Spectral Progressive Diffusion for Efficient Image and Video Generation

Howard Xiao Brian Chao Lior Yariv Gordon Wetzstein
Stanford University
https://howardxiao.ca/speed/

Abstract

Diffusion models have been shown to implicitly generate visual content autoregressively in the frequency domain, where low-frequency components are generated earlier in the denoising process while high-frequency details emerge only in later timesteps. This structure offers a natural opportunity for efficient generation, as high-resolution computation on noise-dominated frequencies is largely redundant. We propose *Spectral Progressive Diffusion*, a general framework that progressively grows resolution along the denoising trajectory of pretrained diffusion models. To this end, we develop a spectral noise expansion mechanism and derive an optimal resolution schedule from the model's power spectrum. Our framework supports training-free acceleration and a novel fine-tuning recipe that further improves efficiency and quality. We demonstrate significant speedups on state-of-the-art pretrained image and video generation models while preserving visual quality.

1 Introduction

Computational demands for visual generative models are increasing rapidly as image resolutions and video sequence lengths continue to grow. This trend reveals a fundamental scaling crisis: while scaling model and data sizes consistently yields better generation quality, the underlying cost of self-attention in Diffusion Transformers (DiTs) [62] scales quadratically with the number of generated tokens. This creates a growing conflict between the desire for high-fidelity, long-duration content and the compute cost of existing architectures. A shift toward token-efficient representations offers a flexible and broadly compatible path for further scaling image and video generation.

The seminal observation by Dieleman [9] revealed that diffusion models implicitly learn to generate visual content autoregressively in the frequency domain, where low-frequency components are generated earlier in the denoising process while high-frequency details emerge only in later timesteps. While this observation has motivated a variety of frequency-domain designs for diffusion models [63, 6, 14, 43], they either do not support existing pretrained models or provide limited efficiency gains. A natural way to exploit spectral autoregression is through progressive resolution growth, since the frequency content representable by a signal is intrinsically tied to its spatial resolution. However, prior progressive resolution approaches [33, 58, 32, 74] either require significant modifications to the model architecture or rely heavily on heuristics for when and how to upsample, limiting their compatibility with state-of-the-art pretrained models [5, 40, 80, 87].

In this paper, we propose *Spectral Progressive Diffusion*: a general framework for progressive-resolution generation, guided by the spectral structure of the denoising process. Motivated by the spectral autoregression property of diffusion models [9], we expand the resolution only at the timesteps where high-frequency content begins to emerge from noise (see Fig. 1). This keeps early denoising steps in a reduced token space and avoids redundant computation on noise-dominated frequencies. To achieve this, we introduce a spectral noise expansion mechanism that uses a spectral transformation to inject high-frequency noise at the correct level while preserving the partially-

Preprint.

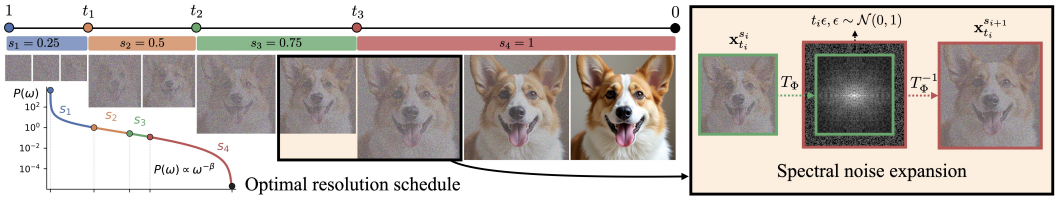

Figure 1: **Spectral Progressive Diffusion.** We progressively grow the resolution along the denoising trajectory using an *optimal resolution schedule* derived from the spectral power of pretrained models **(left)**. At each scheduled transition, our *spectral noise expansion* mechanism **(right)** injects high-frequency noise at the correct level while preserving the partially-denoised low-frequency content.

denoised low-frequency content. We further derive optimal resolution transition times directly from the model's power spectrum, determined by a single error-tolerance hyperparameter.

Our framework applies directly to state-of-the-art pretrained diffusion and flow-matching models without architectural modifications. We demonstrate the flexibility of our framework across three visual generation modalities: latent-space image generation [39, 5], pixel-space image generation [55], and latent-space video generation [80]. In a training-free setting, our method delivers immediate speedups on existing pretrained models. We additionally introduce a fine-tuning recipe that further improves efficiency and quality, a direction previously unexplored for progressive-resolution generation. Extensive experiments show speedups of up to $7\times$ on image generation and $2.5\times$ on video generation, outperforming prior spatial acceleration methods in both runtime and visual fidelity.

To summarize:

- We propose a progressive-resolution generation framework guided by the spectral autoregression of diffusion models, that applies directly to pretrained models and improves efficiency.
- We introduce a principled spectral noise expansion mechanism and derive optimal resolution transition times from the model's power spectrum.
- We demonstrate our framework on both image and video generation models, supporting both training-free inference and fine-tuning while maintaining high generation quality.

2 Related Work

Spectral-domain designs in diffusion models. The seminal discussion by Dieleman [9] on the spectral autoregression property of diffusion has inspired many spectral-domain designs in diffusion models. Most of these models [63, 21, 64, 45, 29, 11, 59] require training from scratch, making them incompatible with existing pretrained image and video generation models [40, 5, 75, 87]. Other methods use frequency-aware losses [71, 6], autoencoder regularization [72], or noise schedules [41, 3, 13, 14] to improve the generation quality of existing models but do not improve efficiency. Recent training-free methods use spectral-domain designs for efficient high-resolution generation [43, 37, 79]. However, as they require fully denoised lower-resolution images, their overall efficiency gains are limited. Our method leverages similar insights of spectral autoregression but progressively grows resolution within a single denoising trajectory, achieving greater efficiency gains while maintaining compatibility with pretrained models.

Progressive resolution for efficient generation. Progressive resolution approaches have been proposed to improve generation efficiency by reducing token counts early in the denoising process. However, most existing methods [26, 60, 76, 33, 19, 68, 94, 58] require training from scratch due to model-specific architectures, limiting their compatibility with pretrained models. Current training-free [78, 32, 10, 93, 65] and fine-tuning progressive resolution approaches [20, 74] either require significant heuristics-based hyperparameter tuning or specialized modules that limit their effectiveness and their compatibility with adaptation methods such as LoRA [27]. Progressive resolution generation has also been explored in visual autoregressive models [77, 23, 66, 30], where next-token prediction is reformulated as next-scale prediction. Our method uniquely leverages the spectral autoregression property of diffusion models to design a principled progressive resolution growing mechanism for efficient image and video generation using minimal hyperparameters. Furthermore, our method is fully compatible with current pretrained models for both training-free inference and fine-tuning.

Other methods for efficient generation. Aside from progressive resolution pipelines, a wide variety of acceleration methods have been proposed for efficient generation. Model distillation approaches reduce the number of denoising steps by training a student model to approximate the outputs of a pretrained teacher [89, 88, 90, 53, 81, 69]. Feature caching approaches reuse features across denoising timesteps to reduce redundant computation [48, 95, 49, 54]. Token merging techniques identify redundant tokens and merge similar tokens based on predefined heuristics [52, 24, 4, 36, 84, 42, 16]. Sparse attention mechanisms identify sparse patterns in the attention maps and use block-wise calculations to improve efficiency [44, 85, 91, 35, 83, 8, 2]. In this paper, we present an orthogonal and complementary progressive resolution generation approach that accelerates both image and video generation through spectral-domain transformations.

3 Preliminaries

3.1 Diffusion Models and Flow Matching

Diffusion models [25, 73] define a generative process that gradually transforms samples from a simple prior (e.g. Gaussian) to data samples via a learned reverse-time process. Flow Matching [51, 47] reformulates the diffusion process as a continuous-time optimal transport problem and learns a velocity field that deterministically transforms noisy samples to data samples along straight paths.

Specifically, given a clean data point $\mathbf{x}_0$ sampled from a data distribution p_{data}, and a noise sample $\mathbf{x}_1 = \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, the noise-to-data path is a straight line defined by

$$\mathbf{x}_t \;=\; (1-t)\,\mathbf{x}_0 \;+\; t\,\boldsymbol{\epsilon}, \qquad t \in [0,1]. \tag{1}$$

A neural network $\mathbf{v}_\theta(\mathbf{x}_t, t)$ is trained to predict the target velocity $\dot{\mathbf{x}}_t = \boldsymbol{\epsilon} - \mathbf{x}_0$ with the following objective:

$$\mathcal{L}_{\text{flow}}(\theta) \;=\; \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\big[\,\|\mathbf{v}_\theta(\mathbf{x}_t,t) - (\boldsymbol{\epsilon}-\mathbf{x}_0)\|^2\,\big]. \tag{2}$$

A network with sufficient capacity that is trained on enough data converges to the *Bayes-optimal velocity predictor* $\mathbf{v}_\theta(\mathbf{x}_t, t) = \mathbf{v}^*(\mathbf{x}_t, t)$ [51, 47].

During inference, data samples are generated through sampling from the prior $\mathbf{x}_1 \sim \mathcal{N}(0, I)$ and solving the probability-flow Ordinary Differential Equation (ODE) $\dot{\mathbf{x}}_t = \mathbf{v}_\theta(\mathbf{x}_t, t)$. Due to its faster training and stable convergence, recent image and video generative models [40, 5, 12, 75] have increasingly adopted the flow matching paradigm.

Most state-of-the-art diffusion and flow matching models adopt the Diffusion Transformer (DiT) architecture [62], which processes tokens through a sequence of self-attention and Multilayer Perceptron (MLP)-based operations. This motivates our approach to reduce the number of tokens processed along the denoising trajectory using the spectral autoregression property of diffusion models [9].

3.2 Spectral Autoregression in Diffusion Models

Natural images exhibit a characteristic power-law decay [67], which is the foundation of the spectral autoregression property of diffusion models [9]. Specifically, let $\Phi = \{\phi_\omega\}_{\omega\in\Omega}$ be a spectral basis indexed by frequency $\omega \in \Omega$ (e.g., Fourier [17], discrete cosine [1], or Haar wavelet [22]), and denote the associated spectral transformation by T_Φ. We define the per-frequency signal power, or *power spectrum*, of a data distribution p_{data} as

$$P_\omega \;:=\; \mathbb{E}_{\mathbf{x}_0}\Big[\,\big|x_0^{(\omega)}\big|^2\,\Big], \quad \text{where } x_0^{(\omega)} := \langle \mathbf{x}_0, \phi_\omega\rangle \text{ under } T_\Phi. \tag{3}$$

When p_{data} represents natural images, P_ω is known to exhibit a power-law decay, that is

$$P_\omega \;\propto\; |\omega|^{-\beta}, \tag{4}$$

with the exponent term typically in the range $\beta \in [2, 3]$ [9, 14]. We validated similar trends in both image and video latent representations of modern latent-space diffusion models (see Fig. 2).

The power-law decay of P_ω in Eq. (4) implies that high frequencies carry substantially weaker signal than low frequencies. Since the Gaussian noise $\boldsymbol{\epsilon}$ added during the forward process has a flat, i.e. frequency-independent, power spectrum, higher frequencies are dominated by noise early in the denoising trajectory. In this work, we leverage this spectral autoregression property to progressively grow resolution within a single denoising trajectory via spectral domain transformations for efficient image and video generation.

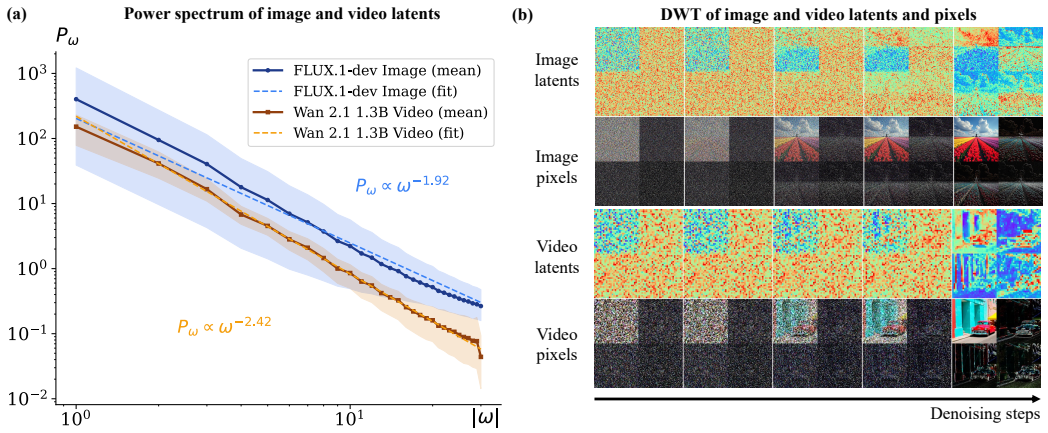

Figure 2: **Diffusion process in the spectral domain.** Latent power spectra in both image and video models decay rapidly with frequency (Fig. **(a)**), consistent with natural images. Diffusion exhibits a frequency-domain autoregressive structure (Fig. **(b)**) due to the aforementioned property: low frequencies emerge early in the denoising process, while high frequencies remain noise-dominated.

4 Spectral Progressive Diffusion

In this section, we present a spectral-transformation-based progressive generation framework that enables arbitrary resolution transitions within a single denoising trajectory for pretrained image and video generation models (Sec. 4.1). We then derive an optimal resolution transition schedule based on power spectrum analysis and resolution-dependent frequency limits (Sec. 4.2). Finally, we introduce a principled fine-tuning strategy to further improve both generation quality and efficiency (Sec. 4.3).

4.1 Training-Free Inference with Spectral Noise Expansion

Motivated by the spectral autoregression property of diffusion (Sec. 3.2), we progressively increase image resolution by injecting higher-frequency components along the denoising trajectory using a spectral transformation. Our framework applies natively to existing pretrained image and video generation models, without additional custom modules or architectural modifications.

Given an orthonormal spectral basis Φ, let T_Φ and T_Φ^{-1} denote the forward and inverse spectral transformations. We define S progressive resolution scales $s_{1:S}$, where $0 < s_1 < s_2 < \cdots < s_S = 1$, and their corresponding resolution transition times $t_{1:S-1}$, with $1 > t_1 > \cdots > t_{S-1} > 0$, that are matched to the noise schedules of pretrained models. We denote the final full-resolution state as $\mathbf{x}_0^{s_S} \in \mathbb{R}^{C\times T\times H\times W}$, and assume stage i runs at spatial resolution (s_iH, s_iW) for $t \in (t_i,\, t_{i-1}]$.

At each transition time t_i, we expand the current low-resolution state $\mathbf{x}_{t_i}^{s_i} \in \mathbb{R}^{C\times T\times (s_iH)\times (s_iW)}$ to the next resolution state $\mathbf{x}_{t_i}^{s_{i+1}} \in \mathbb{R}^{C\times T\times (s_{i+1}H)\times (s_{i+1}W)}$ using *spectral noise expansion* and timestep alignment (see Fig. 1).

Spectral noise expansion. Given a low-resolution $\mathbf{x}_{t_i}^{s_i}$ at transition time t_i, we expand its resolution to the next scale s_{i+1} through the following steps:

(i) Compute the spectrum $\boldsymbol{\xi}_{t_i}^{s_i} = T_\Phi(\mathbf{x}_{t_i}^{s_i})$ supported on the frequency set Ω_{s_i} representable at scale s_i.

(ii) Embed $\boldsymbol{\xi}_{t_i}^{s_i}$ in the lower-frequency part of the spectrum for $\Omega_{s_{i+1}}$ and fill the slots $\Omega_{s_{i+1}} \setminus \Omega_{s_i}$ with $t_i\epsilon$, $\epsilon \sim \mathcal{N}(0,1)$ to get $\boldsymbol{\xi}_{t_i}^{s_{i+1}}$.

(iii) Convert the spectrum back to spatial domain $\mathbf{x}_{t_i}^{s_{i+1}}$ at resolution $(s_{i+1}H, s_{i+1}W)$ via $\mathbf{x}_{t_i}^{s_{i+1}} = T_\Phi^{-1}(\boldsymbol{\xi}_{t_i}^{s_{i+1}})$.

Steps (i)–(iii) expand the resolution scale from s_i to s_{i+1} for $\mathbf{x}_{t_i}^{s_{i+1}}$ by injecting high frequencies at the correct noise level while preserving the partially-denoised low-frequency content of $\mathbf{x}_{t_i}^{s_i}$.

Method	Speedup (s) ↑	TFLOPs ↓	Overall	Image quality		Text alignment	
			ImageReward ↑	CLIP-IQA ↑	NIQE ↓	T2I-Comp. ↑	GenEval ↑
FLUX (50 steps)	1.00×	2991.01	**1.095**	0.707	6.75	**0.634**	**0.698**
RALU [32]	1.58×	**1749.94**	1.028	0.712	**6.07**	0.613	0.648
Ours ($S = 2$)	**1.66**×	1755.22	1.049	**0.719**	6.43	0.617	0.654
FLUX (10 steps)	4.84×	610.02	0.981	0.679	6.93	0.618	0.647
Bottleneck [78]	4.67×	571.23	0.889	0.661	9.16	0.620	**0.687**
RALU [32]	4.98×	540.47	1.022	0.700	**6.43**	**0.626**	0.652
Ours ($S = 2$)	5.77×	500.34	**1.059**	0.696	6.69	0.624	0.655
Ours ($S = 3$)	**6.09**×	**469.15**	1.042	**0.701**	6.53	0.623	0.637
FLUX (7 steps)	6.62×	431.45	0.920	0.660	8.25	0.594	0.583
Bottleneck [78]	6.64×	431.52	0.792	0.631	8.71	0.605	0.672
RALU [32]	6.69×	426.01	0.999	0.681	6.87	**0.633**	**0.682**
Ours ($S = 2$)	6.78×	427.03	**1.039**	0.689	6.78	0.620	0.667
Ours ($S = 3$)	**7.09**×	**406.24**	1.015	**0.694**	**5.99**	0.627	0.637

Table 1: **Training-free quantitative comparisons on FLUX.1-dev [39].** Image resolution is 1024^2 and baseline rows are copied from Table 1 in RALU [32]. We group results based on $1\times, 5\times, 7\times$ speedup settings where progressive resolution approaches are applied in combination with reducing denoising steps. Our method improves the speed–quality tradeoff across all speedup settings compared to baseline approaches. Further evaluation details and comparisons are shown in Appendix Sec. D.1.

Timestep alignment. As Φ is orthonormal, the high-frequency noise padding in $\Omega_{s_{i+1}} \setminus \Omega_{s_i}$ raises the overall noise level of the enlarged $\mathbf{x}_{t_i}^{s_{i+1}}$, which makes it no longer correspond to the original timestep t_i. We therefore rescale the inverse-transformed output and re-index the timestep by:

$$\tilde{\mathbf{x}}_{\tilde{t}_i}^{s_{i+1}} := \frac{s_{i+1}/s_i}{1 + ((s_{i+1}/s_i) - 1)t_i} \cdot \mathbf{x}_{t_i}^{s_{i+1}}, \tag{5}$$

where

$$\tilde{t}_i := \frac{(s_{i+1}/s_i)\, t_i}{1 + ((s_{i+1}/s_i) - 1)t_i}. \tag{6}$$

Eq. 5 and Eq. 6 give us the aligned state $\tilde{\mathbf{x}}_{\tilde{t}_i}^{s_{i+1}}$ at the aligned timestep $\tilde{t}_i$. The integration of the probability-flow ODE then resumes at resolution $(s_{i+1}H, s_{i+1}W)$ until the next transition time t_{i+1}. The full derivation of Eqs. (5)–(6) is provided in Appendix A.2. In practice, S and $s_{1:S}$ are chosen to align with the multi-resolution training distribution of pretrained generative models. By default, we choose T_Φ as the Discrete Cosine Transform (DCT) [1] with discrete cosine basis Φ. Alternative spectral transforms (such as Discrete Wavelet Transform (DWT) and Fourier Transform (FFT)) are compared and evaluated in Sec. 5.4.

4.2 Optimal Resolution Schedule

Section 4.1 described our progressive resolution transition mechanism during the denoising process given a predetermined resolution transition schedule $t_{1:S-1}$. In practice, while increasing the number of resolution scales S improves token efficiency, it introduces additional transition times as hyperparameters to tune. As a result, existing progressive resolution approaches typically limit $S = 2$ or leave $t_{1:S-1}$ as a brittle, model-specific design choice [78, 32]. In contrast, our method supports an arbitrary number of resolutions S and arbitrary resolution scales $s_{1:S}$. In the following section, we derive a *δ-optimal resolution schedule* for pretrained flow matching models that determines the optimal transition times $t^*_{1:S-1}$ from a single error threshold parameter δ independent of S.

Spectral-domain flow matching. The spatial-domain flow matching forward process defined in Eq. (1) can be written in the spectral domain under transformation T_Φ with orthonormal basis $\Phi = \{\phi_\omega\}_{\omega\in\Omega}$:

$$x_t^{(\omega)} \;=\; (1-t)\, x_0^{(\omega)} \;+\; t\, \epsilon^{(\omega)}, \qquad \omega \in \Omega, \tag{7}$$

where $\epsilon^{(\omega)} := \langle \boldsymbol{\epsilon}, \phi_\omega \rangle$. By orthonormality of Φ, $\epsilon^{(\omega)} \sim \mathcal{N}(0, 1)$ is i.i.d. Gaussian noise for each $\omega \in \Omega$. We denote the per-frequency component of the Bayes-optimal velocity predictor described in Sec. 3.1 by $v^{*(\omega)}(\mathbf{x}_t, t)$. We then determine when a spectral component ω is noise-dominated in Proposition 1 and link this to spatial resolution to derive the δ-optimal resolution schedule in Proposition 2.

