	15.72	12.52
Raw Energy	26.43	15.17	11.97
Noise Corrected Energy	36.31	14.65	11.51
PatchAFR	**24.89**	**14.51**	**11.38**

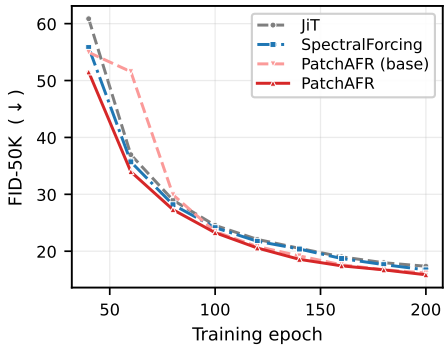

Figure 4: PatchAFR outperforms baselines, removes the early spike of the noise-corrected, uniform-band PatchAFR (base), and achieves stable gains.

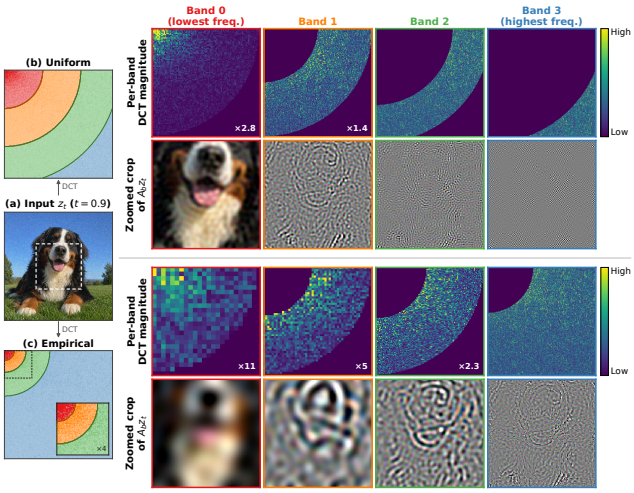

Figure 5: Visualization of uniform and empirical radial band partitions. Starting from the same noisy input z_t at $t = 0.9$, we show the radial partition, per-band DCT magnitude, and spatial reconstruction $A_b z_t$ for each band.

Table 6: Ablation of radial band edges on L/32. FID-50K (↓) at different training epochs, with all other components fixed.

Band Edges	40	80	120
Uniform	28.82	10.46	8.08
Empirical	**26.72**	**10.06**	**7.85**

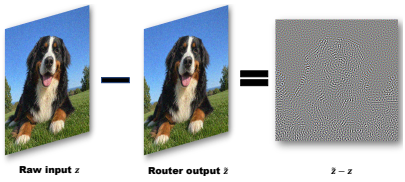

Figure 6: Router input–output visualization at t=0.9. The output is visually close to the input, while their difference highlights texture patterns and localized modifications.

Residual Features Resolve the Normalization Instability. We compare different spectral features for the patch wise router in Table 4. Directly using $\widehat{S}_{p,b}(t)$ is less effective due to the unstable t^2 normalization near the noise endpoint. Using the noise subtracted residual substantially improves FID, while adding the auxiliary signal to noise feature provides further gains across training.

Recoverability Guided Bands Outperform Uniform Partitioning. We compare uniform and empirical frequency partitions while keeping all other settings fixed. Uniform partitioning uses four equal-width DCT bands with edges $[0, 0.25, 0.50, 0.75, 1]$, whereas the empirical partition is derived from ImageNet-256 recoverability statistics, with approximate edges $[0, 0.06, 0.14, 0.30, 1]$. The intuition is that frequency recoverability does not change uniformly across the spectrum. Low- and

mid-frequency regions show more distinct recoverability changes and therefore benefit from finer partitioning, while a broader range of high frequencies can be grouped together. As shown in Figure 5 and Table 6, this empirical partition consistently achieves lower FID than uniform spacing.

5.5 Analysis

PatchAFR Improves DINO Alignment. To examine the learned representations, we measure unbiased linear CKA between intermediate features of JiT B/32 and our model at epoch 200 and DINOv2 (Oquab et al., 2024) features. As shown in Figure 7, at $t = 0.5$, our model achieves higher DINO alignment in nearly all layers, with the largest gain in the late-middle blocks. At other timesteps, the effect is more layer-dependent.

Adjustments Capture Spatially Structured Details. As shown in Figure 6, the router output remains visually close to the raw input, while their difference reveals clear spatially structured textures and fine-grained details.

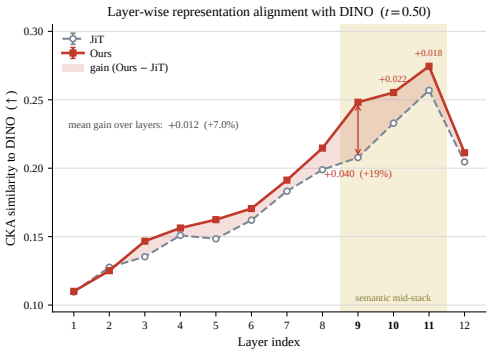

Figure 7: DINO alignment analysis. Layer-wise unbiased linear CKA with DINO at $t = 0.5$, where PatchAFR improves alignment over JiT-B/32 across nearly all layers.