Method	Speedup (s) ↑	TFLOPs ↓	ImageReward ↑	CLIP-IQA ↑	NIQE ↓	T2I-Comp. ↑	GenEval ↑
Z-Image (50 steps)	1.00×	4941.23	0.965	0.700	5.41	0.731	0.745
Ours (TF, $S = 2$)	1.65×	3132.03	0.904	0.688	5.87	0.658	0.730
Ours (LoRA, $S = 2$)	1.65×	3132.03	**0.982**	**0.699**	**5.72**	**0.725**	**0.731**
Ours (TF, $S = 3$)	1.74×	2871.09	0.875	0.690	**5.59**	0.650	0.682
Ours (LoRA, $S = 3$)	1.74×	2871.09	**0.954**	**0.697**	5.75	**0.717**	**0.728**
PixelGen (25 steps)	1.00×	65.36	0.921	0.734	5.95	0.574	0.794
Ours (TF, $S = 2$)	1.60×	33.72	0.799	0.718	6.10	0.568	**0.782**
Ours (LoRA, $S = 2$)	1.55×	33.72	**0.913**	**0.728**	**5.87**	**0.580**	0.776

Table 2: **Fine-tuning quantitative comparisons on latent- (Z-Image [5]) and pixel-space image generation (PixelGen [55]).** Image resolution is 1024^2 for Z-Image and 512^2 for PixelGen. Across both latent- and pixel-space image generation, our fine-tuning method bridges the gap between model pretraining and progressive resolution inference, further improving generation quality while preserving efficiency gains. Further evaluation details and comparisons are shown in Appendix Sec. D.2.

Proposition 1 (Per-frequency δ-optimal activation time): Under the simplified modelling assumption $x_0^{(\omega)} \sim \mathcal{N}(0, P_\omega)$ with $P_\omega > 0$, for all $t \geq t_\omega$ and $\delta \in (0, 1)$, we have:

$$\mathbb{E}\Big[\big|v^{*(\omega)}(\mathbf{x}_t, t) - \epsilon^{(\omega)}\big|^2\Big] \;\leq\; \delta, \tag{8}$$

where t_ω is the δ-optimal activation time for ω:

$$t_\omega \;:=\; \frac{1}{1 + \sqrt{\dfrac{\delta}{P_\omega\,(1 + P_\omega - \delta)}}}. \tag{9}$$

Intuitively, for $t \geq t_\omega$, spectral component ω is noise-dominated as its optimal velocity predictions are approximately noise $\epsilon^{(\omega)}$. After t_ω, the signal power overcomes the error tolerance δ and structural information is recovered. The proof of Proposition 1 is given in Appendix Sec. A.3. We further include empirical evidence supporting Proposition 1 in Appendix Sec. B.

Proposition 2 (Per-resolution δ-optimal transition time): Under the setting of Proposition 1, and assuming P_ω is monotonically decreasing in $|\omega|$ (consistent with the power-law decay Eq. (4)), for any resolution scale s_i, s_{i+1}, the optimal transition time from scale s_i up to s_{i+1} is

$$t_i^* \;:=\; \min_{\omega \in \Omega_{s_i}} t_\omega \;=\; t_{\omega = s_i \cdot \omega_{\max}(H,W)}, \tag{10}$$

where Ω_{s_i} is the set of frequencies representable on the $(s_i H, s_i W)$ grid. The maximum representable frequency of the full-resolution spatial grid $\Omega_{s_S} = \omega_{\max}(H, W) = \min(H, W)/2$ is given by the Nyquist–Shannon sampling theorem [61, 70].

Proposition 2 connects frequency activation times with spatial resolution transitions through the Nyquist limit. It implies that high-resolution computation in early denoising steps is redundant as most representable frequencies ω are still noise-dominated, motivating our approach of Spectral Progressive Diffusion. We apply spectral noise expansion at t_i^* to progressively increase resolution precisely when finer details emerge from noise, thereby maximizing token efficiency. The hyperparameter δ is independent of S, making our method particularly robust to tuning compared to prior methods. The proof of Proposition 2 is deferred to Appendix A.4 and we analyze the effect of the number of resolution stages S and the tolerance δ in Sec. 5.4.

4.3 Spectral-transformation-based Fine-tuning

While Spectral Progressive Diffusion is compatible with pretrained image and video generation models in a training-free manner (Sec. 4.1), it implicitly assumes multi-resolution generation capability of pretrained models across scales $s_{1:S}$. Moreover, our optimal resolution schedule (Sec. 4.2) may not accurately reflect pretrained models' training dynamics, potentially introducing training-inference gaps. In this subsection, we outline a spectral-transformation-based fine-tuning approach that directly follows our Spectral Progressive Diffusion framework and our optimal resolution schedule. We include additional implementation details of our fine-tuning framework in Sec. 5.2 and Appendix Sec. D.2.

Method	Speedup (s) ↑	TFLOPs ↓	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Dynamic Degree ↑	Aesthetic Quality ↑	Image Quality ↑
WAN 2.1 (50)	1.00×	119292	0.9492	0.9621	0.9874	0.4800	0.5993	0.6133
WAN 2.1 (25)	2.00×	59646	0.9434	0.9597	**0.9879**	0.4250	0.5893	0.5706
Ours ($S = 2$)	**2.03**×	**57417**	**0.9462**	**0.9598**	0.9859	**0.4950**	**0.5975**	**0.6114**
Ours ($S = 3$)	2.54×	45953	0.9299	0.9499	0.9815	0.5700	0.5696	0.5990

Table 3: **Quantitative comparison on latent-space video generation on WAN 2.1 [75].** Video resolution is maintained at 720P. Our training-free approach demonstrates more than 2× speedup while maintaining high generation quality and outperforming the 25-step full-resolution baseline.

Resolution-specific velocity targets. We assume S progressive resolution scales $s_{1:S}$ and their corresponding δ-optimal resolution transition times $t_{1:S-1}$ from Sec. 4.2 (denoted here as t_i instead of t_i^* for simplicity). For resolution stage i, the model operates at scale s_i for $t \in (t_i, \tilde{t}_{i-1}]$, where $\tilde{t}_{i-1}$ is the aligned timestep obtained from Eq. (6) after expanding from s_{i-1} to s_i at transition time t_{i-1}. The state at the beginning of the stage i is hence the enlarged and timestep-aligned $\tilde{\mathbf{x}}^{s_i}_{\tilde{t}_{i-1}}$. The model is fine-tuned to follow a straight-line path from $\tilde{\mathbf{x}}^{s_i}_{\tilde{t}_{i-1}}$ to the standard flow-matching state at the next transition time:

$$\mathbf{x}^{s_i}_{t_i} = (1 - t_i)\mathbf{x}^{s_i}_0 + t_i \boldsymbol{\epsilon}^{s_i}. \tag{11}$$

where $\mathbf{x}^{s_i}_0$ denotes the clean data sample at scale s_i. The corresponding stage-specific velocity target is therefore:

$$\mathbf{v}^{s_i} = \frac{\tilde{\mathbf{x}}^{s_i}_{\tilde{t}_{i-1}} - \mathbf{x}^{s_i}_{t_i}}{\tilde{t}_{i-1} - t_i}. \tag{12}$$

Training and inference. At each training step we sample $\mathbf{x}_0 \sim p_{\text{data}}$ and $t \sim \mathcal{U}(0, 1)$. We assign t to the resolution stage i satisfying $t \in (t_i, t_{i-1}]$, and then sample $\boldsymbol{\epsilon}^{s_i} \sim \mathcal{N}(0, I)$ at scale s_i. We construct the stage input $\tilde{\mathbf{x}}^{s_i}_{\tilde{t}_{i-1}}$ by applying the same spectral noise expansion and timestep alignment used at inference, where newly introduced spectral coefficients are filled using $t_{i-1} \cdot T_\Phi(\boldsymbol{\epsilon}^{s_i})$. We use the same $\boldsymbol{\epsilon}^{s_i}$ to construct the endpoint $\mathbf{x}^{s_i}_{t_i}$ from Eq. (11), so that $\tilde{\mathbf{x}}^{s_i}_{\tilde{t}_{i-1}}$ and $\mathbf{x}^{s_i}_{t_i}$ are correlated through a shared noise realization. The training sample is the point on the straight path between them:

$$\mathbf{x}^{s_i}_t = \mathbf{x}^{s_i}_{t_i} + (t - t_i)\, \mathbf{v}^{s_i}, \tag{13}$$

with target velocity $\mathbf{v}^{s_i}$ from Eq. (12).

We fine-tune a pretrained model $\mathbf{v}_\theta$ by minimizing

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{t,\, \mathbf{x}_0,\, \boldsymbol{\epsilon}} \big[\,\| \,\mathbf{v}_\theta(\mathbf{x}^{s_i}_t, t) - \mathbf{v}^{s_i} \,\|^2 \,\big]. \tag{14}$$

At inference, we use the same procedure as in Sec. 4.1 with the fine-tuned model, using the same resolution schedule $t_{1:S-1}$ and transform T_Φ.

5 Experiments

5.1 Setup

We extensively evaluate our Spectral Progressive Diffusion framework on latent-space image generation with **FLUX.1-dev** [39] and **Z-Image** [5], covering both training-free inference acceleration and fine-tuning with LoRA [27]. We further demonstrate that our framework applies to pixel-space image generation with **PixelGen-XXL/16 T2I** [55] and latent-space video generation with **WAN 2.1-T2V-1.3B** [75]. Each pretrained backbone uses its native resolution and default model-specific inference schedule.

For latent- and pixel-space image generation, we follow RALU's [32] evaluation protocol, reporting ImageReward [86] for overall quality, CLIP-IQA [82] and NIQE [57] for image quality together with T2I-CompBench [28] and GenEval [18] for prompt alignment. For video generation, we report VBench [31] scores across its standard evaluation dimensions. We report total FLOPs integrated over the full denoising trajectory using the same convention as RALU and end-to-end normalized wall-clock speedup. We use $\delta = 0.01$ and Discrete Cosine Transform (DCT) as our spectral transformation T_Φ across the experiments and include ablation studies on δ, S, and T_Φ in Sec. 5.4. We include additional implementation and experimental details in Appendix Secs. D and E.

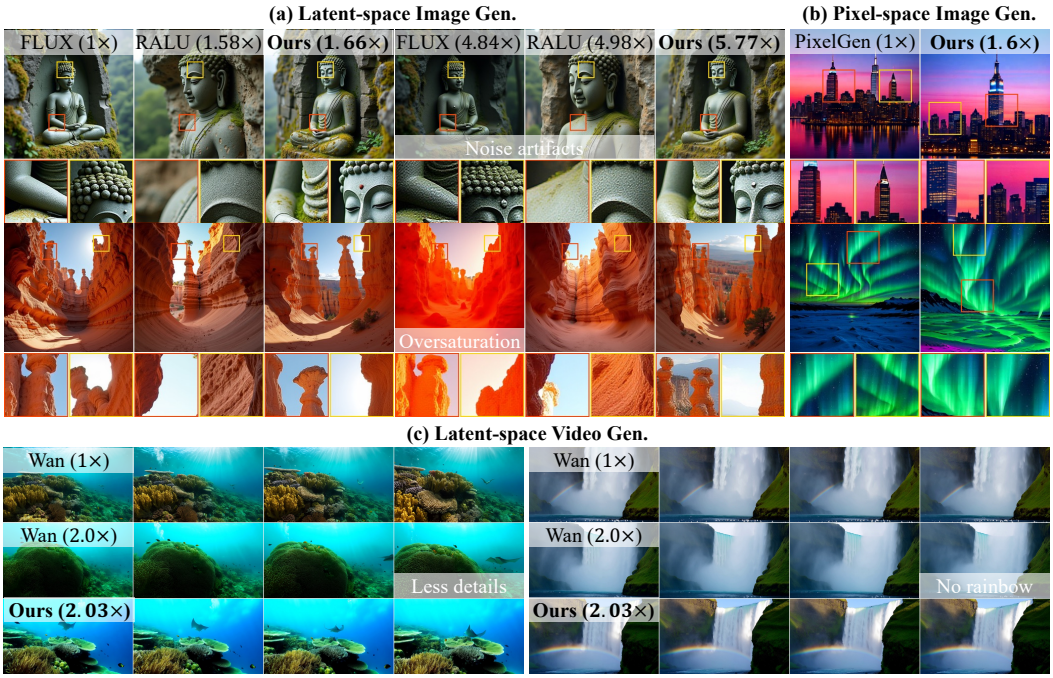

Figure 3: **Visual Generation Qualitative Comparisons.** For the main comparison of latent-space image generation, our method outperforms the state-of-the-art spatial acceleration method RALU [32] in both visual fidelity and inference speed. Across all evaluated modalities (latent/pixel-space image generation and latent-space video generation), we achieve substantial acceleration over standard high-resolution baselines while preserving generation quality.

5.2 Latent- and Pixel-Space Image Generation

Training-free acceleration. In Table 1 and Fig. 3, we compare our training-free acceleration approach (Sec. 4.1) on FLUX.1-dev with training-free acceleration baselines that implement progressive resolution growing, including Bottleneck Sampling [78] and RALU [32]. Additional FLUX.1-dev comparisons are included in Appendix Sec. D.1. Our approach consistently improves the speed-quality tradeoff over other methods across all three evaluated speedup regimes. Increasing $S = 2$ to $S = 3$ further improves efficiency, offering up to $7.36\times$ FLOPs speedup and $7.09\times$ wall clock speedup while maintaining high generation quality.

Spectral-transformation-based fine-tuning. In Table 2, we demonstrate that our spectral-transformation-based fine-tuning method (Sec. 4.3) further improves image quality and efficiency compared to training-free acceleration. Since FLUX.1-dev is guidance-distilled and its base model is not open-sourced, we select Z-Image [5] as the base model for our latent-space image fine-tuning experiments. All models are fine-tuned with LoRA [27] using rank 32 for 2000 iterations in both latent-space and pixel-space image generation experiments. Across both latent- and pixel-space image generation, our fine-tuning framework narrows the gap between training and inference, further improving generation quality while maintaining efficiency gains. We include additional implementation details and results in Appendix Sec. D.2.

5.3 Latent-space Video Generation

We further demonstrate the flexibility and robustness of our training-free acceleration approach on latent-space video generation (Table 3, Fig. 3). To our knowledge, there are no diffusion acceleration approaches that were validated on all three generation modalities presented in Secs. 5.2–5.3. We observe trends consistent with image generation, where our method achieves significant speedups while preserving generation quality. We include additional implementation details and results in Appendix Sec. E.

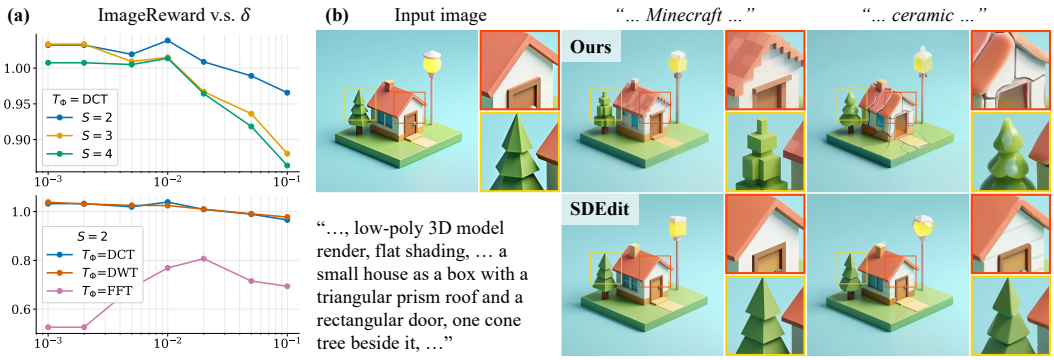

Figure 4: **(a): Ablation Studies on δ, S and T_Φ.** We observe a clear tradeoff between image quality and efficiency when varying δ and S as shown in the top plot. Across transforms, DCT achieves similar quality as DWT and outperforms FFT as shown in the bottom plot. **(b): Frequency-based Image Editing.** Our method demonstrates superior prompt alignment and geometric consistency compared to standard SDEdit-style spatial-domain editing under identical step counts. Our method successfully edits the texture and style of the input image, whereas SDEdit fails to do so.

5.4 Ablation Studies

We ablate our core design choices, including error threshold δ, resolution stages S, and spectral transformation T_Φ, on latent-space image generation using FLUX.1-dev. We show the speed-quality Pareto frontier in Fig. 4(a) for $\delta \in [0.001, 0.1]$, $S \in \{2, 3, 4\}$, and different spectral transformations T_Φ. Consistent with our theory, increasing δ or S further accelerates generation at the cost of gradual image quality loss. The Fourier Transform (FFT) tends to oversmooth results, whereas Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) achieve similar high-fidelity performance. Qualitative comparisons are given in Appendix Sec. F.

5.5 Frequency-based Image Editing

We demonstrate a unique *frequency-based* image editing capability enabled by our method. Specifically, given an input $\mathbf{x}_{\text{in}}$ to edit, we add noise to the low-frequency part of $T_\Phi(\mathbf{x}_{\text{in}})$, perform spectral noise expansion and timestep alignment (Sec. 4.1), and then resume denoising from the timestep t corresponding to the resolution schedule with the editing prompt. Fig. 4(b) shows our method achieves substantially improved prompt alignment and geometric consistency compared to a spatial-domain SDEdit-style baseline [56]. Additional details and editing results are in Appendix Sec. G.

6 Discussion

In this work, we introduce Spectral Progressive Diffusion, a generation framework that leverages spectral autoregression in diffusion models to match computation to frequencies where signal emerges from noise. Through a principled spectral noise expansion mechanism and an optimal resolution schedule derived from the model's power spectrum, we enable progressive resolution growth along the denoising trajectory of pre-trained image and video models, achieving notable speedups. While our approach applies directly in a training-free setting, it implicitly assumes that the underlying model can generalize across varying resolutions. Though modern DiT models are trained on multiple resolutions, the continuous schedules we employ may include intermediate resolutions not seen during training. In practice, using a small number of transitions near the training distribution suffices to achieve substantial efficiency gains in both image and video generation. Our fine-tuning strategy aligns the model with the progressive-resolution trajectory, reducing the training–inference gap and improving quality, suggesting the use of our method during pretraining as a promising direction. While we focus on spatial spectra, extending the framework to temporal frequencies is a natural direction for accelerating video generation. Overall, our approach provides a principled and complementary approach for improving generation efficiency via the spectral autoregressive structure of diffusion models while maintaining high generation quality.

Acknowledgments

Howard Xiao and Brian Chao are supported by Stanford Graduate Fellowships (SGF). Brian Chao is also supported by the NSF Graduate Research Fellowship Program (GRFP). We thank Google and Toyota Research Institute (TRI). We thank Shengqu Cai, Hansheng Chen, Kiyohiro Nakayama, and Zichun Xu for fruitful discussions. Compute resources were provided by the Marlowe cluster at Stanford University [34].

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

[3] Roi Benita, Miki Elad, and Joseph Keshet. Spectral analysis of diffusion models with application to schedule design. In *Adv. Neural Inform. Process. Syst.*, volume 38, pages 2073–2127, 2026.

[4] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4599–4603, 2023.

[5] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer, 2025. URL https://arxiv.org/abs/2511.22699.

[6] Satish Chandran, Nicolas Roque dos Santos, Yunshu Wu, Greg Ver Steeg, and Evangelos Papalexakis. Spectral regularization for diffusion models, 2026. URL https://arxiv.org/abs/2603.02447.

[7] Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. δ-DiT: A training-free acceleration method tailored for diffusion transformers, 2024. URL https://arxiv.org/abs/2406.01125.

[8] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *Int. Conf. Learn. Represent.*, 2021.

[9] Sander Dieleman. Diffusion is spectral autoregression. Blog post, September 2024. URL https://sander.ai/2024/09/02/spectral-autoregression.html.

[10] Ruoyi Du, Dongliang Chang, et al. DemoFusion: Democratising high-resolution image generation with no $$$. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[11] Weitao Du, Shuning Chang, Jiasheng Tang, Yu Rong, Fan Wang, and Shengchao Liu. Flow along the k-amplitude for generative modeling, 2025. URL https://arxiv.org/abs/2504.19353.

[12] 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 *Int. Conf. Mach. Learn.*, 2024.

[13] Carlos Esteves and Ameesh Makadia. Spectrally-guided diffusion noise schedules. In *Int. Conf. Mach. Learn.*, 2026.

[14] Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A fourier space perspective on diffusion models, 2025. URL https://arxiv.org/abs/2505.11278.

[15] Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, Ziqi Huang, Ziyue Dong, Jingwen He, Dongwei Pan, et al. Vchitect-2.0: Parallel transformer for scaling up video diffusion models, 2025. URL https://arxiv.org/abs/2501.08453.

[16] Haipeng Fang, Sheng Tang, Juan Cao, Enshuo Zhang, Fan Tang, and Tong-Yee Lee. Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18083–18092, 2025.

[17] Jean-Baptiste Joseph Fourier. *Théorie Analytique de la Chaleur*. Firmin Didot, 1822.

[18] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Adv. Neural Inform. Process. Syst.*, 2023.

[19] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Joshua M Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *Int. Conf. Learn. Represent.*, 2023.

[20] Lanqing Guo, Yingqing He, Haoxin Chen, Menghan Xia, Xiaodong Cun, Yufei Wang, Siyu Huang, Yong Zhang, Xintao Wang, Qifeng Chen, Ying Shan, and Bihan Wen. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation. In *Eur. Conf. Comput. Vis.*, 2024.

[21] Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat. Wavelet score-based generative modeling. In *Adv. Neural Inform. Process. Syst.*, volume 35, pages 478–491, 2022.

[22] Alfréd Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.

[23] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise AutoRegressive modeling for high-resolution image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[24] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Agglomerative token clustering. In *Eur. Conf. Comput. Vis.*, pages 200–218. Springer, 2024.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, 2020.

[26] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.

[27] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Int. Conf. Learn. Represent.*, 2022.

[28] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Adv. Neural Inform. Process. Syst.*, 2023.