Patch-Adaptive Routing Introduces Minimal Additional Cost. Our router introduces only 2,164 trainable parameters and approximately 1.01 GFLOPs per 256×256 image and forward pass. This corresponds to 6.61%, 1.90%, and 2.00% additional computation over JiT-B/32, JiT-L/32, and JiT-B/16, respectively, while the parameter increase is below 0.002% for all models. The overhead is dominated by the global DCT/IDCT transforms, whereas the patch-wise router MLP itself contributes less than 1 MFLOP. Consequently, increasing the number of routed patches from 64 to 256 has little effect on the total routing cost, making the relative overhead particularly small for larger or higher-resolution backbones. The training throughput is 1704.92 images/s for JiT B/32 and 1509.51 images/s for PatchAFR.



Figure 8: Qualitative results generated from PatchAFR +PixelREPA of B/16 at epoch 200.

6 Conclusion

We introduced PatchAFR, a patch-adaptive frequency routing method for pixel-space diffusion models. Experiments across model scales demonstrate its effectiveness, while ablations validate the key design choices. Overall, our results show that frequency recoverability varies across both diffusion timesteps and spatial regions, and that modeling this variation can improve training efficiency.

AI USE STATEMENT

In this work, we used generative AI tools, including ChatGPT (GPT-5.6), Codex, and Claude (Opus), as described below.

Tasks with required disclosure.

- **Code and implementation.** Generative AI was used as a coding assistant to improve code organization, structure, and readability, and to help implement a limited number of components based on algorithms and design choices specified by the authors. It was not used to determine the method design or algorithmic formulation. All AI-assisted code was reviewed and tested by the authors.
- **Mathematical derivations.** Generative AI was used to check selected intermediate steps and improve the presentation of mathematical derivations.

We did *not* use generative AI tools to generate synthetic datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for proving mathematical claims; propose or refine hypotheses; design or provide feedback on the research methodology or experiments; assist with translation; clean or reformat datasets; or interpret results. Writing proofs and qualitative or thematic data analysis are not applicable to this work.

Tasks with recommended disclosure.

- **Writing.** To assist with drafting and revising parts of the manuscript and to improve clarity and readability.
- **Literature and references.** To identify relevant prior work and related methods, locate references for standard mathematical results, and format BibTeX citation keys according to conventions specified by the authors.

Review and responsibility. We reviewed all AI-assisted work. AI-assisted code was reviewed and tested by the authors; the mathematical derivations were independently checked by the authors; and all cited works and mathematical results were independently checked against their original sources. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the source code and configuration files used in our experiments as a ZIP archive in the supplementary materials. We also include our implementations of the reproduced methods discussed in the paper.

REFERENCES

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024.

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

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36136–36146, 2026.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE

Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 12606–12633, 2024.

Weichen Fan, Haiwen Diao, Penghao Wu, and Ziwei Liu. Show the signal, hide the noise: Spectral Forcing for pixel-space diffusion. *arXiv preprint arXiv:2606.15236*, 2026.

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. Pixelu: A u-shaped transformer for efficient end-to-end pixel diffusion. In *European Conference on Computer Vision*, pp. 425–442. Springer, 2026.

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

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18062–18071, 2025.

Bolin Lai, Xudong Wang, Saketh Rambhatla, James M Rehg, Zsolt Kira, Rohit Girdhar, and Ishan Misra. Toward diffusible high-dimensional latent spaces: A frequency perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43450–43460, 2026.

Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no VAE: End-to-end pixel-space generative modeling via self-supervised pre-training. In *International Conference on Learning Representations*, volume 2026, pp. 102818–102839, 2026.

Tianhong Li and Kaiming He. Back to Basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 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, 2024.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026, Denver, CO, USA, June 3-7, 2026*, pp. 43600–43610. Computer Vision Foundation, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Ma_DeCo_Frequency-Decoupled_Pixel_Diffusion_for_End-to-End_Image_Generation_CVPR_2026_paper.html.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. PixelGen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026b.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

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.

Pablo Pernias, Dominic Rampas, Mats L. Richter, Christopher J. Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *International Conference on Learning Representations*, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

Yufan Ren, Zicong Jiang, Tong Zhang, Søren Forchhammer, and Sabine Süsstrunk. FDS: frequency-aware denoising score for text-guided latent diffusion image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 2651–2660. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00253. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ren_FDS_Frequency-Aware_Denoising_Score_for_Text-Guided_Latent_Diffusion_Image_Editing_CVPR_2025_paper.html.

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.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for Just Image Transformers is not easier than you think. In *European Conference on Computer Vision*, volume 17013 of *Lecture Notes in Computer Science*. Springer, 2026. doi: 10.1007/978-3-032-37271-0_33.

Luigi Sigillo, Shengfeng He, and Danilo Comminiello. Latent wavelet diffusion: Enabling 4k image synthesis for free. *CoRR*, abs/2506.00433, 2025. doi: 10.48550/ARXIV.2506.00433. URL https://doi.org/10.48550/arXiv.2506.00433.

Haoyang Tong, Yu He, Fang Li, Lichen Ma, Jingling Fu, Dong Chen, Zhen Chen, Junshi Huang, and Jie Cao. Energy-guided flow matching. *arXiv preprint arXiv:2608.05811*, 2026.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11287–11302, 2021.

Hainuo Wang, Mingjia Li, and Xiaojie Guo. WiT: Waypoint diffusion transformers via trajectory conflict navigation. *arXiv preprint arXiv:2603.15132*, 2026a.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, 2026b.

Xianglu Wang, Bangxian Han, and Hu Ding. Faps: A general and fast training method for diffusion models. In *Forty-third International Conference on Machine Learning*, 2026c.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *International Conference on Learning Representations*, 2025.

Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 22552–22562. IEEE, 2023. doi: 10.1109/CVPR52729.2023.02160. URL https://doi.org/10.1109/CVPR52729.2023.02160.

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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

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.

A More Experimental Results

Spatial variation of learned routing. To examine whether PatchAFR learns a patch-adaptive routing policy, we analyze the spatial variation of its routing outputs across the 8×8 patch grid. For each timestep and frequency band, we compute the within-image variation of the raw router logits across patches and aggregate it over 1,000 validation images. As shown on the left of Figure 9, the raw logits exhibit clear spatial variation, which is strongest at intermediate and late timesteps and peaks at $t = 0.75$ in Band 2. The representative post-budget gate maps at $t = 0.5$ on the right further show that this spatial variation remains after budget matching: low-frequency bands are nearly uniform, whereas middle- and high-frequency bands exhibit stronger patch-wise variation.

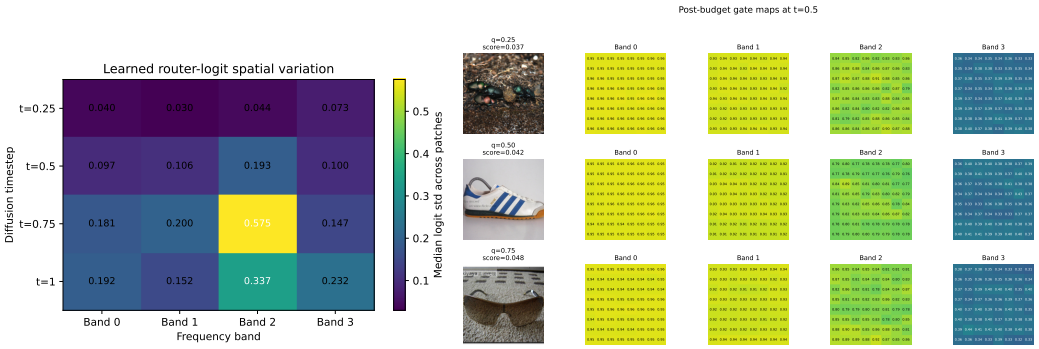

Figure 9: **Spatial variation of learned routing.** Left: median patch-wise standard deviation of the pre-budget router logits. Both are computed over the 8×8 patch grid and aggregated over 1,000 validation images. Right: representative gate maps at $t = 0.5$ for samples with low, median, and high aggregate gate variation.

Budget Sensitivity on B/16. We further study the effect of the routing budget on JiT B/16 in Table 7. Spectral Forcing reports its largest gains in the 64-token coarse-tokenization regime, while the improvement on JiT-B/16 with 256 tokens is within evaluator noise (Fan et al., 2026). Consistent with this result, our reproduced Spectral Forcing baseline performs substantially worse than JiT under our setting. Using the same SF schedule as the routing budget also degrades PatchAFR, suggesting that the SF-derived budget is not well suited to B/16. Without the budget constraint, PatchAFR slightly outperforms JiT at epoch 60, but the improvement plateaus after epoch 160. These results show that PatchAFR is sensitive to the target budget on B/16. Since our budget projection explicitly matches the prescribed gate mass, an inappropriate target budget can directly restrict the learned routing policy.

Table 7: Budget sensitivity on JiT B/16. FID-50K (↓) is reported at different training epochs.

Method	Budget	60	120	200
JiT B/16	none	10.59	5.38	4.77
SpectralForcing (Fan et al., 2026)	global sf	12.87	6.25	5.10
PatchAFR	sf match	13.33	5.83	5.12
PatchAFR	none	10.24	5.64	5.35
PatchAFR +PixelREPA	sf match	8.46	4.71	4.14

Sensitivity to sampling randomness. We assess the variability of generation metrics under repeated sampling from the same L/32 checkpoint at epoch 200. Tables 8 and 9 report FID and Inception Score for ten sampling seeds at each CFG scale from 1.0 to 7.0, using 10,000 generated images per evaluation. Statistics are computed separately for each guidance scale. Across these settings, the coefficient of variation (sample standard deviation divided by the mean) ranges from 0.49% to 1.51% for FID and from 0.59% to 1.75% for Inception Score. At the CFG scale of 4.0 used in our main evaluation protocol, FID is 9.2656 ± 0.0548 and Inception Score is 196.5560 ± 1.6095,

Table 8: Sampling-seed variability of FID (↓) for the L/32 checkpoint at epoch 200. Each entry uses 10,000 generated images. Mean and SD are computed across the ten seeds at each CFG scale; SD denotes the sample standard deviation ($n - 1$ denominator).