[29] Yi Huang, Jiancheng Huang, Jianzhuang Liu, Mingfu Yan, Yu Dong, Jiaxi Lv, Chaoqi Chen, and Shifeng Chen. Wavedm: Wavelet-based diffusion models for image restoration. *IEEE Trans. Multimedia*, 26: 7058–7073, 2024.

[30] Yuanhui Huang, Weiliang Chen, Wenzhao Zheng, Yueqi Duan, Jie Zhou, and Jiwen Lu. Spectralar: Spectral autoregressive visual generation. In *Int. Conf. Comput. Vis.*, pages 15842–15852, 2025.

[31] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[32] Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers, 2026. URL `https://arxiv.org/abs/2507.08422`.

[33] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *Int. Conf. Learn. Represent.*, 2025.

[34] C. Kapfer, K. Stine, B. Narasimhan, C. Mentzel, and E. Candès. Marlowe: Stanford's GPU-based computational instrument, 2025. URL `https://doi.org/10.5281/zenodo.14751899`.

[35] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Int. Conf. Mach. Learn.*, pages 5156–5165. PMLR, 2020.

[36] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

[37] Younghyun Kim, Geunmin Hwang, Junyu Zhang, and Eunbyung Park. DiffuseHigh: Training-free progressive high-resolution image synthesis through structure guidance. In *AAAI*, 2025.

[38] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Adv. Neural Inform. Process. Syst.*, 2021.

[39] Black Forest Labs. FLUX. Software repository, 2024. URL `https://github.com/black-forest-labs/flux`.

[40] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. Blog post, 2025. URL `https://bfl.ai/blog/flux-2`.

[41] Haeil Lee, Hansang Lee, Seoyeon Gye, and Junmo Kim. Beta sampling is all you need: Efficient image generation strategy for diffusion models using stepwise spectral analysis, 2024. URL https://arxiv.org/abs/2407.12173.

[42] Min-Jeong Lee, Hee-Dong Kim, and Seong-Whan Lee. Local representative token guided merging for text-to-image generation, 2025. URL https://arxiv.org/abs/2507.12771.

[43] Ruihuang Li, Lei Zhang, et al. Frecas: Efficient higher-resolution image generation via frequency-aware cascaded sampling. In *Int. Conf. Learn. Represent.*, volume 2025, pages 6400–6412, 2025.

[44] Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, et al. Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. In *Adv. Neural Inform. Process. Syst.*, 2025.

[45] Zongjian Li, Bin Lin, Yang Ye, Liuhan Chen, Xinhua Cheng, Shenghai Yuan, and Li Yuan. Wf-vae: Enhancing video vae by wavelet-driven energy flow for latent video diffusion model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17778–17788, 2025.

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014.

[47] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Int. Conf. Learn. Represent.*, 2023.

[48] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model, 2024. URL https://arxiv.org/abs/2411.19108.

[49] Jiacheng Liu, Peiliang Cai, Qinming Zhou, Yuqi Lin, Deyang Kong, Benhao Huang, Yupei Pan, Haowen Xu, Chang Zou, Junshu Tang, Shikang Zheng, and Linfeng Zhang. FreqCa: Accelerating diffusion models via frequency-aware caching, 2025. URL https://arxiv.org/abs/2510.08669.

[50] Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with TaylorSeers. In *Int. Conf. Comput. Vis.*, 2025.

[51] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Int. Conf. Learn. Represent.*, 2023.

[52] Wenbo Lu, Shaoyi Zheng, Yuxuan Xia, and Shengjie Wang. ToMA: Token merge with attention for diffusion models. In *Int. Conf. Mach. Learn.*, 2025.

[53] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023. URL https://arxiv.org/abs/2310.04378.

[54] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[55] Zehong Ma, Ruihan Xu, and Shiliang Zhang. PixelGen: Pixel diffusion beats latent diffusion with perceptual loss, 2026. URL https://arxiv.org/abs/2602.02493.

[56] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *Int. Conf. Learn. Represent.*, 2022.

[57] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Sign. Process. Letters*, 20(3):209–212, 2013. doi: 10.1109/LSP.2012.2227726.

[58] Soumik Mukhopadhyay, Prateksha Udhayanan, and Abhinav Shrivastava. Scale space diffusion, 2026. URL https://arxiv.org/abs/2603.08709.

[59] Mang Ning, Mingxiao Li, Jianlin Su, Jia Haozhe, Lanmiao Liu, Martin Benes, Wenshuo Chen, Albert Ali Salah, and Itir Onal Ertugrul. DCTdiff: Intriguing properties of image generative modeling in the DCT space. In *Int. Conf. Mach. Learn.*, volume 267, pages 46498–46524. PMLR, 2025.

[60] NVIDIA, Yuval Atzmon, Maciej Bala, Yogesh Balaji, Tiffany Cai, Yin Cui, Jiaojiao Fan, Yunhao Ge, Siddharth Gururani, Jacob Huffman, Ronald Isaac, Pooya Jannaty, Tero Karras, Grace Lam, J. P. Lewis, Aaron Licata, Yen-Chen Lin, Ming-Yu Liu, Qianli Ma, Arun Mallya, Ashlee Martino-Tarr, Doug Mendez, Seungjun Nah, Chris Pruett, Fitsum Reda, Jiaming Song, Ting-Chun Wang, Fangyin Wei, Xiaohui Zeng, Yu Zeng, and Qinsheng Zhang. Edify image: High-quality image generation with pixel space laplacian diffusion models, 2024. URL https://arxiv.org/abs/2411.07126.

[61] Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[62] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Int. Conf. Comput. Vis.*, 2023.

[63] Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[64] Hao Phung, Quan Dao, Trung Dao, Hoang Phan, Dimitris Metaxas, and Anh Tran. DiMSUM: Diffusion Mamba – a scalable and unified spatial-frequency method for image generation. In *Adv. Neural Inform. Process. Syst.*, 2024.

[65] Haonan Qiu et al. FreeScale: Unleashing the resolution of diffusion models via tuning-free scale fusion. In *Int. Conf. Comput. Vis.*, 2025.

[66] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. FlowAR: Scale-wise autoregressive image generation meets flow matching. In *Int. Conf. Mach. Learn.*, 2025.

[67] Daniel L Ruderman. Origins of scaling in natural images. *Vis. Res.*, 37(23):3385–3398, 1997.

[68] Dohoon Ryu and Jong Chul Ye. Pyramidal denoising diffusion probabilistic models, 2022. URL `https://arxiv.org/abs/2208.01864`.

[69] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *Int. Conf. Learn. Represent.*, 2022.

[70] Claude E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[71] Luigi Sigillo, Shengfeng He, and Danilo Comminiello. Latent wavelet diffusion for ultra-high-resolution image synthesis, 2025. URL `https://arxiv.org/abs/2506.00433`.

[72] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. In *Int. Conf. Mach. Learn.*, 2025.

[73] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Int. Conf. Learn. Represent.*, 2021.

[74] Jyun-Ze Tang, Chih-Fan Hsu, Jeng-Lin Li, Ming-Ching Chang, and Wei-Chao Chen. Lssgen: Leveraging latent space scaling in flow and diffusion for efficient text to image generation. In *Int. Conf. Comput. Vis.*, pages 5048–5057, 2025.

[75] Team Wan, Ang Wang, Shiwei Zhang, Jingren Zhou, et al. Wan: Open and advanced large-scale video generative models, 2025. URL `https://arxiv.org/abs/2503.20314`.

[76] 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 *Int. Conf. Learn. Represent.*, 2024. Spotlight.

[77] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Adv. Neural Inform. Process. Syst.*, 2024.

[78] Ye Tian, Xin Xia, Yuxi Ren, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Yunhai Tong, Ling Yang, and Bin Cui. Training-free diffusion acceleration with bottleneck sampling, 2025. URL `https://arxiv.org/abs/2503.18940`.

[79] Tobias Vontobel, Seyedmorteza Sadat, Farnood Salehi, and Romann Weber. Hiwave: Training-free high-resolution image generation via wavelet-based diffusion sampling. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11, 2025.

[80] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models, 2025. URL `https://arxiv.org/abs/2503.20314`.

[81] Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Hongsheng Li, and Hao Ouyang. Phased consistency models. In *Adv. Neural Inform. Process. Syst.*, 2024.

[82] Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *AAAI*, 2023.

[83] Sinong Wang, Belinda Z Li, Madian Khabsa, Hao Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. URL `https://arxiv.org/abs/2006.04768`.

[84] Haoyu Wu, Jingyi Xu, Hieu Le, and Dimitris Samaras. Importance-based token merging for efficient image and video generation. In *Int. Conf. Comput. Vis.*, pages 4983–4995, 2025.

[85] Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation. In *Int. Conf. Comput. Vis.*, 2025.

[86] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Adv. Neural Inform. Process. Syst.*, 2023.

[87] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *Int. Conf. Learn. Represent.*, 2025.

[88] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Adv. Neural Inform. Process. Syst.*, 2024.

[89] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[90] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xide Xia. From slow bidirectional to fast autoregressive video diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[91] Chenlu Zhan, Wen Li, Chuyu Shen, Jun Zhang, Suhui Wu, and Hao Zhang. Bidirectional sparse attention for faster video diffusion training, 2025. URL `https://arxiv.org/abs/2509.01085`.

[92] Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[93] Shikang Zheng, Guantao Chen, Lixuan He, Jiacheng Liu, Yuqi Lin, Chang Zou, and Linfeng Zhang. From sketch to fresco: Efficient diffusion transformer with progressive resolution, 2026. URL `https://arxiv.org/abs/2601.07462`.

[94] Wendi Zheng, Jiayan Teng, Zhuoyi Yang, Weihan Wang, Jidong Chen, Xiaotao Gu, Yuxiao Dong, Ming Ding, and Jie Tang. CogView3: Finer and faster text-to-image generation via relay diffusion. In *Eur. Conf. Comput. Vis.*, 2024.

[95] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching, 2024. URL `https://arxiv.org/abs/2410.05317`.

Appendix Contents

A Supporting Definitions and Proofs for Section 4.2

This appendix provides the justification of timestep alignment in Spectral Progressive Diffusion, the deferred proofs of Propositions 1 and 2 from Sec. 4.2, along with supporting definitions.

A.1 Definitions

Per-frequency SNR. Following Kingma et al. [38], we define the *per-frequency signal-to-noise ratio (SNR)* as the ratio of the clean- and noise-component second moments of the per-frequency forward process in Eq. (7):

$$\mathrm{SNR}_\omega(t) \;:=\; \frac{\mathbb{E}[\,|(1-t)\,x_0^{(\omega)}|^2\,]}{\mathbb{E}[\,|t\,\epsilon^{(\omega)}|^2\,]} \;=\; \underbrace{\frac{(1-t)^2}{t^2}}_{\mathrm{SNR}(t)} \cdot P_\omega, \tag{15}$$

which factorizes into a frequency-independent timestep term $\mathrm{SNR}(t) = (1-t)^2/t^2$ and the frequency-specific signal power P_ω.

Resolution grids and representable frequencies. By the Nyquist–Shannon sampling theorem [61, 70], a spatial grid of size $H \times W$ can represent 2D frequencies in the rectangle $|\omega_x| \le W/2$, $|\omega_y| \le H/2$. We define the maximum (fully) representable radial frequency as

$$\omega_{\max}(H, W) \;=\; \frac{\min(H, W)}{2}. \tag{16}$$

Note that our definition of the maximum (fully) representable frequency above is conservative, as it captures only the maximum radial frequency fully supported on the spatial grid $H \times W$. A stage operating at reduced resolution (sH, sW) for $s \in (0, 1]$ can represent only a subset of the representable frequencies at full-resolution:

$$\Omega_s \;:=\; \{\,\omega \in \Omega : |\omega| \le s \cdot \omega_{\max}(H, W)\,\} \;\subseteq\; \Omega. \tag{17}$$

A.2 Justification of Timestep Alignment

We verify that applying the scalar rescaling in Eq. (5) and querying the pretrained model at the aligned timestep $\tilde{t}_i$ in Eq. (6) maps the spectral noise expansion output $\mathbf{x}_{t_i}^{s_{i+1}}$ to a valid flow-matching state $\tilde{\mathbf{x}}_{\tilde{t}_i}^{s_{i+1}}$ at the new resolution scale $s_{i+1} > s_i$. Throughout this subsection we write $r := s_{i+1}/s_i > 1$ for the resolution ratio; the final formulas are stated in both the r and s_{i+1}/s_i forms to tie directly back to Eqs. (5)–(6) in Sec. 4.1. The derivations below assume the ideal setting in which the network outputs match the ground-truth flow targets.

1. Setup. The spectral noise expansion produces an expanded state $\mathbf{x}_{t_i}^{s_{i+1}}$ in the frequency domain of the expanded resolution grid. For a grid with $N_{s_{i+1}}$ points, the coefficients are:

$$\left(x_{t_i}^{s_{i+1}}\right)^{(\omega)} = \begin{cases} (1-t_i)\,(x_0^{s_i})^{(\omega)} + t_i\epsilon^{(\omega)}, & \omega \in \Omega_{s_i}, \\ t_i\epsilon'^{(\omega)}, & \omega \in \Omega_{s_{i+1}} \setminus \Omega_{s_i}, \end{cases}$$

where $\epsilon^{(\omega)}, \epsilon'^{(\omega)} \sim \mathcal{N}(0, 1)$ are independent standard Gaussian variables. Under the orthonormal identification of spectra across resolutions, the clean-signal coefficient at a shared frequency $\omega \in \Omega_{s_i}$ relates to its scale-s_{i+1} counterpart by the resolution ratio r:

$$\left(x_0^{s_{i+1}}\right)^{(\omega)} = r \cdot (x_0^{s_i})^{(\omega)} = (s_{i+1}/s_i) \cdot \left(x_0^{s_i}\right)^{(\omega)}, \quad \omega \in \Omega_{s_i}.$$

For the newly exposed high-frequency slots $\omega \in \Omega_{s_{i+1}} \setminus \Omega_{s_i}$, the clean coefficients are treated as noise-dominated at timestep t_i per Proposition 1, so spectral noise expansion initializes them with the correct noise level.

2. Timestep alignment. We seek a scaling factor κ_i and an aligned timestep $\tilde{t}_i \in (0,1)$ such that, for every $\omega \in \Omega_{s_{i+1}}$, the transformed state satisfies:

$$\left(\tilde{x}_{\tilde{t}_i}^{s_{i+1}}\right)^{(\omega)} = \kappa_i \left(x_{t_i}^{s_{i+1}}\right)^{(\omega)} = (1-\tilde{t}_i)\left(x_0^{s_{i+1}}\right)^{(\omega)} + \tilde{t}_i \tilde{\epsilon}^{(\omega)}, \quad \tilde{\epsilon}^{(\omega)} \sim \mathcal{N}(0,1) \text{ i.i.d.}$$

By matching the noise coefficients across all bands and the signal coefficients on the shared band Ω_{s_i}, we derive two scalar conditions:

$$\kappa_i \cdot t_i = \tilde{t}_i, \quad \kappa_i(1-t_i) = r(1-\tilde{t}_i).$$

Substituting $\kappa_i = \tilde{t}_i/t_i$ from the noise match into the signal match equation gives:

$$\frac{\tilde{t}_i}{t_i}(1-t_i) = r(1-\tilde{t}_i)$$

Solving for $\tilde{t}_i$ and re-expressing the result in s_{i+1}/s_i notation gives:

$$\tilde{t}_i = \frac{r \cdot t_i}{1+(r-1)t_i} = \frac{(s_{i+1}/s_i)\cdot t_i}{1+((s_{i+1}/s_i)-1)t_i}.$$

Consequently, the scaling factor κ_i is:

$$\kappa_i = \frac{r}{1+(r-1)t_i} = \frac{s_{i+1}/s_i}{1+((s_{i+1}/s_i)-1)t_i}.$$

Substituting these back into the setup confirms that $\left(\tilde{x}_{\tilde{t}_i}^{s_{i+1}}\right)^{(\omega)} = (1-\tilde{t}_i)\left(x_0^{s_{i+1}}\right)^{(\omega)} + \tilde{t}_i\tilde{\epsilon}^{(\omega)}$ holds for all $\omega \in \Omega_{s_{i+1}}$. This proves that the resulting scaled and timestep-aligned state $\tilde{\mathbf{x}}_{\tilde{t}_i}^{s_{i+1}}$ is a valid flow-matching state at resolution s_{i+1} and timestep $\tilde{t}_i$. □

A.3 Proof of Proposition 1: Per-frequency δ-optimal Activation Time

We first establish a lemma giving the Bayes-optimal velocity error in closed form under a simplified modelling assumption $x_0^{(\omega)} \sim \mathcal{N}(0, P_\omega)$. In practice, we measure the power spectrum on centered (i.e. zero-mean) inputs $\mathbf{x}_0$ to get the power spectrum P_ω, and this modelling assumption serves as a second moment approximation that allows the analytical derivation of our δ-optimal resolution schedule. The proof of Proposition 1 then follows by solving the δ-bound on the Bayes-optimal velocity error for t to obtain the activation time t_ω in Eq. (9).

Lemma 1 (Bayes-optimal velocity error)**.** *Let $v^{*(\omega)}$ be the per-frequency component of the Bayes-optimal velocity predictor $\mathbf{v}^*$ defined in Sec. 3.1:*

$$\mathbf{v}^*(\mathbf{x}_t, t) \;=\; \mathbb{E}\big[\,\boldsymbol{\epsilon} - \mathbf{x}_0 \,\big|\, \mathbf{x}_t\,\big]\,. \tag{18}$$

Under the modelling assumption $x_0^{(\omega)} \sim \mathcal{N}(0, P_\omega)$, $v^{*(\omega)}$ deviates from the noise-only prediction $\epsilon^{(\omega)}$ in expectation by

$$\mathbb{E}\Big[\big|v^{*(\omega)}(\mathbf{x}_t,t) - \epsilon^{(\omega)}\big|^2\Big] \;=\; \frac{\mathrm{SNR}_\omega(t)\,(1+P_\omega)}{1+\mathrm{SNR}_\omega(t)}. \tag{19}$$

Proof.

1. Assumptions. The proof assumes a linear-Gaussian model, where:

- (A1) we model the clean data spectrum as zero-mean Gaussian $x_0^{(\omega)} \sim \mathcal{N}(0,\, P_\omega)$ with second moment $P_\omega > 0$;
- (A2) we assume $\epsilon^{(\omega)} \sim \mathcal{N}(0,1)$, by orthonormality of Φ applied to $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$;
- (A3) we assume $x_0^{(\omega)} \perp \epsilon^{(\omega)}$, i.e. the signal and noise coefficients are independent.

(A1) is the simplified modelling assumption; (A2) and (A3) follow directly from the flow-matching forward process. Under (A1), the triple $(x_0^{(\omega)},\, \epsilon^{(\omega)},\, x_t^{(\omega)})$ is jointly Gaussian, so conditional expectations are linear and closed-form.

2. Second-moment quantities. From the per-frequency forward process in Eq. (7) and (A1)–(A3),

$$\mathbb{E}[x_t^{(\omega)}] = 0, \qquad \mathrm{Var}(x_t^{(\omega)}) \;=\; (1-t)^2 P_\omega + t^2, \tag{20}$$

$$\mathrm{Cov}(x_t^{(\omega)},\, x_0^{(\omega)}) = (1-t)\, P_\omega, \qquad \mathrm{Cov}(x_t^{(\omega)},\, \epsilon^{(\omega)}) \;=\; t. \tag{21}$$

3. Bayes-optimal velocity as a linear predictor. Under (A1)–(A3), the triplet $(x_0^{(\omega)},\, \epsilon^{(\omega)},\, x_t^{(\omega)})$ is jointly Gaussian with zero mean, since $x_t^{(\omega)} = (1-t)x_0^{(\omega)} + t\,\epsilon^{(\omega)}$ is a linear combination of two independent zero-mean Gaussians. For any pair of zero-mean jointly Gaussian random variables (X, Y) with $\mathrm{Var}(Y) > 0$, the conditional expectation is a linear function of Y:

$$\mathbb{E}[X \,|\, Y] \;=\; \frac{\mathrm{Cov}(X,Y)}{\mathrm{Var}(Y)}\, Y. \tag{22}$$

We apply (22) twice, using the second moments computed above. First, with $X = x_0^{(\omega)}$ and $Y = x_t^{(\omega)}$,

$$\mathbb{E}\big[x_0^{(\omega)} \,\big|\, x_t^{(\omega)}\big] \;=\; \frac{(1-t)\, P_\omega}{(1-t)^2 P_\omega + t^2}\, x_t^{(\omega)}. \tag{23}$$

Second, with $X = \epsilon^{(\omega)}$ and $Y = x_t^{(\omega)}$,

$$\mathbb{E}\big[\epsilon^{(\omega)} \,\big|\, x_t^{(\omega)}\big] \;=\; \frac{t}{(1-t)^2 P_\omega + t^2}\, x_t^{(\omega)}. \tag{24}$$

The Bayes-optimal velocity at frequency ω is $v^{*(\omega)} = \mathbb{E}\big[\epsilon^{(\omega)} - x_0^{(\omega)} \,\big|\, \mathbf{x}_t\big]$, the ϕ_ω-projection of the full-input optimal velocity in Eq. (18). Under the modelling assumption that $\mathbf{x}_0$ has diagonal covariance in basis Φ (i.e., (A1) holds independently for every ω), the joint Gaussian $(\mathbf{x}_0, \boldsymbol{\epsilon}, \mathbf{x}_t)$ factorizes across Ω, so the posterior collapses to its single-coefficient marginal:

$$\mathbb{E}\big[\epsilon^{(\omega)} - x_0^{(\omega)} \,\big|\, \mathbf{x}_t\big] \;=\; \mathbb{E}\big[\epsilon^{(\omega)} - x_0^{(\omega)} \,\big|\, x_t^{(\omega)}\big]. \tag{25}$$

Subtracting (23) from (24) and combining with (25),

$$v^{*(\omega)} \;=\; A \cdot x_t^{(\omega)}, \qquad A \;:=\; \frac{t - (1-t)\, P_\omega}{(1-t)^2 P_\omega + t^2}. \tag{26}$$

4. Substitute and simplify. Plugging $x_t^{(\omega)} = (1-t)\, x_0^{(\omega)} + t\,\epsilon^{(\omega)}$ into (26) and collecting terms in $x_0^{(\omega)}$ and $\epsilon^{(\omega)}$,

$$v^{*(\omega)} - \epsilon^{(\omega)} \;=\; A\big[(1-t)\, x_0^{(\omega)} + t\,\epsilon^{(\omega)}\big] - \epsilon^{(\omega)} \;=\; A(1-t)\, x_0^{(\omega)} \;+\; (At-1)\, \epsilon^{(\omega)}. \tag{27}$$

Squaring and taking expectation, using (A1)–(A3) (so $\mathbb{E}[(x_0^{(\omega)})^2] = P_\omega$, $\mathbb{E}[(\epsilon^{(\omega)})^2] = 1$, and $\mathbb{E}[x_0^{(\omega)}\, \epsilon^{(\omega)}] = 0$), the cross term vanishes:

$$\mathbb{E}\Big[\big|v^{*(\omega)} - \epsilon^{(\omega)}\big|^2\Big] \;=\; A^2(1-t)^2\, P_\omega \;+\; (At-1)^2. \tag{28}$$

Expanding $(At-1)^2 = A^2t^2 - 2At + 1$ and grouping the A^2 terms,

$$\mathbb{E}\Big[\big|v^{*(\omega)} - \epsilon^{(\omega)}\big|^2\Big] \;=\; A^2\big[(1-t)^2 P_\omega + t^2\big] - 2At + 1. \tag{29}$$

Denote $D := (1-t)^2 P_\omega + t^2 = \mathrm{Var}(x_t^{(\omega)})$; from Eq. (26),

$$A \;=\; \frac{t - (1-t)\, P_\omega}{D}. \tag{30}$$

Substituting into Eq. (29) term by term,

$$A^2\big[(1-t)^2 P_\omega + t^2\big] \;=\; A^2 D \;=\; \frac{(t - (1-t)\, P_\omega)^2}{D}, \tag{31}$$

$$2At \;=\; \frac{2t\,(t - (1-t)\, P_\omega)}{D}, \tag{32}$$

so the right-hand side of Eq. (29) combines into a single fraction over D:

$$\mathbb{E}\Big[\big|v^{*(\omega)} - \epsilon^{(\omega)}\big|^2\Big] \;=\; \frac{(t-(1-t)\,P_\omega)^2 \;-\; 2t\,(t-(1-t)\,P_\omega) \;+\; D}{D}. \tag{33}$$

Expanding $(t-(1-t)\,P_\omega)^2 = t^2 - 2t(1-t)P_\omega + (1-t)^2P_\omega^2$, the numerator of Eq. (33) simplifies as

$$\begin{aligned}&\big[t^2 - 2t(1-t)P_\omega + (1-t)^2P_\omega^2\big] - \big[2t^2 - 2t(1-t)P_\omega\big] + \big[(1-t)^2P_\omega + t^2\big] \\ &= (1-t)^2P_\omega^2 + (1-t)^2P_\omega \;=\; (1-t)^2\,P_\omega\,(1+P_\omega).\end{aligned} \tag{34}$$

Substituting back into Eq. (33),

$$\mathbb{E}\Big[\big|v^{*(\omega)} - \epsilon^{(\omega)}\big|^2\Big] \;=\; \frac{(1-t)^2\,P_\omega\,(1+P_\omega)}{(1-t)^2P_\omega + t^2}. \tag{35}$$

5. Rewrite via $\mathrm{SNR}_\omega(t)$. Using $\mathrm{SNR}_\omega(t) = (1-t)^2P_\omega/t^2$ from Eq. (15), the denominator is $(1-t)^2P_\omega + t^2 = t^2(1+\mathrm{SNR}_\omega(t))$ and the numerator equals $t^2\,\mathrm{SNR}_\omega(t)\,(1+P_\omega)$. The t^2 cancels, recovering the right-hand side of Eq. (19) and completing the proof of the lemma.

□

Proposition 1 (Per-frequency δ-optimal activation time)**.** *Denote the per-frequency Bayes-optimal velocity predictor by $v^{*(\omega)}(\mathbf{x}_t, t)$, whose spatial-domain counterpart is defined in Eq.* (18)*. Under the simplified modelling assumption $x_0^{(\omega)} \sim \mathcal{N}(0, P_\omega)$ with $P_\omega > 0$, we have*

$$\mathbb{E}\Big[\big|v^{*(\omega)}(\mathbf{x}_t, t) - \epsilon^{(\omega)}\big|^2\Big] \;\le\; \delta \tag{36}$$

for all $t \ge t_\omega$ and $\delta \in (0,1)$, where

$$t_\omega \;:=\; \frac{1}{1+\sqrt{\dfrac{\delta}{P_\omega\,(1+P_\omega-\delta)}}}. \tag{37}$$

Proof.

Solving the δ-bound for t. Substituting the closed-form error into the tolerance bound $\mathbb{E}[|v^{*(\omega)} - \epsilon^{(\omega)}|^2] \le \delta$ of Proposition 1 yields

$$\frac{\mathrm{SNR}_\omega(t)\,(1+P_\omega)}{1+\mathrm{SNR}_\omega(t)} \;\le\; \delta. \tag{38}$$

Clearing the (positive) denominator and collecting terms in $\mathrm{SNR}_\omega(t)$,

$$\mathrm{SNR}_\omega(t)\,(1+P_\omega-\delta) \;\le\; \delta, \qquad \text{i.e.,} \qquad \mathrm{SNR}_\omega(t) \;\le\; \frac{\delta}{1+P_\omega-\delta}, \tag{39}$$

where the rearrangement is valid because $\delta < 1 \le 1+P_\omega$ keeps $1+P_\omega-\delta > 0$. Substituting $\mathrm{SNR}_\omega(t) = (1-t)^2P_\omega/t^2$ and taking the positive square root (all quantities positive for $t \in (0,1)$),

$$\frac{1-t}{t} \;\le\; \sqrt{\frac{\delta}{P_\omega\,(1+P_\omega-\delta)}}. \tag{40}$$

Adding 1 to both sides gives $1/t \le 1 + \sqrt{\delta/(P_\omega(1+P_\omega-\delta))}$, which inverts to

$$t \;\ge\; \frac{1}{1+\sqrt{\dfrac{\delta}{P_\omega\,(1+P_\omega-\delta)}}} \;=\; t_\omega, \tag{41}$$

recovering Eq. (37). □

A.4 Proof of Proposition 2: Per-resolution δ-optimal Transition Time

Proposition 2 (Per-resolution δ-optimal transition time)**.** *Under the setting of Proposition 1, and assuming P_ω is monotonically decreasing in $|\omega|$ based on the power-law decay in Eq.* (4)*, for any resolution scale $s_i, s_{i+1} \in (0, 1]$, the optimal transition time from scale s_i up to s_{i+1} is*

$$t_i^* \;:=\; \min_{\omega \in \Omega_{s_i}} t_\omega \;=\; t_{\omega = s_i \cdot \omega_{\max}(H,W)}, \tag{42}$$

where Ω_{s_i} is the set of frequencies representable on the $(s_i H, s_i W)$ grid.

Proof.

We show that the activation time t_ω of Eq. (9) is strictly decreasing in $|\omega|$ under the assumptions of the proposition; the claim then follows immediately.

1. t_ω is strictly increasing in P_ω. For $\delta \in (0, 1)$ and $P_\omega > 0$, the quadratic $P_\omega\,(1 + P_\omega - \delta) = P_\omega^2 + (1 - \delta)\,P_\omega$ has derivative $2P_\omega + (1 - \delta) > 0$, hence is strictly increasing in P_ω. Therefore the radicand $\delta / \big(P_\omega(1 + P_\omega - \delta)\big)$ in Eq. (9) is strictly decreasing in P_ω, and so is its square root. Hence

$$t_\omega \;=\; \frac{1}{1 + \sqrt{\dfrac{\delta}{P_\omega\,(1 + P_\omega - \delta)}}} \tag{43}$$

is strictly increasing in P_ω.

2. P_ω is strictly decreasing in $|\omega|$. The power-law assumption in Eq. (4) gives $P_\omega \propto |\omega|^{-\beta}$ with $\beta > 0$. Thus P_ω is strictly decreasing in $|\omega|$.

3. Minimum over Ω_s. Combining the above observations, t_ω is strictly decreasing in $|\omega|$. The set $\Omega_s = \{\omega \in \Omega : |\omega| \le s \cdot \omega_{\max}(H, W)\}$ consists exactly of the frequencies with radius at most $s \cdot \omega_{\max}(H, W)$, so

$$\min_{\omega \in \Omega_s} t_\omega \;=\; t_{\,s \cdot \omega_{\max}(H,W)}, \tag{44}$$

which is Eq. (10). □

B Empirical Validation of the Per-frequency Activation Criterion

We empirically validate Proposition 1 with a simple *spectral noise passthrough* experiment. While maintaining full-resolution processing at all denoising steps, for each frequency $\omega \in \Omega$, if the current timestep t is before its activation time t_ω (i.e., $t > t_\omega$), we replace that frequency component with $t \cdot T_\Phi(\boldsymbol{\epsilon})^{(\omega)}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ is the initial noise at $t = 1$. We use the Discrete Cosine Transform (DCT) as T_Φ in this experiment and sweep over multiple δ values. As described in Sec. 4.2, a larger δ corresponds to transitioning to the high-resolution stage later in denoising with more speedup, while a smaller δ indicates transitioning to high-resolution earlier in denoising with less speedup.

From Fig. 5, we observe that at smaller $\delta \in [0.0001, 0.001]$, there is almost no observable difference compared to native full-resolution generation. As δ increases, we see a clear transition to blurry and lower-quality images around $\delta = 0.05$, and larger values further degrade generation quality. This finding provides empirical support for the spectral autoregression property of diffusion models and Proposition 1, which serves as the foundation of our Spectral Progressive Diffusion framework and the δ-optimal resolution schedule. It also implies that the Gaussian modelling assumption in Proposition 1 serves as a reasonable approximation for subsequent analytical resolution schedule derivations.

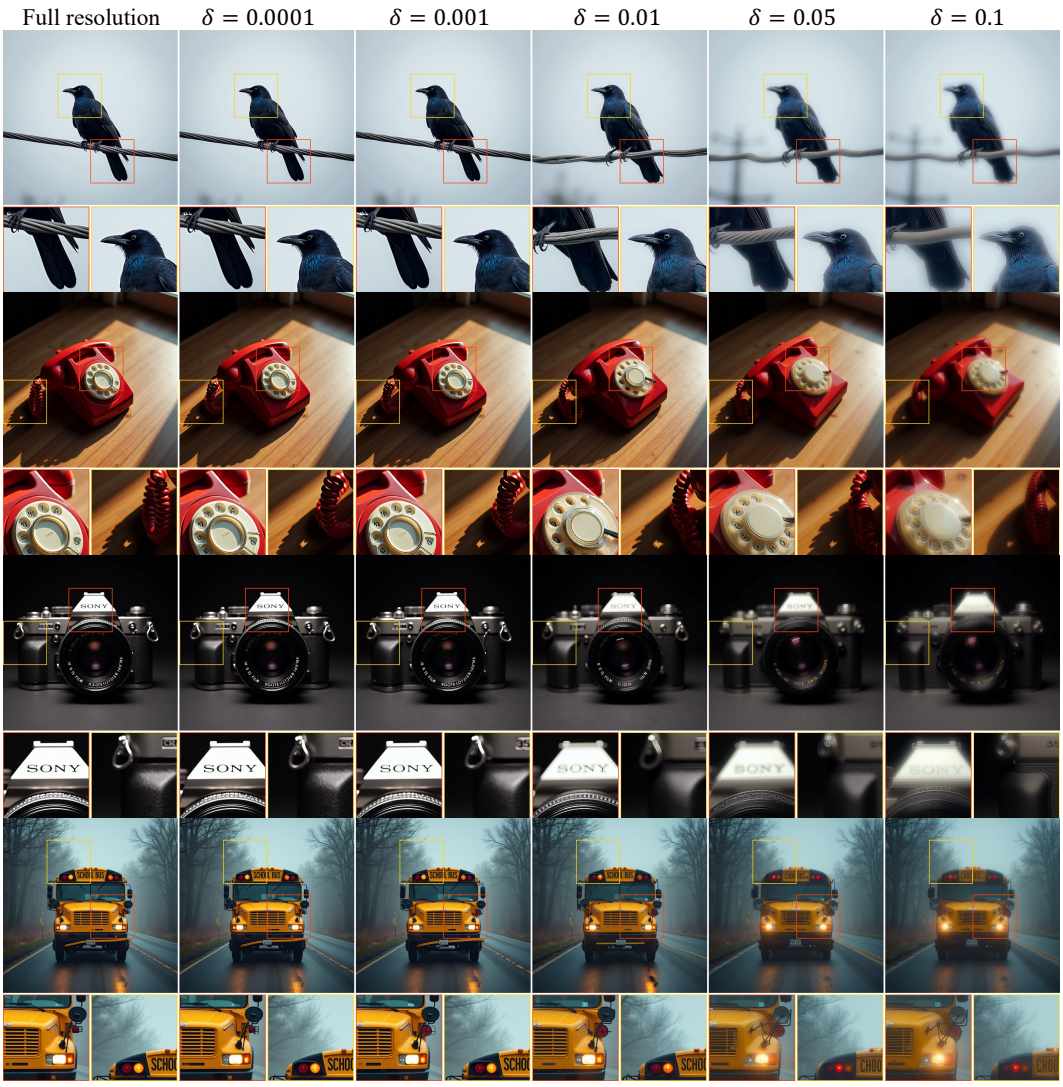

Figure 5: **Spectral noise passthrough experiment.** At smaller $\delta \in [0.0001, 0.001]$, there is almost no observable difference compared to native full-resolution generation. As larger δ values cause high-frequency replacement to persist later in the denoising trajectory, we observe increasingly blurry and distorted results (i.e., ghosting artifacts and "CHOOLBUS" instead of "SCHOOLBUS").

C Additional Details on Power Spectrum Measurement

To remain consistent with the zero-mean Gaussian modelling assumption in Sec. 4.2, we center each data sample before measuring the radially averaged power spectrum of the VAE latent space of FLUX.1-dev [39] and Z-Image [5] for image generation, and the VAE latent space of WAN 2.1 [75] for video generation. We fit the power-law model $P_\omega \approx A \cdot |\omega|^{-\beta}$ and report the fitted A and β values for image latents, video latents, and image pixels in Table 4. These fitted values instantiate the per-frequency signal power P_ω in the activation-time and transition-time formulas in Sec. 4.2, which we use to compute the δ-optimal resolution schedule for each model family.

For image latents, we run the FLUX.1-dev VAE on 100K images from the Aesthetics-Train-V2 [92] dataset, center-cropped and resized to 1024^2 resolution. We then average power over channels and radial frequency bins before fitting the spectrum. For video latents, we use the same procedure with the WAN 2.1 VAE on 10K videos from the VChitect-T2V-Dataverse [15] dataset, averaging over channels, latent frames, and radial frequency bins to estimate the spatial power spectrum. We further measure the pixel-space power spectrum on images from the Aesthetics-Train-V2 dataset, center-cropped and resized to 512^2, the native resolution supported by the PixelGen model.

Table 4: **Power-law fits to measured power spectra.** We fit $P_\omega \approx A \cdot |\omega|^{-\beta}$ on centered samples for image latents, video latents, and image pixels.

Spectrum	β	A	R^2	Samples
FLUX.1-dev (Z-Image) latent	1.9155	203.62	0.978	100K images
WAN latent	2.4227	219.48	0.999	10K videos
Pixel 512^2	2.4493	3745.46	1.000	100K images

D Additional Details and Results on Image Generation

D.1 Training-free Acceleration of Latent-space Image Generation

For training-free inference acceleration of image generation, we use NVIDIA A100 GPUs with 80 GB of VRAM to match RALU [32]'s evaluation protocol. For training-free inference acceleration of video generation, as well as image editing and ablation studies, we use NVIDIA H100 GPUs with 80 GB of VRAM. For training-free acceleration of latent-space image generation, we use FLUX.1-dev [39] as the base model and follow the evaluation protocol of RALU [32]. Table 5 extends Table 1 with additional training-free acceleration baselines from RALU. We organize the comparison into three wall-clock speedup tiers:

- **Small speedup**: the 50-step native full-resolution baseline, compared against RALU (50 steps) and our method (40 steps), yielding approximately $1.7\times$ speedup;
- **Medium speedup**: the 10-step native full-resolution baseline, used as an iso-compute reference for RALU (15 steps) and our method (12 steps), yielding approximately $5\times$ speedup; and
- **Large speedup**: the 7-step native full-resolution baseline, used as an iso-compute reference for RALU (10 steps) and our method (10 steps), yielding approximately $7\times$ speedup.

All speedups are calculated relative to the wall-clock latency of the 50-step FLUX.1-dev baseline on a single A100 GPU, following RALU [32]. As in Table 1, we evaluate image quality using CLIP-IQA [82] and NIQE [57], prompt alignment using T2I-CompBench [28] and GenEval [18], and overall quality using ImageReward [86]. Following RALU's evaluation protocol, ImageReward, CLIP-IQA, and NIQE are averaged over 5,000 images generated from MS-COCO validation prompts. For compositional alignment, GenEval is evaluated with four random seeds per prompt on its object-focused prompt suite, and T2I-CompBench is evaluated on its spatial, non-spatial, and complex prompt subsets with four random seeds per prompt. Unless otherwise specified, all metrics are computed using their official or default evaluation settings.

We keep the error threshold fixed at $\delta = 0.01$, use $T_\Phi = \text{DCT}$, and compare $S = 2$ and $S = 3$ resolution-stage settings. To align with the multi-resolution training distribution of FLUX.1-dev, we use latent dimensions 64^2 ($s_1 = 0.5$) and 128^2 ($s_2 = 1.0$) (pixel dimensions 512^2 and 1024^2) for $S = 2$, and latent dimensions 32^2 ($s_1 = 0.25$), 64^2 ($s_2 = 0.5$), and 128^2 ($s_3 = 1.0$) (pixel dimensions 256^2, 512^2, and 1024^2) for $S = 3$.

By combining spectral transformations for progressive-resolution generation with our derived resolution schedule, our method achieves a better speed-quality tradeoff than the reduced-step FLUX.1-dev baselines and existing temporal and spatial diffusion acceleration methods, including progressive-resolution methods such as RALU [32]. In the largest speedup tier, our $S = 3$ setting reaches $7.09\times$ wall-clock speedup and $7.36\times$ FLOPs speedup while preserving competitive image quality. We further provide extended qualitative baseline comparisons in Figures 6–8. FLUX.1-dev with reduced steps degrades image quality and exhibits over-saturation artifacts, while RALU introduces noticeable noise artifacts in reduced-step settings.

D.2 Spectral-transformation-based Fine-tuning for Latent- and Pixel-Space Image Generation

Additional latent-space fine-tuning results. Since FLUX.1-dev is a guidance-distilled model and its base model is not open-sourced, we additionally conduct latent-space image generation fine-tuning experiments on Z-Image [5], which employs the same VAE architecture as FLUX.1-dev and hence the same δ-optimal resolution schedules. We adopt Low-Rank Adaptation (LoRA) of rank 32 and train for 2000 steps following the spectral-transformation-based fine-tuning procedure in Sec. 4.3. For fair comparison with the full-resolution and training-free acceleration baselines, we fine-tune on synthetic images generated by full-resolution Z-Image with 50 denoising steps using a set of 5K MS-COCO prompts that is disjoint from the 5K MS-COCO validation prompts used for evaluation. Training is conducted on our internal cluster of 8 NVIDIA H100 GPUs, each with 80 GB of VRAM, at an effective batch size of 8 and takes approximately 2 hours. We keep the optimizer and all other LoRA fine-tuning hyperparameters at the default settings of the `DiffSynth Studio` library, including a constant learning rate of 1e-4. Fine-tuning compute scales linearly with the number of sampled

Table 5: **Extended quantitative comparisons on training-free latent-space image generation.** We evaluate FLUX.1-dev at 1024^2 resolution and group methods into the same $1\times$, $5\times$, and $7\times$ wall-clock speedup tiers as Table 1. Rows shared with Table 1 use the same values. Speedup is calculated relative to the 50-step FLUX.1-dev baseline wall-clock runtime on a single A100 GPU. **T** and **S** denote temporal and spatial acceleration methods, respectively. Baseline metrics are copied from RALU [32] under the same evaluation protocol.