Seed	CFG 1.0	CFG 2.0	CFG 3.0	CFG 4.0	CFG 5.0	CFG 6.0	CFG 7.0
0	35.4340	14.8030	10.1114	9.2680	9.3988	9.9567	10.6085
1	36.0955	14.9517	10.1671	9.2492	9.3800	9.8709	10.5196
17	35.8660	14.8914	10.0930	9.1810	9.3374	9.8422	10.4925
36	36.7016	15.5258	10.3524	9.3636	9.4519	9.9011	10.5375
41	35.5673	14.8493	10.1627	9.3236	9.4292	9.8918	10.5174
82	35.8084	14.8470	10.1161	9.3018	9.4885	10.0225	10.6502
97	36.0220	15.0185	10.1635	9.2138	9.3530	9.7935	10.4507
218	35.7497	14.6775	10.0802	9.2536	9.3708	9.8568	10.4632
2222	36.4448	14.9145	10.0924	9.2190	9.3971	9.8618	10.5664
4096	36.2808	14.9164	10.1669	9.2825	9.3763	9.8151	10.4543
Mean	35.9970	14.9395	10.1506	9.2656	9.3983	9.8812	10.5260
SD	0.3949	0.2255	0.0789	0.0548	0.0463	0.0674	0.0667

Table 9: Sampling-seed variability of Inception Score (↑) under the same protocol as Table 8. Mean and SD are computed across the ten sampling seeds at each CFG scale; SD is not the variation across Inception Score splits.

Seed	CFG 1.0	CFG 2.0	CFG 3.0	CFG 4.0	CFG 5.0	CFG 6.0	CFG 7.0
0	44.8836	113.9092	165.3351	197.2731	215.3577	223.6170	227.6978
1	44.8715	113.4369	166.4624	196.0717	213.5645	222.6009	227.2787
17	44.3983	114.2379	164.6871	195.9029	214.9987	224.2602	227.8643
36	43.4667	111.3697	163.1350	194.2901	212.4338	222.7632	227.7401
41	45.0460	113.2988	164.2905	193.9360	211.4030	222.4441	227.5006
82	45.4859	116.3727	167.8251	198.8562	217.8223	226.0634	229.3655
97	44.6211	114.2392	165.7921	198.7004	215.4239	225.2479	230.9091
218	45.4811	114.4186	166.5733	196.8853	214.7884	224.3096	228.2802
2222	43.8355	115.6645	166.8112	196.8147	215.6416	226.2984	230.7657
4096	46.0861	115.7186	168.3361	196.8299	214.4377	225.5254	229.5155
Mean	44.8176	114.2666	165.9248	196.5560	214.5872	224.3130	228.6918
SD	0.7859	1.4402	1.6080	1.6095	1.7928	1.4400	1.3546

corresponding to relative standard deviations of 0.59% and 0.82%, respectively. These measurements indicate modest sampling-seed variability for this checkpoint under the evaluated settings. The reported standard deviations describe repeated sampling from a fixed trained model; they do not measure variability across independent training runs. This 10,000-sample analysis supplements the main FID-50K evaluation, and its absolute scores should not be directly compared with the 50,000-sample results.

Four Frequency Bands Are Sufficient. Table 10 compares 4 and 8 frequency bands. Using 8 bands slightly improves FID at early stages but does not improve the final result. This suggests that finer frequency partitioning can provide some early benefit, while four bands already capture sufficient spectral differences for effective routing.

Table 10: Ablation of the number of frequency bands on JiT L/32. FID-50K (↓) at different training epochs.

Band Num	60	120	200
4	12.32	**7.47**	**6.78**
8	**12.11**	7.49	6.85

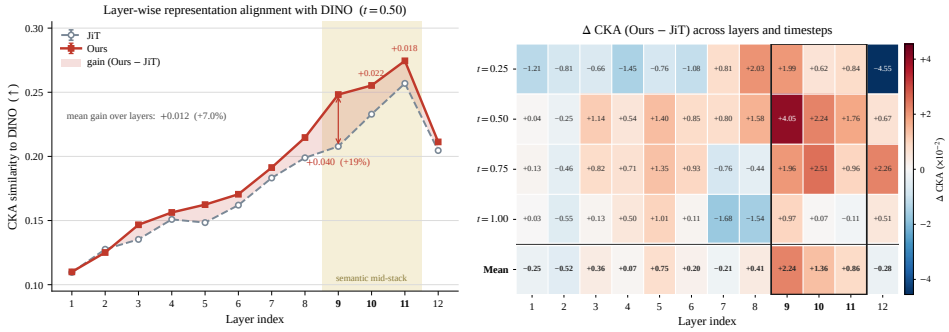

Figure 10: **DINO alignment analysis.** Left: Layer-wise unbiased linear CKA with DINO at $t = 0.5$, where PatchAFR improves alignment over JiT-B/32 across nearly all layers. Right: CKA difference ($\Delta\text{CKA} = \text{Ours} - \text{JiT}$). The gains vary across network depth and diffusion timesteps.