Method	Accel.	Latency (s) ↓	Speedup (s) ↑	TFLOPs ↓	Overall	Image quality		Text alignment	
					ImageReward ↑	CLIP-IQA ↑	NIQE ↓	T2I-Comp. ↑	GenEval ↑
FLUX (50 steps)	-	25.1	1.00×	2991.01	**1.095**	0.707	6.75	**0.634**	**0.698**
RALU [32]	S	15.85	1.58×	1749.94	1.028	0.712	**6.07**	0.613	0.648
Ours ($S = 2$)	S	15.16	1.66×	1755.22	1.049	**0.719**	6.43	0.617	0.654
Ours ($S = 3$)	S	**14.54**	**1.73**×	**1672.04**	1.015	0.711	6.33	0.593	0.640
FLUX (10 steps)	T	5.18	4.84×	610.02	0.981	0.679	6.93	0.618	0.647
Δ-DiT [7]	T	7.42	3.38×	772.10	0.102	0.487	9.60	0.306	0.397
ToCa [95]	T	15.5	1.62×	601.12	-1.827	0.253	10.6	0.259	0.137
TeaCache [48]	T	5.23	4.80×	610.59	0.944	0.665	7.92	0.620	0.647
TaylorSeer [50]	T	9.34	2.69×	556.72	0.972	0.684	6.77	0.594	0.619
Bottleneck [78]	S	5.37	4.67×	571.23	0.889	0.661	9.16	0.620	**0.687**
RALU [32]	S	5.04	4.98×	540.47	1.022	0.700	**6.43**	**0.626**	0.652
Ours ($S = 2$)	S	4.35	5.77×	500.34	**1.059**	0.696	6.69	0.624	0.655
Ours ($S = 3$)	S	**4.12**	**6.09**×	**469.15**	1.042	**0.701**	6.53	0.623	0.637
FLUX (7 steps)	T	3.79	6.62×	431.45	0.920	0.660	8.25	0.594	0.583
TeaCache [48]	T	4.21	5.96×	431.83	0.733	0.623	13.7	0.599	0.594
TaylorSeer [50]	T	7.00	3.59×	431.74	0.660	0.646	9.43	0.514	0.446
Bottleneck [78]	S	3.78	6.64×	431.52	0.792	0.631	8.71	0.605	0.672
RALU [32]	S	3.75	6.69×	426.01	0.999	0.681	6.87	**0.633**	**0.682**
Ours ($S = 2$)	S	3.70	6.78×	427.03	**1.039**	0.689	6.78	0.620	0.667
Ours ($S = 3$)	S	**3.54**	**7.09**×	**406.24**	1.015	**0.694**	**5.99**	0.627	0.637

training images, training iterations, and batch size. In our experiments we fix the synthetic fine-tuning set to 5K prompts and train LoRA adapters for 2000 iterations, so increasing the fine-tuning dataset size would only affect compute through additional sampled training examples and does not change the inference-time cost of our method.

All latent-image generation experiments are conducted at a native resolution of 1024^2 pixels, which is the default configuration for both FLUX.1-dev and Z-Image. The progressive resolution settings for the $S = 2$ and $S = 3$ cases are the same as in the FLUX.1-dev training-free acceleration experiments in Sec. D.1.

Further model fine-tuning allows a more aggressive error tolerance δ, leading to an even higher speedup. For fine-tuning experiments, we additionally evaluate $\delta = 0.05$, which is larger than the default $\delta = 0.01$. As in Sec. D.1, we organize the comparison into three wall-clock speedup tiers, all calculated relative to the 50-step Z-Image baseline on a single H100 GPU (we adopt H100 wall-clock measurements for Z-Image and all subsequent pixel-space image generation and video generation experiments):

- **Small speedup**: the 50-step native full-resolution Z-Image baseline, compared against our training-free acceleration and LoRA fine-tuned variants at $S = 2$ and $S = 3$ ($\delta = 0.01$, 50 steps), as well as our LoRA fine-tuned variant with the configuration ($\delta = 0.05$, $S = 2$, 50 steps), yielding $1.65\times$ to $2.01\times$ speedup;
- **Medium speedup**: the 10-step native full-resolution Z-Image baseline, used as an iso-compute reference for our training-free acceleration and LoRA fine-tuned variants at $S = 2$ and $S = 3$ with reduced denoising steps, yielding approximately $5\times$ speedup; and
- **Large speedup**: the 7-step native full-resolution Z-Image baseline, used as an iso-compute reference for our training-free acceleration and LoRA fine-tuned variants at $S = 2$ and $S = 3$ with further reduced denoising steps, yielding approximately $7\times$ to $8\times$ speedup.

For quantitative evaluation, we use the same metric evaluation settings as Sec. D.1, following the RALU evaluation protocol, and report the numbers in Table 6. We show extensive qualitative comparisons in Figures 9–11. Across all three speedup tiers, our training-free acceleration variants outperform the reduced-step Z-Image baselines, and our spectral-transformation-based LoRA fine-tuning further closes the gap to the 50-step full-resolution baseline while attaining up to $7.81\times$ wall-clock speedup. In particular, our LoRA fine-tuned model at $S = 3$ outperforms the training-free acceleration variant at $S = 2$ while improving efficiency, and our LoRA fine-tuned model at $S = 2$

Table 6: **Quantitative comparisons on latent-space image generation with fine-tuning.** We evaluate Z-Image at 1024^2 resolution and group methods into wall-clock speedup tiers. Speedup is calculated relative to the 50-step Z-Image baseline wall-clock runtime on a single H100 GPU. **T** and **S** denote temporal and spatial acceleration methods, respectively. All metrics are evaluated using the same protocol as Table 1.

Method	Accel.	Latency (s) ↓	Speedup (s) ↑	TFLOPs ↓	Overall	Image quality		Text alignment	
					ImageReward ↑	CLIP-IQA ↑	NIQE ↓	T2I-Comp. ↑	GenEval ↑
Z-Image (50 steps)	-	21.25	1.00×	4941.23	0.965	0.700	5.41	0.731	0.745
Z-Image (32 steps)	**T**	13.59	1.56×	3166.62	0.957	0.697	**5.44**	0.686	0.725
Ours (TF, $\delta = 0.01$, $S = 2$)	**S**	12.90	1.65×	3132.03	0.904	0.688	5.87	0.658	0.730
Ours (TF, $\delta = 0.01$, $S = 3$)	**S**	12.19	1.74×	2871.09	0.875	0.690	5.59	0.650	0.682
Ours (LoRA, $\delta = 0.01$, $S = 2$)	**S**	12.87	1.65×	3132.03	**0.982**	**0.699**	5.72	**0.725**	**0.731**
Ours (LoRA, $\delta = 0.01$, $S = 3$)	**S**	12.23	1.74×	2871.09	0.954	0.697	5.75	0.717	0.728
Ours (LoRA, $\delta = 0.05$, $S = 2$)	**S**	**10.59**	**2.01**×	**2436.19**	0.919	0.684	6.12	0.661	0.718
Z-Image (10 steps)	**T**	4.26	4.99×	997.68	0.851	0.659	5.95	0.678	0.705
Ours (TF, $\delta = 0.01$, $S = 2$)	**S**	4.22	5.04×	962.95	0.860	0.668	6.17	0.677	0.693
Ours (TF, $\delta = 0.01$, $S = 3$)	**S**	**4.01**	**5.30**×	**875.97**	0.827	0.662	5.78	0.676	0.655
Ours (LoRA, $\delta = 0.01$, $S = 2$)	**S**	4.24	5.01×	962.95	**0.923**	**0.683**	**5.58**	**0.706**	**0.738**
Z-Image (7 steps)	**T**	3.56	5.97×	701.92	0.763	0.631	**6.14**	0.648	0.667
Ours (TF, $\delta = 0.01$, $S = 2$)	**S**	2.75	7.73×	678.77	0.804	0.641	6.46	0.665	0.680
Ours (TF, $\delta = 0.01$, $S = 3$)	**S**	**2.63**	**8.08**×	**609.18**	0.759	0.630	6.16	0.658	0.645
Ours (LoRA, $\delta = 0.01$, $S = 2$)	**S**	2.72	7.81×	678.77	**0.918**	**0.658**	6.57	**0.683**	**0.744**

either with reduced denoising steps or at a larger $\delta = 0.05$ attains image quality comparable to the training-free acceleration $S = 2$, 50-step setting, providing additional efficiency at no quality cost.

Additional pixel-space fine-tuning results. For pixel-space image generation, we evaluate our method on PixelGen-XXL/16 T2I [55] at its native resolution of 512^2 pixels. We use the same fine-tuning data construction and LoRA training procedure as the Z-Image latent-space experiments above, with synthetic fine-tuning images generated by full-resolution PixelGen with 25 denoising steps from a set of 5K MS-COCO prompts disjoint from the 5K MS-COCO validation prompts used for evaluation. Since PixelGen is pretrained with x_0 prediction rather than flow matching, we convert the spectral-transform velocity targets from the fine-tuning procedure in Sec. 4.3 to x_0 targets and supervise the LoRA adapter with the converted x_0.

We use $S = 2$ with $\delta = 0.01$ and, following Sec. D.1, organize the comparison into two wall-clock speedup tiers, all calculated relative to the 25-step PixelGen baseline on a single H100 GPU:

- **Small speedup**: the 25-step native full-resolution PixelGen baseline, used as the reference for the reduced 13-step PixelGen variant and our training-free acceleration and LoRA fine-tuned variants at $S = 2$ (25 steps), yielding approximately $1.5\times$ to $1.9\times$ speedup;
- **Medium speedup**: the 8-step native full-resolution PixelGen variant, used as an iso-compute reference for our training-free acceleration and LoRA fine-tuned variants at $S = 2$ with reduced denoising steps, yielding approximately $2\times$ to $3\times$ speedup.

For quantitative evaluation, we follow the same protocol used in latent-space image generation and report the numbers in Table 7. We show extensive qualitative comparisons in Figures 12–14. Because PixelGen does not natively support a high-fidelity 256^2 resolution generation, our training-free acceleration variant falls short of the reduced-step PixelGen baselines on overall image quality across both speedup tiers. Our spectral-transformation-based LoRA fine-tuning closes this gap: at the same FLOPs as the training-free acceleration variant, the LoRA fine-tuned model matches or surpasses the reduced-step PixelGen baseline on most image-quality and prompt-alignment metrics, including ImageReward, CLIP-IQA, and T2I-CompBench. The qualitative comparisons in Figures 12–14 corroborate this trend, showing that our LoRA fine-tuned model's outputs match the visual quality of the 25-step native-resolution PixelGen baseline at substantially higher throughput.

Table 7: **Quantitative comparisons on pixel-space image generation.** The number in parentheses next to PixelGen indicates the total number of inference steps. ↑ / ↓ denotes that a higher / lower metric is favourable. Speedup is calculated relative to the 25-step PixelGen baseline wall-clock runtime on a single H100 GPU. **T** and **S** denote temporal and spatial acceleration, respectively.

Method	Accel.	Latency (s) ↓	Speedup (s) ↑	TFLOPs ↓	Overall	Image quality		Text alignment	
					ImageReward ↑	CLIP-IQA ↑	NIQE ↓	T2I-Comp. ↑	GenEval ↑
PixelGen (25 steps)	-	0.48	1.00×	65.36	0.921	0.734	5.95	0.574	0.794
PixelGen (13 steps)	**T**	**0.25**	**1.92**×	34.16	0.886	0.726	5.89	0.571	0.781
Ours (TF, $\delta = 0.01$**,** $S = 2$**)**	**S**	0.30	1.60×	**33.72**	0.799	0.718	6.10	0.568	**0.782**
Ours (LoRA, $\delta = 0.01$**,** $S = 2$**)**	**S**	0.31	1.55×	**33.72**	**0.913**	**0.728**	**5.87**	**0.580**	0.776
PixelGen (8 steps)	**T**	**0.16**	**3.00**×	21.16	0.858	0.715	**5.79**	0.572	0.756
Ours (TF, $\delta = 0.01$**,** $S = 2$**)**	**S**	0.21	2.29×	**20.75**	0.779	0.713	6.02	0.567	0.765
Ours (LoRA, $\delta = 0.01$**,** $S = 2$**)**	**S**	0.22	2.18×	**20.75**	**0.908**	**0.724**	5.85	**0.577**	**0.770**

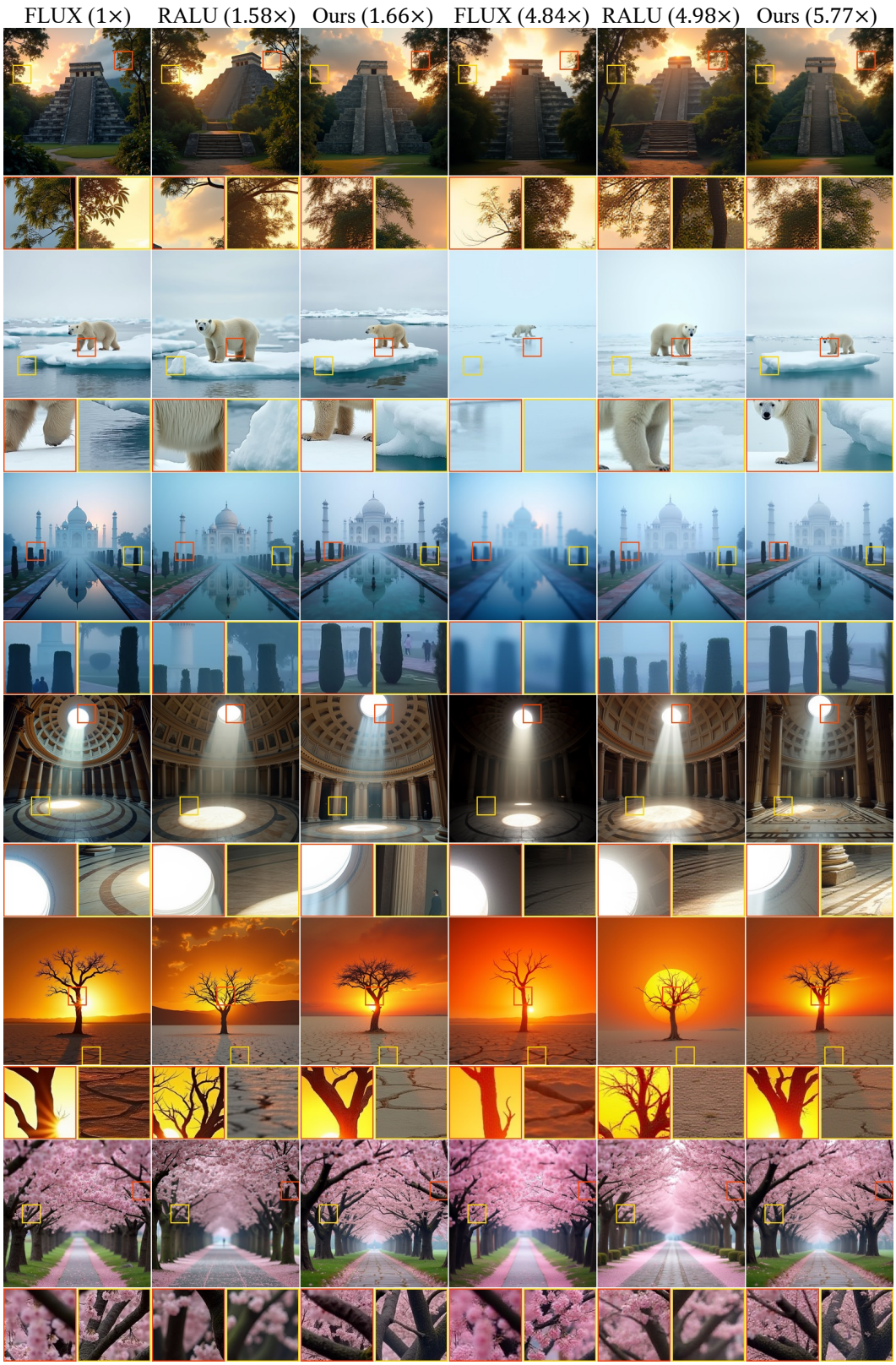

Figure 6: **Qualitative comparisons on latent-space image generation.** We compare our method against default-step generation, reduced-step native-resolution generation on FLUX.1-dev [39], and RALU [32], a state-of-the-art acceleration baseline matched to similar speedups. Our method outperforms both baselines. FLUX.1-dev with reduced steps degrades image quality and exhibits over-saturation artifacts, while RALU introduces noticeable noise artifacts.

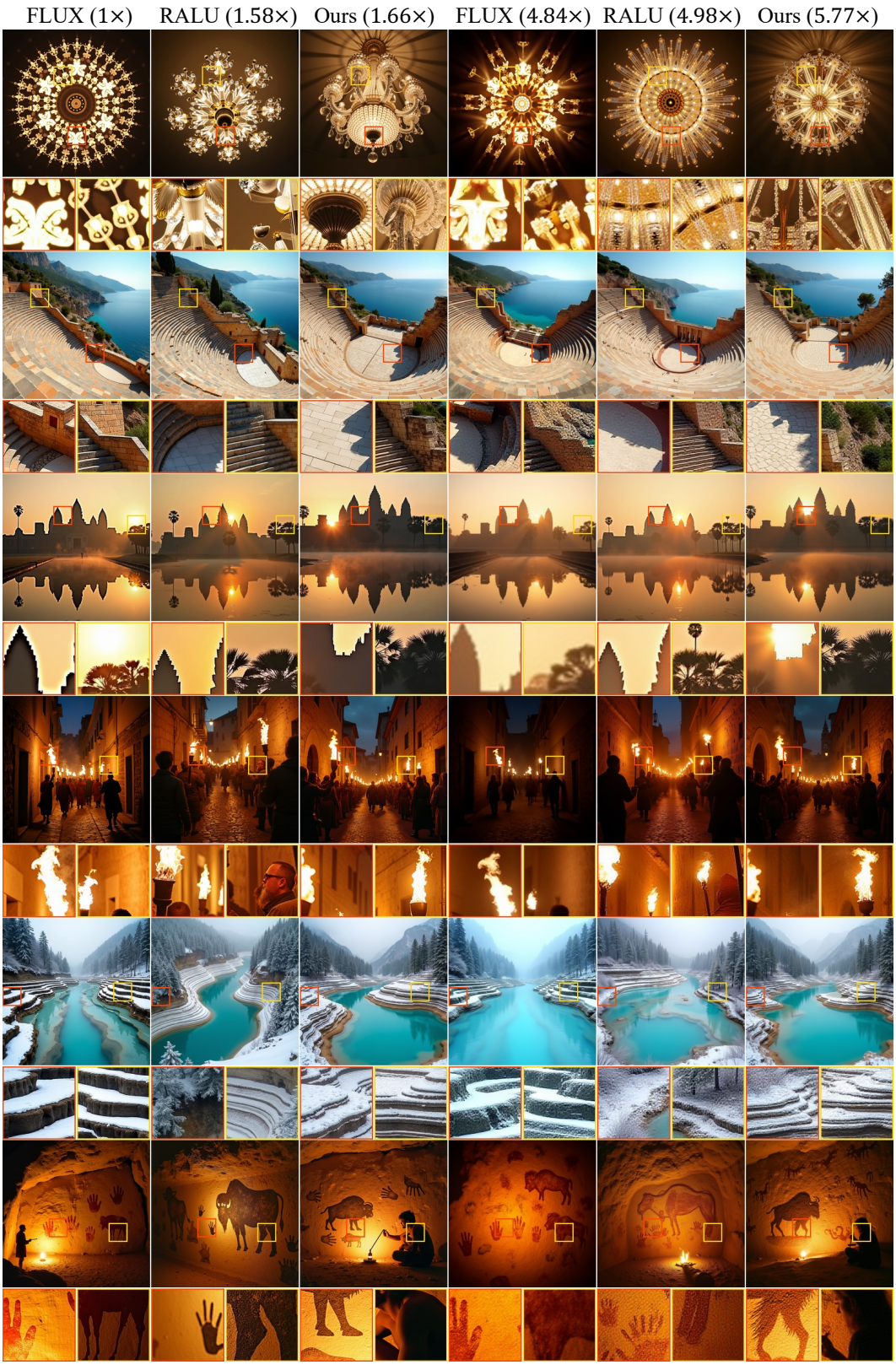

Figure 7: **Qualitative comparisons on latent-space image generation.** We compare our method against default-step generation, reduced-step native-resolution generation on FLUX.1-dev [39], and RALU [32], a state-of-the-art acceleration baseline matched to similar speedups. Our method outperforms both baselines. FLUX.1-dev with reduced steps degrades image quality and exhibits over-saturation artifacts, while RALU introduces noticeable noise artifacts.

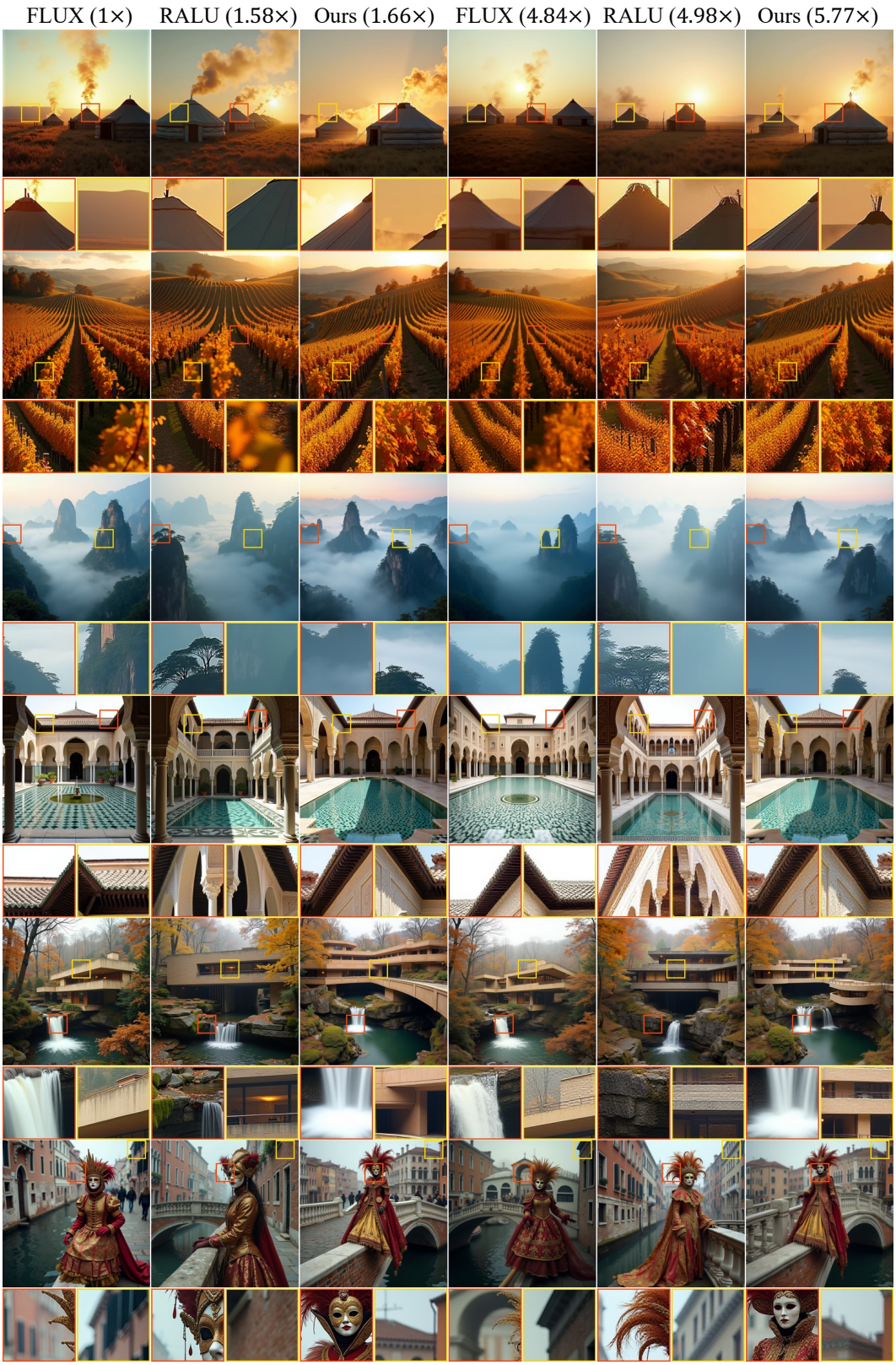

Figure 8: **Qualitative comparisons on latent-space image generation.** We compare our method against default-step generation, reduced-step native-resolution generation on FLUX.1-dev [39], and RALU [32], a state-of-the-art acceleration baseline matched to similar speedups. Our method outperforms both baselines. FLUX.1-dev with reduced steps degrades image quality and exhibits over-saturation artifacts, while RALU introduces noticeable noise artifacts.