DINO Alignment Across Timesteps and Layers. The left panel of Figure 10 shows that at $t = 0.5$, PatchAFR improves DINO alignment across nearly all layers, with the largest gain in the middle-to-late blocks. The right figure extends this comparison across timesteps. The gains are strongest at $t = 0.5$ and $t = 0.75$, while $t = 0.25$ and $t = 1.0$ show weaker or mixed changes. Across layers, the improvements are concentrated in the middle-to-late blocks, especially blocks 9–11. Overall, PatchAFR mainly changes DINO alignment in the middle-to-late layers, with the strongest effect at intermediate timesteps.

B Further Analysis of Frequency-Based Methods

In this section, we further analyze several frequency-based methods originally developed for latent diffusion models. We focus on how they incorporate frequency information, how their formulations differ from PatchAFR, and what changes when these designs are transferred from latent to pixel-space diffusion.

Latent-to-Pixel Representation Mismatch in LWD. LWD (Sigillo et al., 2025) targets ultra-high-resolution latent diffusion by combining VAE regularization with frequency-guided supervision. It first fine-tunes the VAE with a scale-consistency objective to improve the spectral fidelity of the latent representation, and then applies a one-level Haar DWT to the noisy latent z_t. The energy of the high-frequency subbands is used to construct a wavelet energy map A_{wavelet} and a time-dependent



Figure 11: More qualitative results.

binary mask, which modulates the flow-matching loss to allocate more supervision to detail-rich regions. When adapting LWD to JiT, however, A_{wavelet} can no longer be computed in a VAE latent space and is instead extracted directly from the noisy pixel-space state.

This introduces two important differences. First, the wavelet energy used for masking is less stable under diffusion noise in pixel space. To quantify this effect, we compute A_{wavelet} from both the clean state and its noisy counterpart at different noise levels σ, and measure their Spearman correlation. We perform the same analysis in pixel space and in the FLUX VAE latent space. A higher correlation indicates that the noisy-state energy map better preserves the spatial ordering of the corresponding clean-state map. As shown in Table 11, the correlation in pixel space drops from 0.277 at $\sigma = 0.25$ to 0.049 at $\sigma = 0.5$, compared with 0.892 and 0.429 in the FLUX VAE latent. This suggests that the wavelet energy map used by LWD becomes substantially less reliable when computed directly from noisy pixel-space states. Second, VAE fine-tuning is an integral component of LWD rather than an auxiliary preprocessing step. The original ablation shows that both VAE scale consistency and wavelet masking contribute independently, while the full LWD achieves the strongest result. Since JiT does not use a VAE, our adaptation transfers the wavelet-masking mechanism but cannot reproduce this latent-space regularization, which may partly explain its weaker transfer performance.

Table 11: Stability of LWD wavelet energy maps under diffusion noise. We report the Spearman correlation between A_{wavelet} computed from the noisy and corresponding clean states.

Representation	$\sigma = 0.25$	$\sigma = 0.50$	$\sigma = 0.75$
Pixel space	0.277	0.049	0.006
FLUX VAE latent	**0.892**	**0.429**	**0.068**

Coarse Selection Granularity in FaPS. FaPS (Wang et al., 2026c) accelerates diffusion training through frequency-aware patch selection. A gating policy uses the frequency representation of each patch together with the diffusion timestep to select a subset of patches for training, thereby concentrating computation on the selected spatial units. Since no official implementation is publicly available, we implement FaPS following the published formulation at a high level, including frequency-based patch descriptors, timestep conditioning, stochastic patch selection, and policy-gradient optimization; we therefore treat our implementation as an adaptation rather than an exact reproduction.

An important consideration when transferring FaPS to JiT is the granularity of the selectable units. FaPS can be viewed as frequency-aware token sparsification, whose spatial resolution is determined by the underlying tokenization. JiT-B/32 represents a 256×256 image with only $8 \times 8 = 64$ image tokens, so a 50% keep ratio leaves only 32 selectable units, each covering a 32×32 region. We analyze the resulting granularity constraint using an oracle SNR upper bound. For each patch size, the oracle selects the best 50% subset under the same noisy observation, providing the maximum SNR achievable by that spatial partition rather than evaluating a particular gating policy. As shown in Table 12, reducing the selection resolution from 16×16 to 32×32 lowers retained signal power

Table 12: **Oracle SNR analysis of patch granularity.** Under the same 50% spatial budget, coarser patch selection retains less signal and yields a lower oracle SNR, with a 0.19 dB gap between 16×16 and 32×32. Spectral Forcing at $t = 0.5$ is included as a frequency-selective reference.