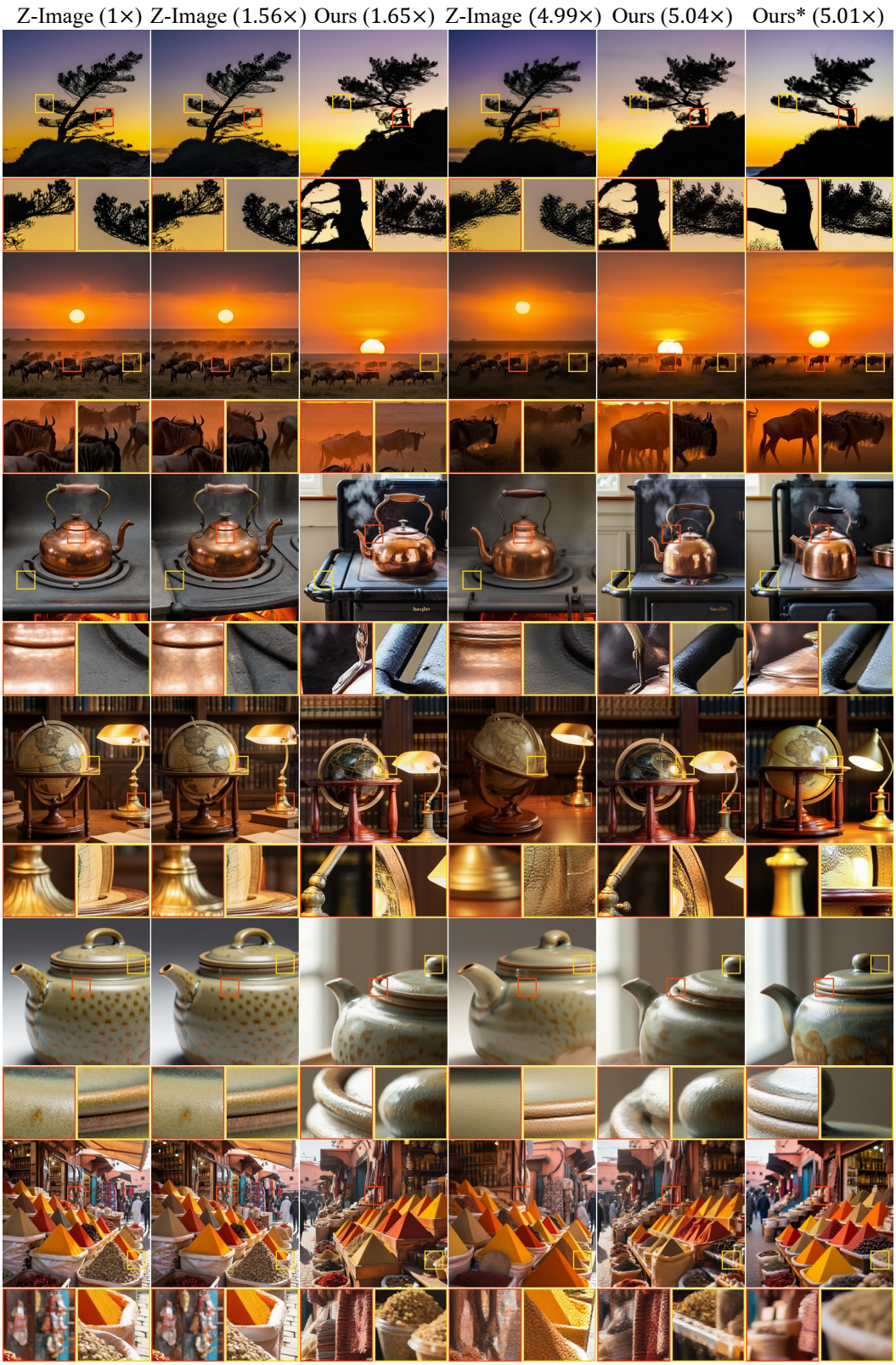

Figure 9: **Qualitative comparisons on latent-space image generation (fine-tuned).** We compare our method against default-step generation, reduced-step native-resolution generation on Z-Image [5] matched to similar speedups. Our fine-tuned model (Ours*) achieves even higher image quality compared to our training-free acceleration variant and outperforms the reduced-step baseline.

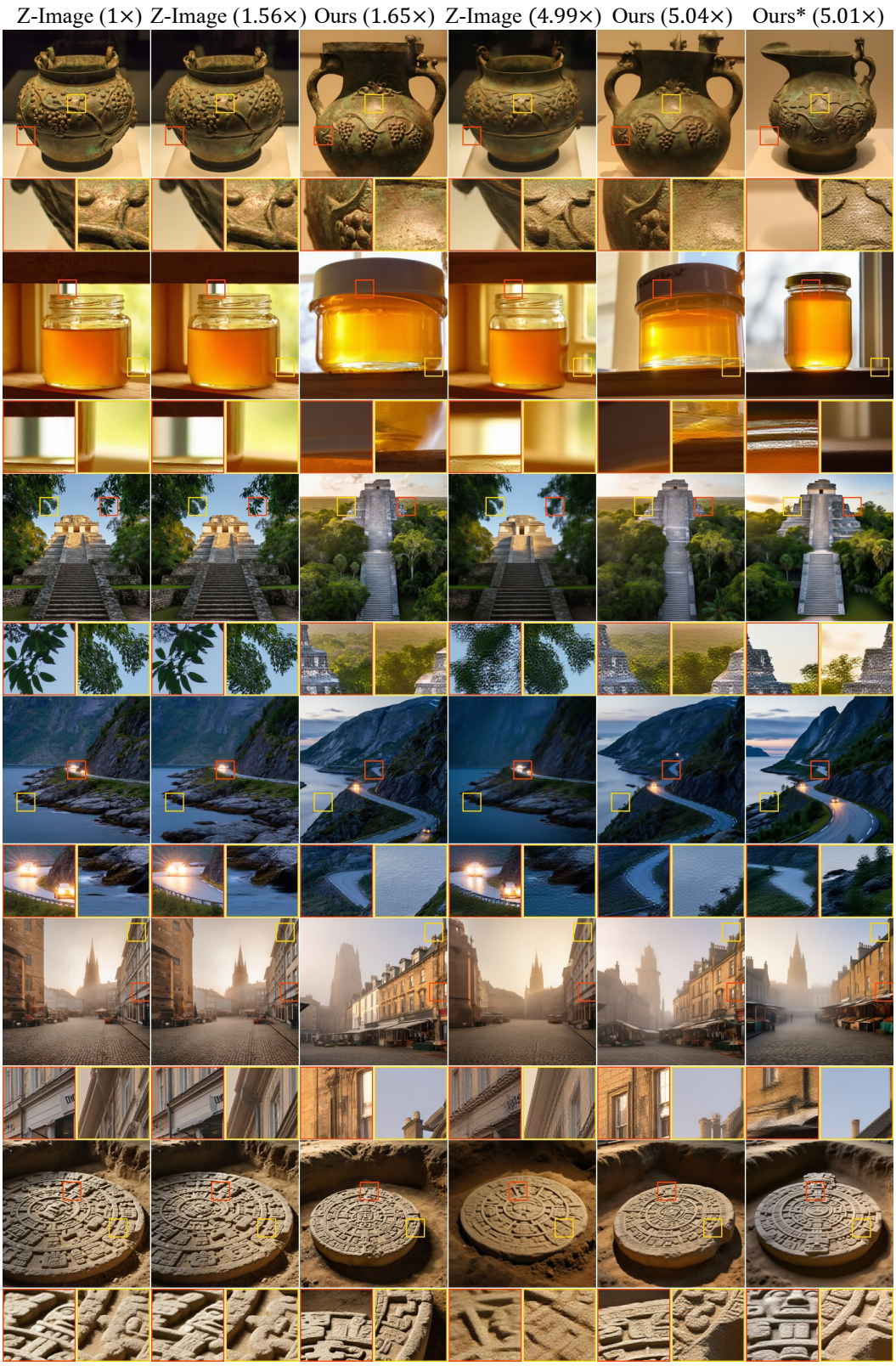

Figure 10: **Qualitative comparisons on latent-space image generation (fine-tuned).** We compare our method against default-step generation, reduced-step native-resolution generation on Z-Image [5] matched to similar speedups. Our fine-tuned model (Ours*) achieves even higher image quality compared to our training-free acceleration variant and outperforms the reduced-step baseline.

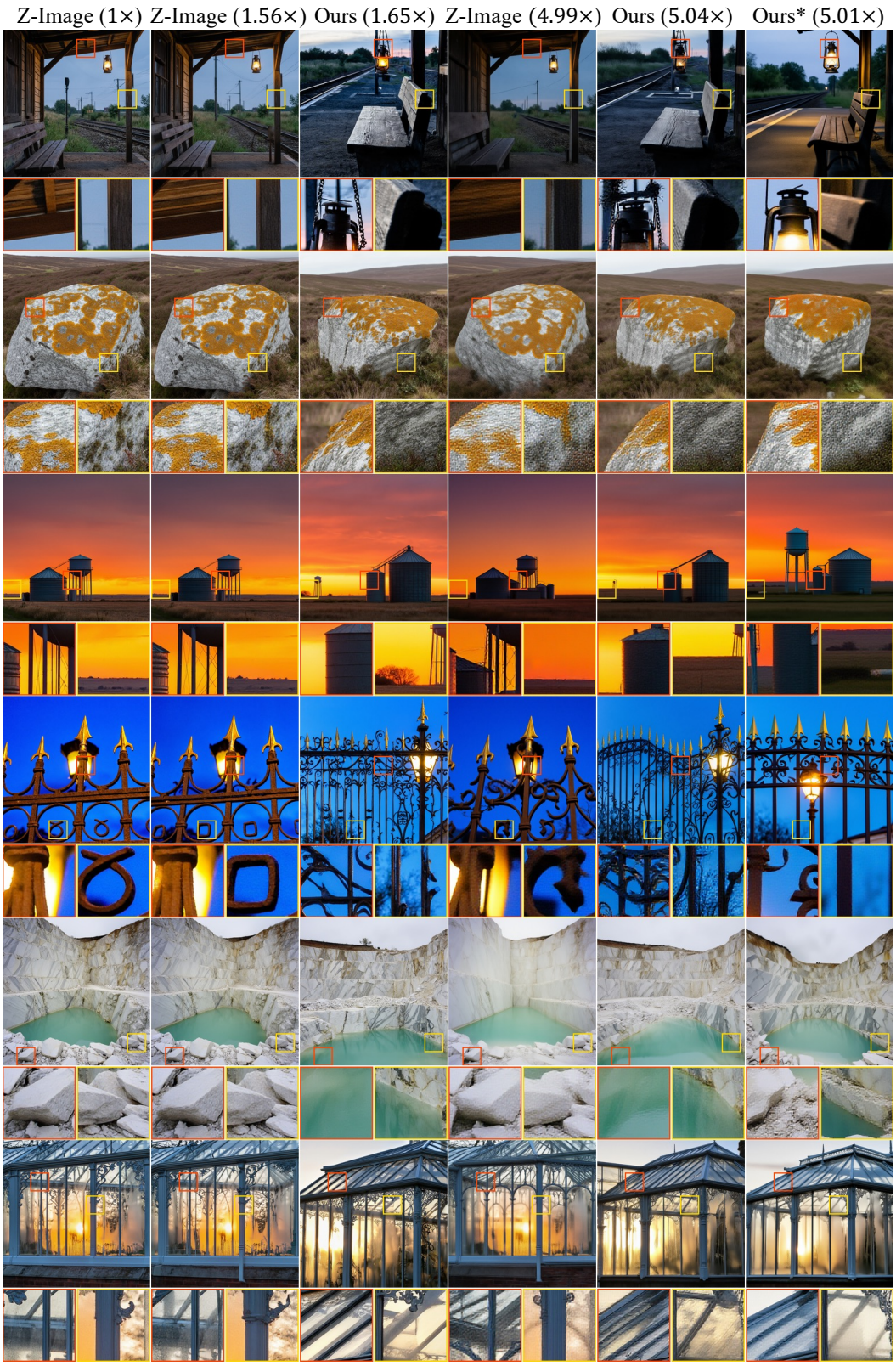

Figure 11: **Qualitative comparisons on latent-space image generation (fine-tuned).** We compare our method against default-step generation, reduced-step native-resolution generation on Z-Image [5] matched to similar speedups. Our fine-tuned model (Ours*) achieves even higher image quality compared to our training-free acceleration variant and outperforms the reduced-step baseline.

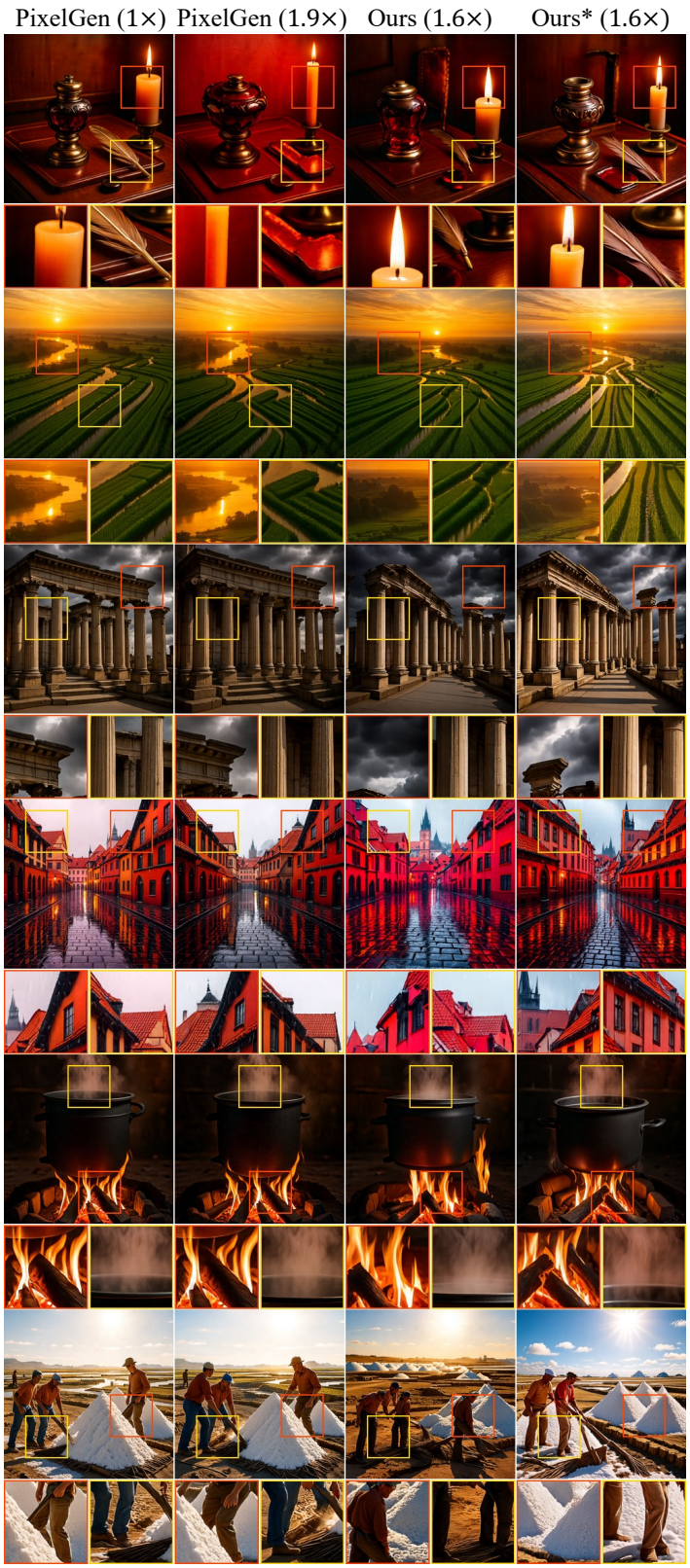

Figure 12: **Qualitative comparisons on pixel-space image generation.** We compare our method against default-step generation and reduced-step native-resolution generation on PixelGen [55], matched to comparable speedups. An asterisk (Ours*) marks the fine-tuned model. Our method achieves similar quality to full-resolution generation while attaining higher speedups.

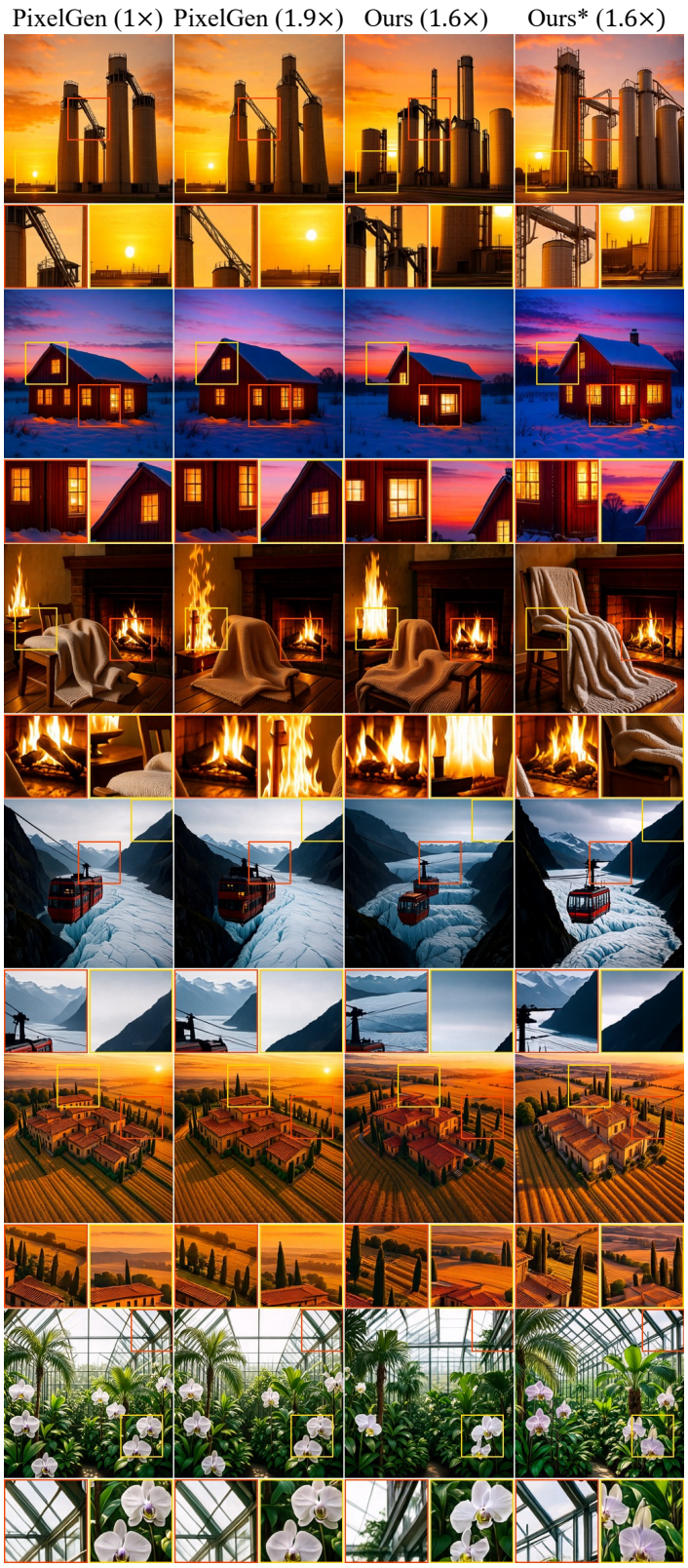

Figure 13: **Qualitative comparisons on pixel-space image generation.** We compare our method against default-step generation and reduced-step native-resolution generation on PixelGen [55], matched to comparable speedups. An asterisk (Ours*) marks the fine-tuned model. Our method achieves similar quality to full-resolution generation while attaining higher speedups.

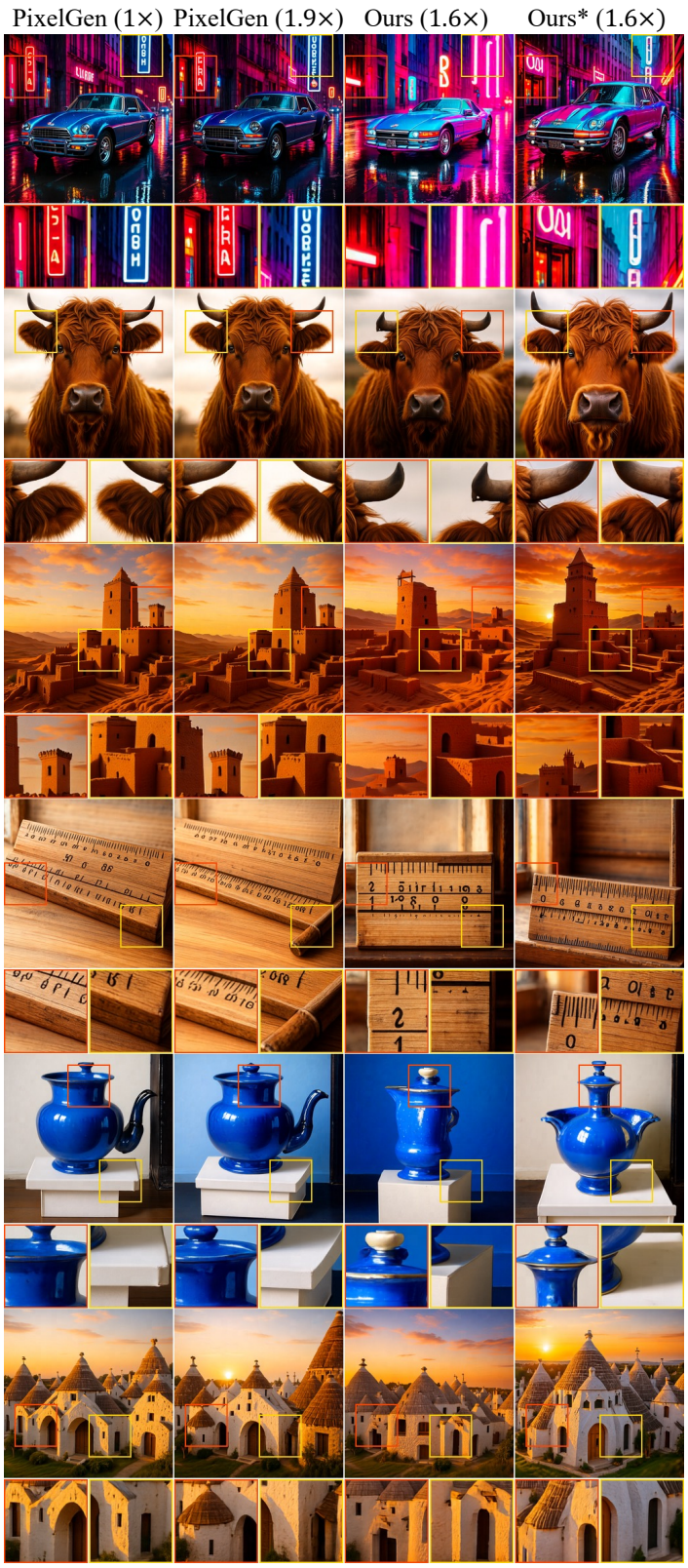

Figure 14: **Qualitative comparisons on pixel-space image generation.** We compare our method against default-step generation and reduced-step native-resolution generation on PixelGen [55], matched to comparable speedups. An asterisk (Ours*) marks the fine-tuned model. Our method achieves similar quality to full-resolution generation while attaining higher speedups.

E Additional Video Generation Results

Training-free acceleration of latent-space video generation. For training-free inference acceleration of video generation, we use our internal cluster of NVIDIA H100 GPUs with 80 GB of VRAM. For quantitative evaluation, we generate 200 videos using 200 test prompts and report the standard VBench scores across six evaluation dimensions, including Subject Consistency, Background Consistency, Motion Smoothness, Dynamic Degree, Aesthetic Quality, and Image Quality, in Table 3. All videos are generated in 720P resolution. For additional video results and qualitative comparisons, please see the project website at `https://howardxiao.ca/speed/`.

F Extended Ablation Studies

We perform extended ablation studies on latent-space image generation. Specifically, we vary the error tolerance δ, the number of resolution stages S, and the spectral transformation T_Φ, and report wall-clock latency, speedup, and ImageReward in Table 8, with qualitative comparisons in Figures 15 to 17.

Computational efficiency and scaling. The inference cost of Spectral Progressive Diffusion is determined by the number of denoising steps and the token count processed at each resolution stage. For a fixed pretrained model and resolution schedule, the per-sample cost is independent of the dataset size, so generating or evaluating more samples scales linearly with the number of samples. The spectral transformation and noise-expansion operations add only lightweight per-transition overhead compared to the DiT forward passes.

Spectral Transformation T_Φ. In Fig. 15, we observe that the Fourier Transform (FFT) tends to over-smooth results, whereas DCT and DWT achieve similar high-fidelity performance. We default to DCT for its native support of non-power-of-two resolution scaling.

Error tolerance δ. In Fig. 16, we observe that increasing δ improves efficiency, but results in ghosting and halo artifacts near detailed edges.

Resolution stages S. In Fig. 17, we find that increasing S leads to marginal speedup improvements and little image quality degradation. Since the primary speedup derives from bypassing native high-resolution steps, $S = 2$ achieves the largest speedup while higher S provides diminishing efficiency gains by further optimizing the already-inexpensive low-resolution trajectory.

Table 8: **Ablation studies on the spectral transformation T_Φ, error tolerance δ, and number of resolution stages S.** Speedup is calculated relative to the 7-step FLUX.1-dev baseline wall-clock runtime on a single H100 GPU.

T_Φ	δ	S	Latency (s) ↓	Speedup (s) ↑	ImageReward ↑
FLUX (7 steps)	-	-	2.08	1.00×	0.920
DCT	0.001	2	**2.08**	**1.00**×	1.032
DWT	0.001	2	2.08	**1.00**×	**1.038**
FFT	0.001	2	2.08	**1.00**×	0.526
DCT	0.001	3	**2.05**	**1.01**×	**1.034**
DWT	0.001	3	2.06	**1.01**×	**1.034**
FFT	0.001	3	2.06	**1.01**×	-0.371
DCT	0.001	4	**2.05**	**1.02**×	**1.008**
DWT	0.001	4	2.05	1.01×	1.002
FFT	0.001	4	2.06	1.01×	-1.178
DCT	0.002	2	**2.07**	**1.01**×	**1.032**
DWT	0.002	2	2.08	1.00×	1.031
FFT	0.002	2	2.08	1.00×	0.526
DCT	0.002	3	**2.05**	**1.02**×	**1.034**
DWT	0.002	3	2.06	1.01×	**1.034**
FFT	0.002	3	2.06	1.01×	-0.371
DCT	0.002	4	**2.05**	**1.02**×	**1.008**
DWT	0.002	4	2.05	1.01×	1.002
FFT	0.002	4	2.06	1.01×	-1.178
DCT	0.005	2	**1.91**	**1.09**×	1.019
DWT	0.005	2	1.92	**1.09**×	**1.026**
FFT	0.005	2	1.92	1.08×	0.688
DCT	0.005	3	**1.87**	**1.12**×	**1.009**
DWT	0.005	3	1.88	1.11×	**1.009**
FFT	0.005	3	1.88	1.11×	-0.207
DCT	0.005	4	**1.87**	**1.12**×	1.005
DWT	0.005	4	1.87	1.11×	**1.008**
FFT	0.005	4	1.87	1.11×	-0.993
DCT	0.01	2	**1.75**	**1.19**×	**1.039**
DWT	0.01	2	1.76	1.18×	1.025
FFT	0.01	2	1.76	1.18×	0.769
DCT	0.01	3	**1.71**	**1.22**×	1.015
DWT	0.01	3	1.72	1.21×	**1.018**
FFT	0.01	3	1.72	1.21×	0.156
DCT	0.01	4	**1.71**	**1.22**×	**1.014**
DWT	0.01	4	1.71	**1.22**×	1.006
FFT	0.01	4	1.72	1.21×	-0.612
DCT	0.02	2	**1.60**	**1.30**×	1.009
DWT	0.02	2	1.61	**1.30**×	**1.010**
FFT	0.02	2	1.61	**1.30**×	0.807
DCT	0.02	3	**1.53**	**1.36**×	**0.967**
DWT	0.02	3	1.53	**1.36**×	0.966
FFT	0.02	3	1.53	**1.36**×	0.243
DCT	0.02	4	**1.53**	**1.36**×	**0.964**
DWT	0.02	4	1.53	**1.36**×	0.955
FFT	0.02	4	1.53	**1.36**×	-0.462
DCT	0.05	2	**1.44**	**1.44**×	0.989
DWT	0.05	2	1.45	**1.44**×	**0.991**
FFT	0.05	2	1.45	1.43×	0.715
DCT	0.05	3	**1.34**	**1.55**×	0.936
DWT	0.05	3	1.34	**1.55**×	**0.937**
FFT	0.05	3	1.35	1.54×	0.227
DCT	0.05	4	**1.33**	**1.56**×	**0.918**
DWT	0.05	4	1.34	**1.56**×	0.902
FFT	0.05	4	1.34	1.55×	-0.511
DCT	0.1	2	**1.29**	**1.62**×	0.966
DWT	0.1	2	1.29	**1.62**×	**0.977**
FFT	0.1	2	1.30	1.61×	0.694
DCT	0.1	3	**1.15**	**1.81**×	0.880
DWT	0.1	3	1.17	1.78×	**0.892**
FFT	0.1	3	1.17	1.79×	0.268
DCT	0.1	4	**1.15**	**1.82**×	0.864
DWT	0.1	4	1.15	1.81×	**0.867**
FFT	0.1	4	1.16	1.80×	-0.281

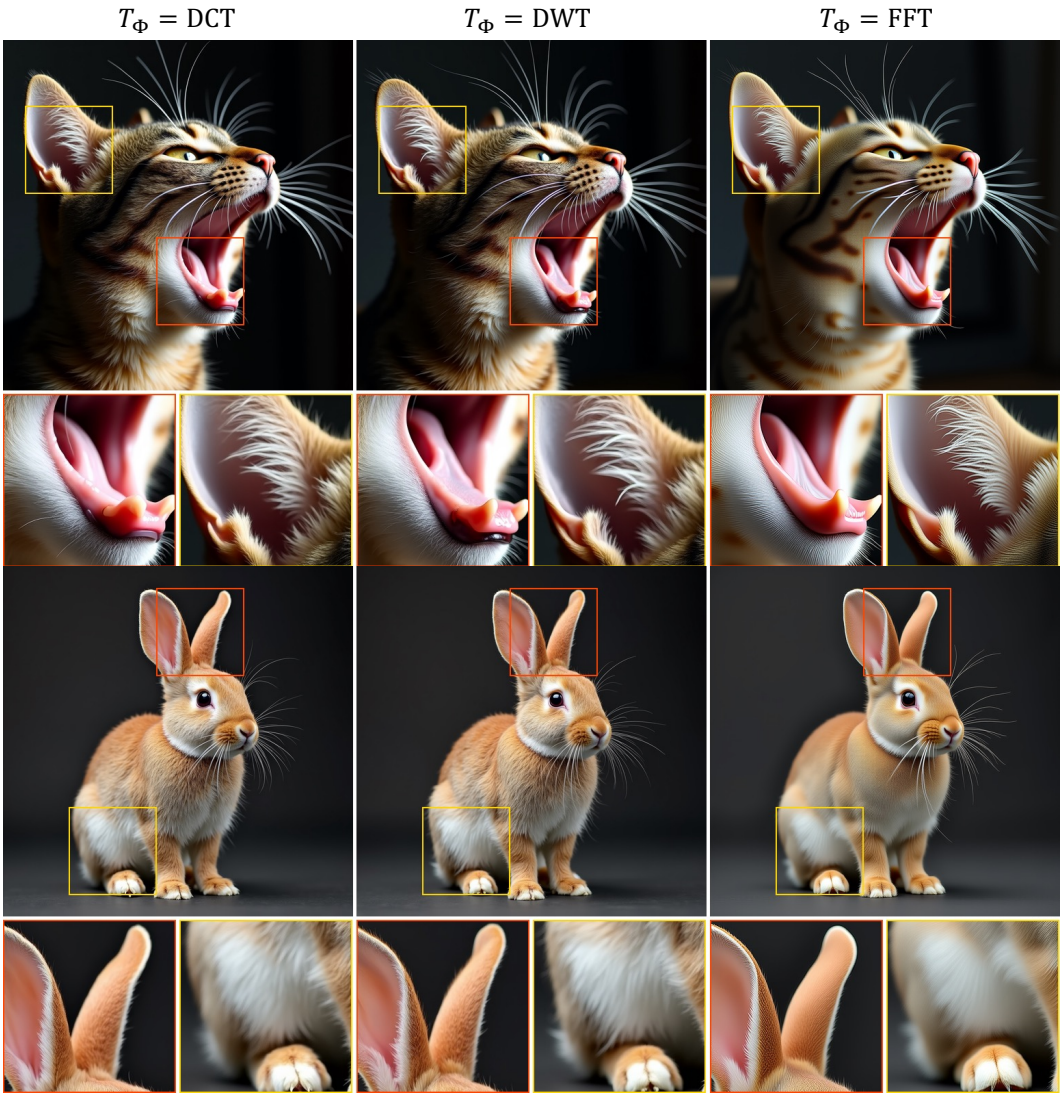

Figure 15: **Qualitative ablation on T_Φ.** We see that FFT leads to overly smooth results while DCT and DWT attain similar image quality.

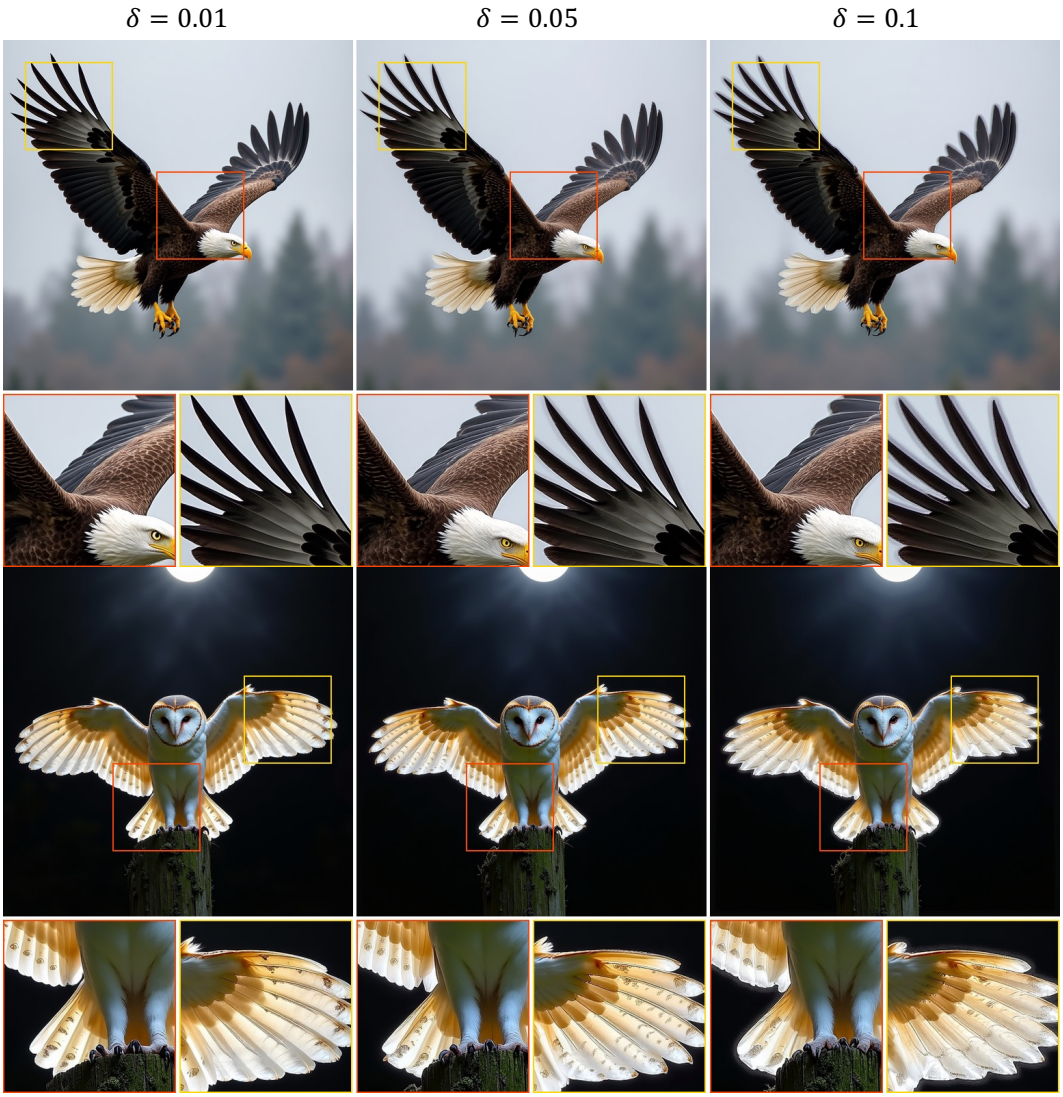

Figure 16: **Qualitative ablation on δ.** We observe that increasing δ improves efficiency, but results in ghosting and halo artifacts near detailed edges.

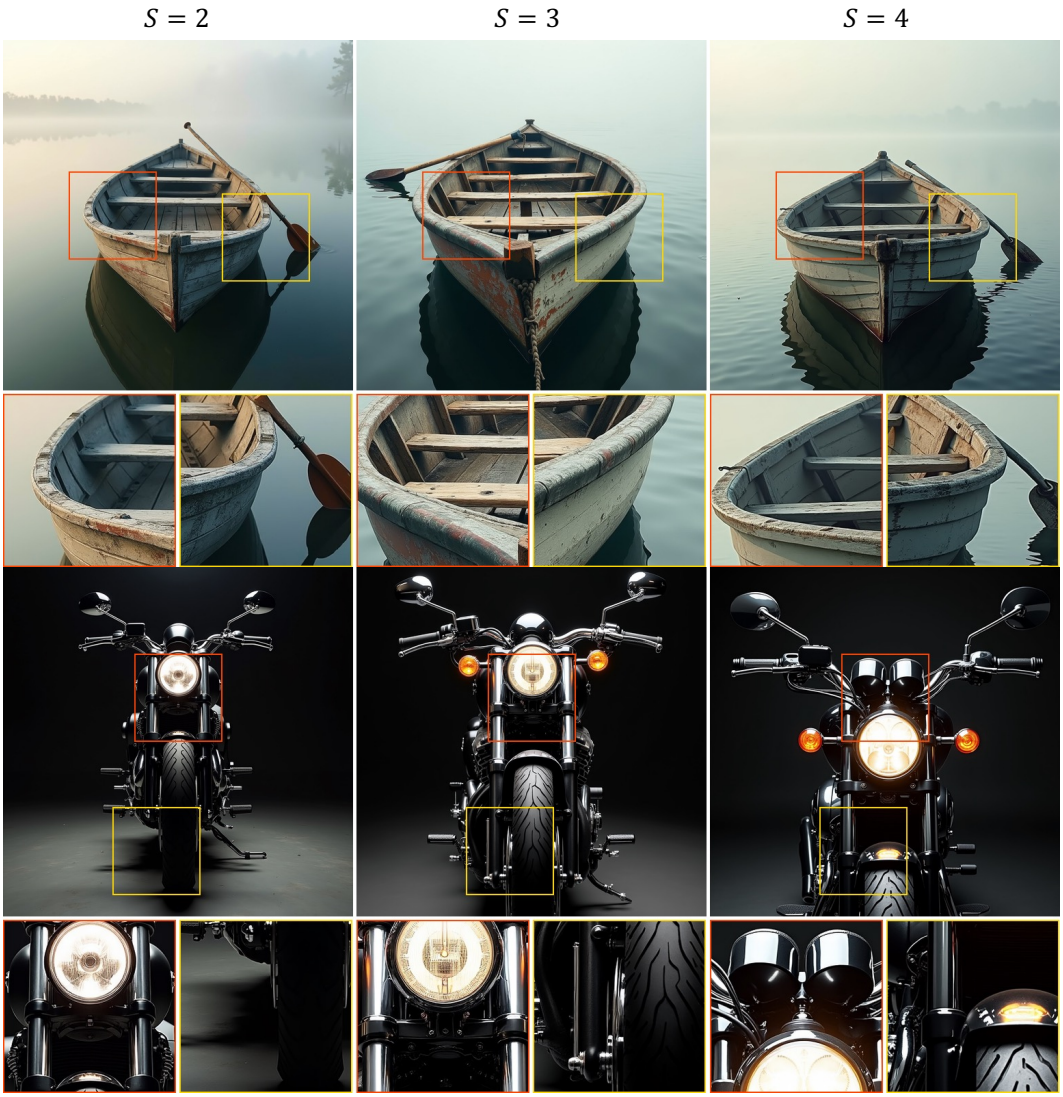

Figure 17: **Qualitative ablation on S.** We find that increasing S leads to marginal speedup improvements and little image quality degradation.

G Additional Frequency-based Image Editing Details

Detailed Procedure of Frequency-based Image Editing. In our frequency-based editing pipeline, we perform the following operations given an input image $\mathbf{x}_{\text{in}}$, resolution scales $s_{1:S}$, and transition times $t_{1:S-1}$. We assume that the image editing process starts from transition time t_k, with $k \in \{1, 2, \ldots, S-1\}$.