Method	Signal Retained (%)	Noise Retained (%)	SNR Gain (dB)
Spatial Patch 8×8	74.39 ± 7.76	48.89 ± 0.49	1.80 ± 0.44
Spatial Patch 16×16	72.72 ± 7.53	49.68 ± 0.24	1.63 ± 0.44
Spatial Patch 32×32	69.85 ± 7.18	49.91 ± 0.17	1.44 ± 0.44
Spectral Forcing	98.65 ± 1.70	38.34 ± 0.14	4.10 ± 0.08

from 72.72% to 69.85% and reduces the oracle SNR gain from 1.63 to 1.44 dB. The 0.19 dB oracle gap indicates that the coarse 8×8 token grid itself restricts the best achievable signal–noise trade-off under the same spatial budget. For reference, Spectral Forcing at $t = 0.5$ retains 98.65% of the signal while retaining 38.34% of the noise, showing that input suppression itself does not imply the same loss of useful signal; the limitation here is associated with coarse spatial selection rather than frequency-aware filtering in general.

C Implementation Details

Training Details. Unless otherwise specified, we train all models on ImageNet at 256×256 resolution for 200 epochs with a global batch size of 1024. Images are center-cropped, randomly flipped horizontally, and normalized to $[-1, 1]$. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and zero weight decay. Following JiT (Li & He, 2026), the learning rate is scaled as $5 \times 10^{-5} \times 1024/256 = 2 \times 10^{-4}$, with 5 epochs of linear warmup followed by a constant learning rate. Timesteps are sampled as $t = \sigma(s)$, where $s \sim \mathcal{N}(-0.8, 0.8^2)$, and the diffusion noise has unit variance. We use a class-label dropout probability of 0.1. The backbone predicts the clean image and is trained with JiT's velocity-space mean squared error, using $\max(1 - t, 0.05)$ in the conversion from image to velocity predictions. This conversion uses the original noisy state z_t; the routed state $\widetilde{z}_t$ is used as the backbone input. We maintain EMA weights with decay rates of 0.9999 and 0.9996 and use the former for evaluation. Both patch-16 and patch-32 models use 256×256 images during training and evaluation.

Testing Details. We evaluate FID using 50K generated samples against the ImageNet 256×256 reference statistics, with 50 samples per class across the 1,000 classes. Unless otherwise specified, we use 50-step Heun sampling on a uniform time grid from $t = 0$ to $t = 1$, with an Euler update for the final step, a generation batch size of 128 per GPU, and sampling seed 17. The CFG scale is 4.0 over $t \in (0.1, 1.0)$ and 1.0 outside this interval. Routing features and gates are recomputed at every denoiser evaluation. Additional results under different CFG scales and sampling seeds are reported in Sec. A.

Frequency Bands and Local Energy. We apply an orthonormal 2D DCT-II independently to each RGB channel over the full image. For an $H \times W$ image, the radial coordinate of coefficient (u, v) is $r_{u,v} = \sqrt{u^2 + v^2}/\sqrt{(H-1)^2 + (W-1)^2}$. We use four fixed empirical bands with boundaries $[0,\ 0.0625,\ 0.140625,\ 0.296875,\ 1]$; an internal boundary belongs to the higher-frequency band. Each band is reconstructed at full resolution before partitioning into non-overlapping patches matching the backbone patch size, giving 64 routing locations for patch-32 and 256 for patch-16. Local band energy is the mean squared response over the patch's spatial positions and RGB channels. The unit-noise energy $N_{p,b}$ is precomputed analytically from the squared DCT basis functions and band masks, then averaged over the same patch. It is fixed throughout training and inference. All frequency-routing computations use float32.

Router Features and Architecture. For each patch and band, we form the signed residual $R_{p,b} = E^{\mathrm{raw}}_{p,b}(t) - (1-t)^2 N_{p,b}$ and transform it as $\mathrm{sign}(R_{p,b}) \log(1 + |R_{p,b}|)$. This preserves negative residuals while compressing their dynamic range. We also use the auxiliary feature defined in Sec. 4.2, computed from $\widehat{S}_{p,b}(t) = [R_{p,b}]_+/(t^2 + \eta)$ and clipped to $[-1, 1]$, with $\eta = 10^{-4}$ in both

denominators. The scalar timestep is embedded by a $1 \to 16 \to 16$ MLP with SiLU between its linear layers. Concatenating this embedding with the four residual features and four auxiliary features yields a 24-dimensional input to a $24 \to 64 \to 4$ MLP, also with SiLU between its linear layers. Router parameters are shared across all spatial locations. All router linear layers use Xavier-uniform initialization with zero biases, and the router is optimized jointly with the backbone using the same learning rate and schedule.

Frequency Budget and Gate Projection. We set the reference budget $q(t)$ to the mean value of the linear Spectral Forcing mask (Fan et al., 2026) over the full DCT grid, using cutoff $0.05 + 0.95t$ and sharpness 30. For each sample, a single offset δ is shared across all patches and bands in Eq. 8 and chosen to satisfy

$$\frac{1}{N_{\text{patch}}} \sum_{p=1}^{N_{\text{patch}}} \sum_{b=1}^{4} w_b g_{p,b} = q(t), \qquad w_b = \frac{|\mathcal{B}_b|}{HW}, \tag{10}$$

where N_{patch} is the number of routing locations and $|\mathcal{B}_b|$ counts the DCT coefficients in band b per channel. The constraint therefore matches the coefficient-count-weighted mean gate amplitude to the reference budget. We use temperature $T = 1.0$, solve for δ with 48 bisection iterations, and compute gradients by implicit differentiation of the budget constraint.

Combination with PixelREPA. For JiT-B/16, we combine PatchAFR with PixelREPA (Shin et al., 2026) using a frozen DINOv2-B teacher with patch size 16. The teacher extracts 768-dimensional patch features from clean images on the same 16×16 grid as the backbone. Features after the fourth JiT block are passed to a two-block, 12-head Masked Transformer Adapter with a token masking probability of 0.2. The adapted features are aligned with the teacher features using a mean cosine-distance loss weighted by 0.1. The teacher and adapter are used only during training.

Ablation Settings. The timestep-only router receives only the timestep embedding and produces the same gates at every spatial location. The raw-energy and noise-corrected-energy variants use $\log(1 + E^{\text{raw}}_{p,b})$ and $\log(1 + \widehat{S}_{p,b})$, respectively, together with the timestep embedding. The residual-only variant uses the signed-log residual features without the auxiliary feature; the full router uses both. These variants retain the same router hidden width, temperature, and budget projection. For the frequency-band ablation, we compare uniform boundaries $[0,\ 0.25,\ 0.50,\ 0.75,\ 1]$ with the empirical boundaries specified above, keeping the router features and other settings fixed.

D Derivation of Global and Local Recoverability

We first reproduce the global recoverability argument of Spectral Forcing (Fan et al., 2026). Let X_k denote a clean coefficient at radial frequency k, with power $\mathbb{E}|X_k|^2 = P_x(k)$, and let $\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Rectified flow observes

$$Z_{t,k} = tX_k + (1-t)\epsilon_k.$$

Spectral Forcing compares clean-image power with the diffusion noise floor:

$$\text{DNR}_{\text{SF}}(k,t) = \frac{P_x(k)}{(1-t)^2\sigma_\epsilon^2}.$$

Under the power-law model $P_x(k) = Ck^{-\alpha}$, the level set $\text{DNR}_{\text{SF}}(k,t) = \tau_{\text{SF}}$ gives

$$\frac{Ck^{-\alpha}}{(1-t)^2\sigma_\epsilon^2} = \tau_{\text{SF}} \quad \Longleftrightarrow \quad k^\star_{\text{SF}}(t;\tau_{\text{SF}}) = \left(\frac{C}{\tau_{\text{SF}}\sigma_\epsilon^2(1-t)^2}\right)^{1/\alpha}.$$

For unit-variance noise,

$$k^\star_{\text{SF}}(t;\tau_{\text{SF}}) \propto (1-t)^{-2/\alpha}.$$

We next derive the local criterion directly from the rectified-flow observation. Recall that $A_{p,b} = R_p\Pi_b$. Conditioning on a fixed clean image x, the signal energy of patch p in band b is

$$\frac{1}{d}\,\|tA_{p,b}x\|_2^2 = t^2 S_{p,b}(x),$$

while the expected noise energy is

$$\frac{1}{d}\mathbb{E}_\epsilon \left\|(1-t)A_{p,b}\epsilon\right\|_2^2 = (1-t)^2 N_{p,b}.$$

Their ratio gives the local observation SNR $\rho_{p,b}(t;x)$ in Eq. (5).

E Proof and Variance of the Noise-Corrected Estimator

Fix a clean image $x \in \mathbb{R}^n$, a patch p, a band b, and $t \in (0,1)$. Let $A = A_{p,b} = R_p \Pi_b$, where Π_b is a fixed orthogonal projector and R_p extracts $d \geq 1$ distinct coordinates, so that $R_p R_p^\top = I_d$. The operators are fixed independently of the noise realization. Throughout this section,

$$z_t = tx + (1-t)\epsilon, \qquad \epsilon \mid x \sim \mathcal{N}(0, I_n).$$

Write

$$C = AA^\top, \qquad S = S_{p,b}(x) = \frac{\|Ax\|_2^2}{d}, \qquad N = N_{p,b} = \frac{\operatorname{tr}(C)}{d}.$$

Raw energy and unbiased correction. Define the observed raw energy by

$$E_{p,b}^{\mathrm{raw}}(t) = \frac{\|Az_t\|_2^2}{d}. \tag{11}$$

Expanding the squared norm gives

$$\|Az_t\|_2^2 = t^2\|Ax\|_2^2 + (1-t)^2\|A\epsilon\|_2^2 + 2t(1-t)(Ax)^\top A\epsilon.$$

Since

$$\mathbb{E}[\|A\epsilon\|_2^2 \mid x] = \operatorname{tr}(A^\top A) = dN, \qquad \mathbb{E}[(Ax)^\top A\epsilon \mid x] = 0,$$

we obtain

$$\mathbb{E}[E_{p,b}^{\mathrm{raw}}(t) \mid x] = t^2 S + (1-t)^2 N. \tag{12}$$

Consequently, the noise-corrected estimator

$$\widetilde{S} = \widetilde{S}_{p,b}(t) = \frac{E_{p,b}^{\mathrm{raw}}(t) - (1-t)^2 N}{t^2} \tag{13}$$

satisfies $\mathbb{E}[\widetilde{S} \mid x] = S$.

Exact conditional variance. Conditional on x,

$$Az_t \sim \mathcal{N}(\mu, \Sigma), \qquad \mu = tAx, \qquad \Sigma = (1-t)^2 C.$$

For a Gaussian vector $y \sim \mathcal{N}(\mu, \Sigma)$, including the case of singular Σ,

$$\operatorname{Var}(\|y\|_2^2) = 2\operatorname{tr}(\Sigma^2) + 4\mu^\top \Sigma \mu.$$

The subtracted noise floor is deterministic, and hence

$$\begin{aligned} V := \operatorname{Var}(\widetilde{S} \mid x) &= \frac{1}{t^4}\operatorname{Var}(E_{p,b}^{\mathrm{raw}}(t) \mid x) \\ &= \frac{2(1-t)^4}{d^2 t^4}\operatorname{tr}(C^2) + \frac{4(1-t)^2}{d^2 t^2}(Ax)^\top C(Ax). \end{aligned} \tag{14}$$

A conditional variance bound. Using $\Pi_b^2 = \Pi_b = \Pi_b^\top$ and $0 \preceq \Pi_b \preceq I_n$, we have

$$0 \preceq C = R_p \Pi_b R_p^\top \preceq R_p R_p^\top = I_d.$$

Therefore,

$$\operatorname{tr}(C^2) \leq \operatorname{tr}(C) = dN, \qquad (Ax)^\top C(Ax) \leq \|Ax\|_2^2 = dS.$$

Substitution into (14) yields

$$V \leq \frac{2(1-t)^4 N}{dt^4} + \frac{4(1-t)^2 S}{dt^2}. \tag{15}$$

This is an upper bound; (14) gives the exact variance. Noise-floor subtraction removes the expected noise contribution, but not its realization-dependent fluctuations. For $A \neq 0$, the noise-only term in the exact variance grows as t^{-4} as $t \downarrow 0$.

Scope of the moment assumptions. The unbiasedness calculation requires only $\mathbb{E}[\epsilon \mid x] = 0$ and $\mathrm{Cov}(\epsilon \mid x) = I_n$. The exact variance above is established under the stated Gaussian model. For general non-Gaussian noise, third- and fourth-order moments may contribute additional terms. The simplified bound also uses $C \preceq I_d$; this condition must be checked if the patch extractor is changed.

Positive-part clipping and stabilization. Let $[u]_+ = \max(u, 0)$. For every $S \geq 0$ and $u \in \mathbb{R}$,

$$([u]_+ - S)^2 \leq (u - S)^2. \tag{16}$$

Indeed, equality holds for $u \geq 0$; for $u < 0$, the difference between the right- and left-hand sides is $u^2 - 2uS \geq 0$. Thus, clipping cannot increase squared estimation error, and

$$\mathbb{E}[([\widetilde{S}]_+ - S)^2 \mid x] \leq V.$$

For a prescribed $\eta \geq 0$ that does not depend on the noise realization, define

$$\begin{aligned} \widehat{S} = \widehat{S}_{p,b}(t) &= \frac{[E^{\mathrm{raw}}_{p,b}(t) - (1-t)^2 N]_+}{t^2 + \eta} \\ &= a[\widetilde{S}]_+, \qquad a = \frac{t^2}{t^2 + \eta} \in (0, 1]. \end{aligned} \tag{17}$$

Since $a[\widetilde{S}]_+ = [a\widetilde{S}]_+$, (16) implies

$$(\widehat{S} - S)^2 \leq (a\widetilde{S} - S)^2.$$

Taking conditional expectations and using $\mathbb{E}[\widetilde{S} \mid x] = S$ gives

$$\begin{aligned} \mathrm{MSE}(\widehat{S} \mid x) &:= \mathbb{E}[(\widehat{S} - S)^2 \mid x] \\ &\leq a^2 V + (1-a)^2 S^2 \\ &\leq \frac{2(1-t)^4 N/d + 4t^2(1-t)^2 S/d + \eta^2 S^2}{(t^2 + \eta)^2}. \end{aligned} \tag{18}$$

For fixed $\eta > 0$, this bound remains finite as $t \downarrow 0$. It separates the attenuated variance contribution from the shrinkage error. Clipping and stabilization do not preserve unbiasedness in general; the exact unbiasedness statement applies to $\widetilde{S}$. Unlike clipping alone, stabilization does not guarantee a smaller MSE than the unregularized estimator. A finite noise-endpoint bound also does not imply that the clipped output vanishes when $S = 0$.