1. Compute the spectrum of the input image $\boldsymbol{\xi}_{\text{in}} = T_\Phi(\mathbf{x}_{\text{in}})$.
2. Extract the low-frequency part of $\boldsymbol{\xi}_{\text{in}}$ corresponding to the representable spectrum Ω_{s_k} at scale s_k.
3. Fill the spectrum $\Omega_{s_{k+1}} \setminus \Omega_{s_k}$ with $t_k\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.
4. Convert the expanded spectrum $\Omega_{s_{k+1}}$ back to the spatial domain $\mathbf{x}_{t_k}^{s_{k+1}}$ via T_Φ^{-1} and continue denoising starting from the transition time t_k.

The total time of denoising is thus $1 - t_k$.

We conduct image editing experiments using Z-Image [5] as the backbone, evaluating our frequency-domain image editing pipeline against the SDEdit baseline [56]. We set $S = 2$ and $\delta = 0.01$.

Extended Qualitative Results. We show extended image editing results on two tasks: *texture editing* and *artistic stylization*. We configure our method with $\delta = 0.01$ and $S = 2$, which corresponds to a resolution transition at step 26 of the default 50-step denoising trajectory. We set $T_\Phi = \text{DCT}$, and evaluate the effect of different spectral transforms in the next section. For the SDEdit baseline, we initialize the process by corrupting the clean source image to the equivalent noise level at step 26 and subsequently proceed with the denoising schedule. This setting accurately reflects the editing method's capability to inject textural or style information, which are considered high-frequency details, in later denoising steps.

As shown in Figures 18 to 22, our frequency-based editing method achieves superior prompt alignment and geometric consistency compared to SDEdit, successfully changing the texture and artistic style of the input image while retaining the topological structure of the input. While SDEdit-style editing at earlier timesteps (higher noise levels at step 17) can facilitate stylistic changes, it introduces significant structural drift and geometric divergence from the source image. Furthermore, starting from early timesteps incurs substantial computational overhead and requires a brittle renoising to a specific, difficult-to-tune initialization timestep to balance editability with structural fidelity.

Effect of T_Φ on Image Editing. As shown in Fig. 23, FFT-based editing results in overly-smooth results as well as block noise artifacts. DCT and DWT-based editing achieve similar editing quality.

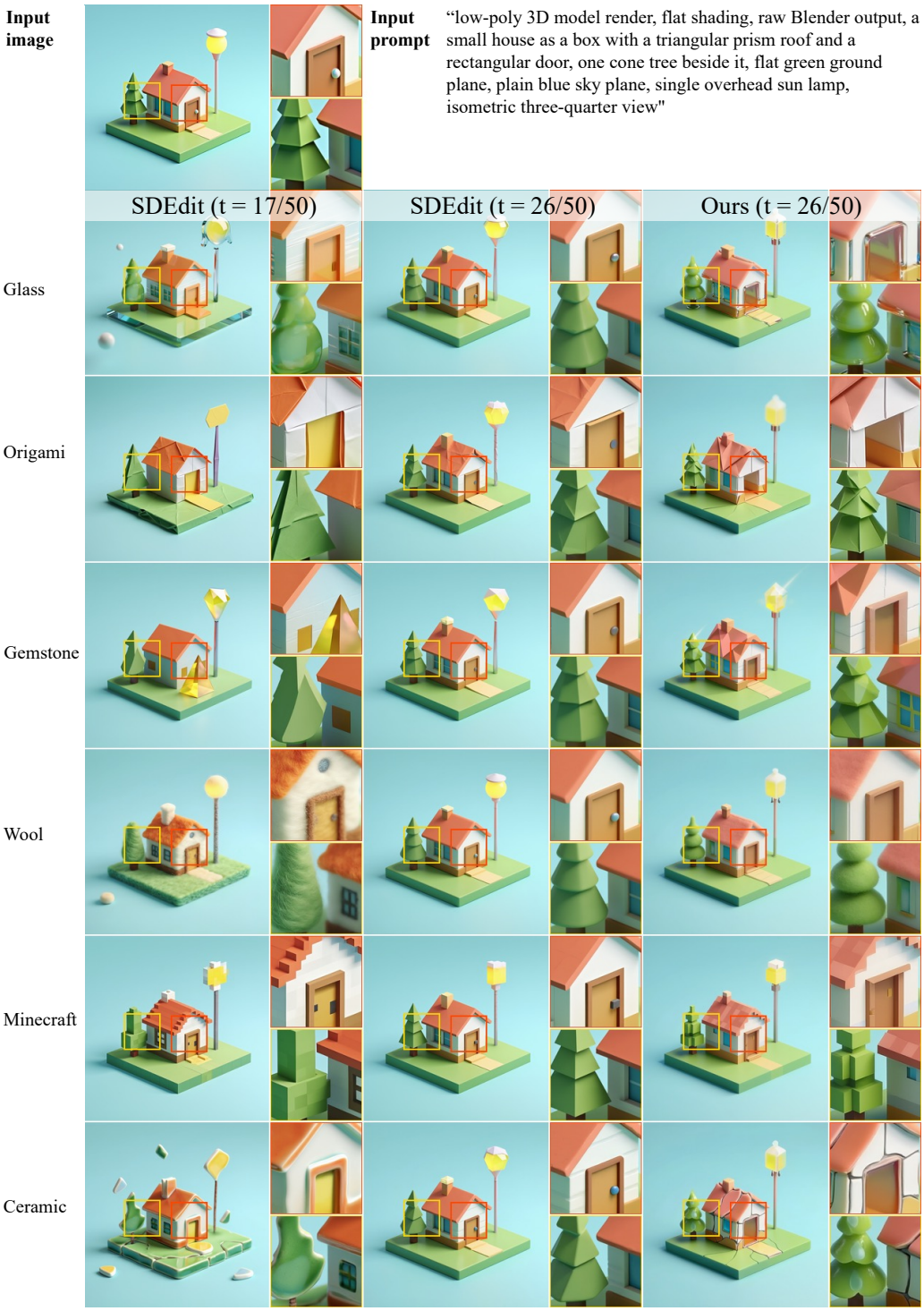

Figure 18: **Texture editing results.** Our frequency-based editing framework outperforms SDEdit, enabling high-fidelity texture transfer while preserving the geometric structure of the input image.

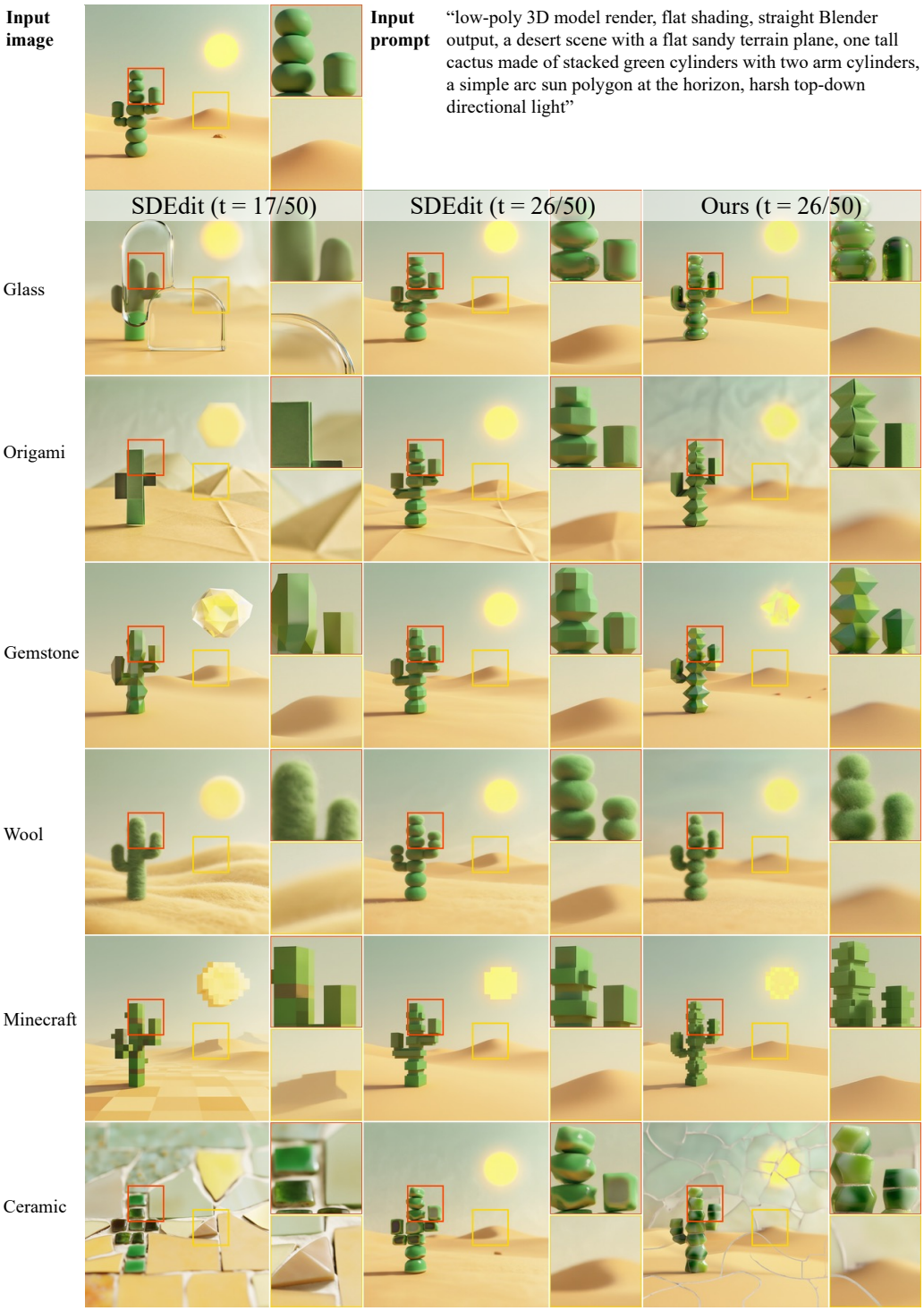

Figure 19: **Texture editing results.** Our frequency-based editing framework outperforms SDEdit, enabling high-fidelity texture transfer while preserving the geometric structure of the input image.

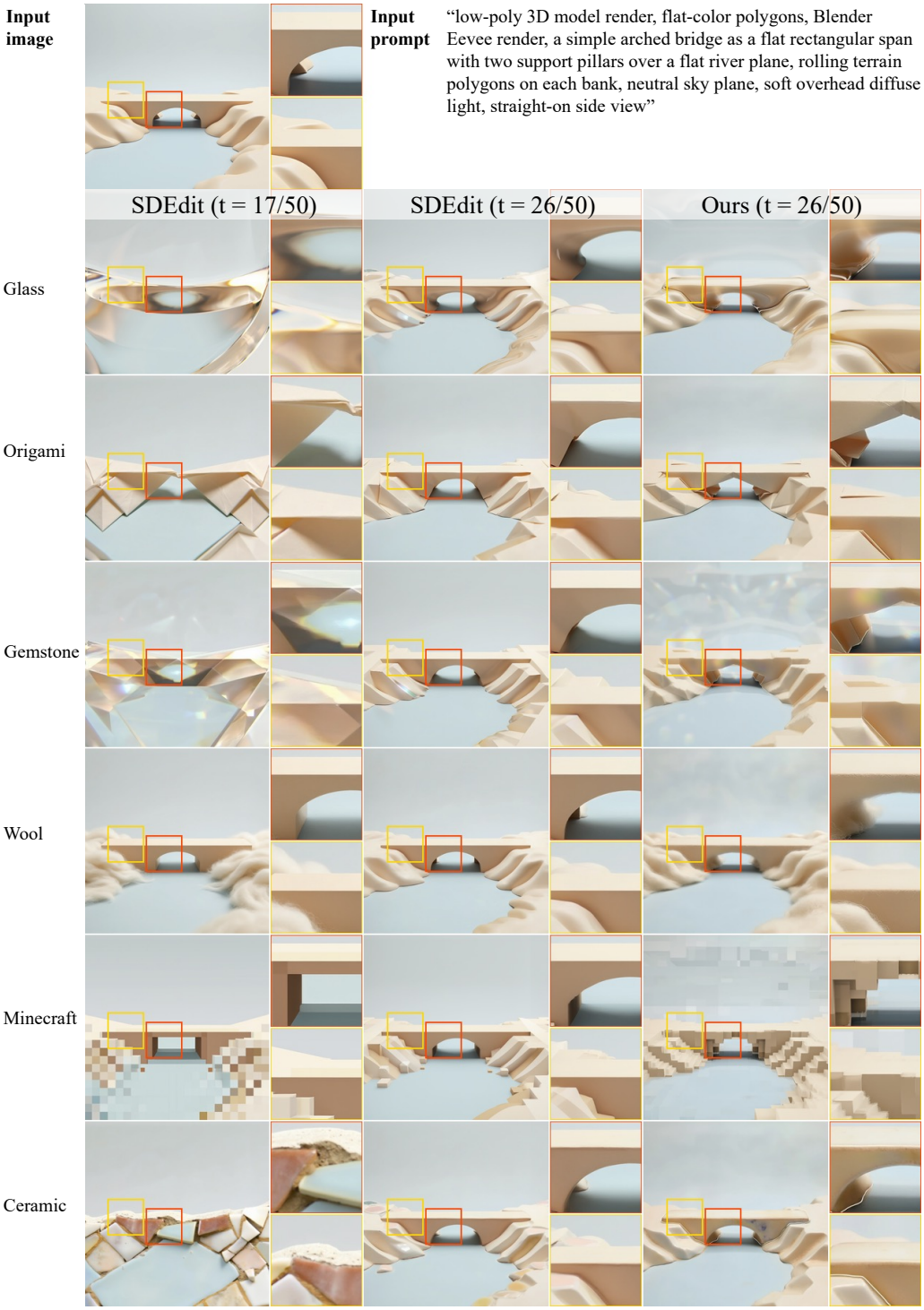

Figure 20: **Texture editing results.** Our frequency-based editing framework outperforms SDEdit, enabling high-fidelity texture transfer while preserving the geometric structure of the input image.

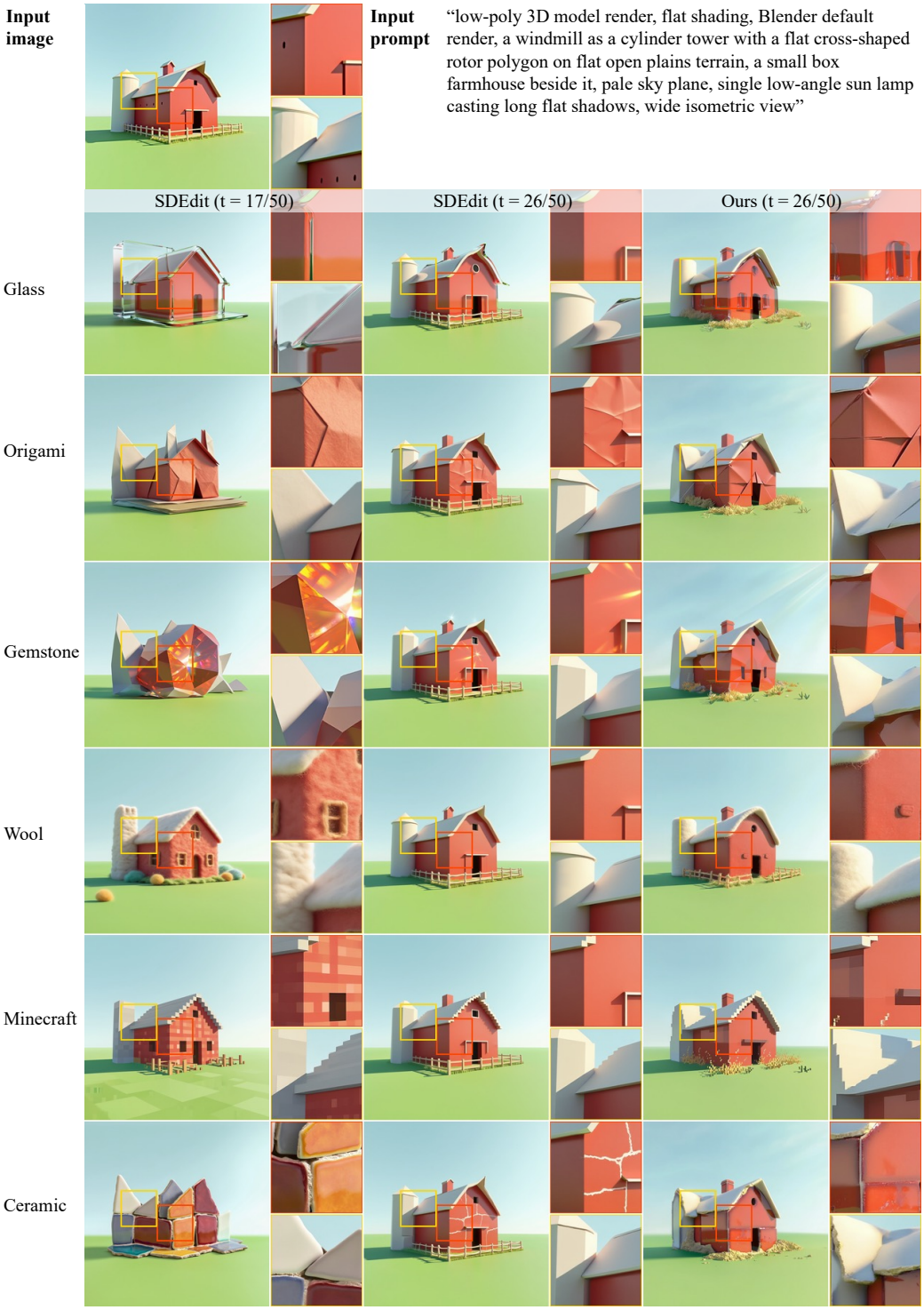

Figure 21: **Texture editing results.** Our frequency-based editing framework outperforms SDEdit, enabling high-fidelity texture transfer while preserving the geometric structure of the input image.

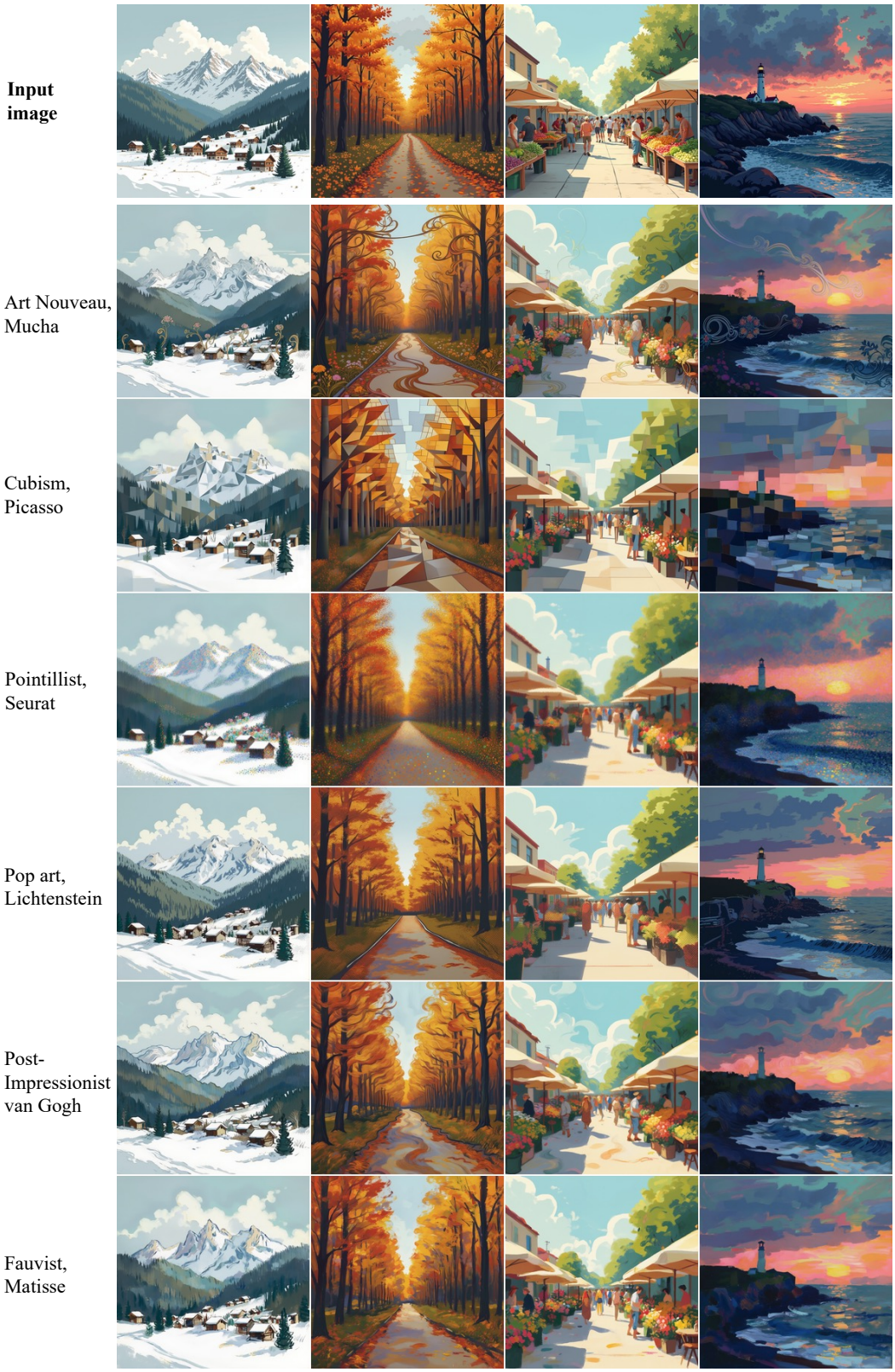

Figure 22: **Artistic stylization results.** Aside from texture editing, our frequency-based editing approach also supports artistic stylization given stylistic descriptions and a representative artist.

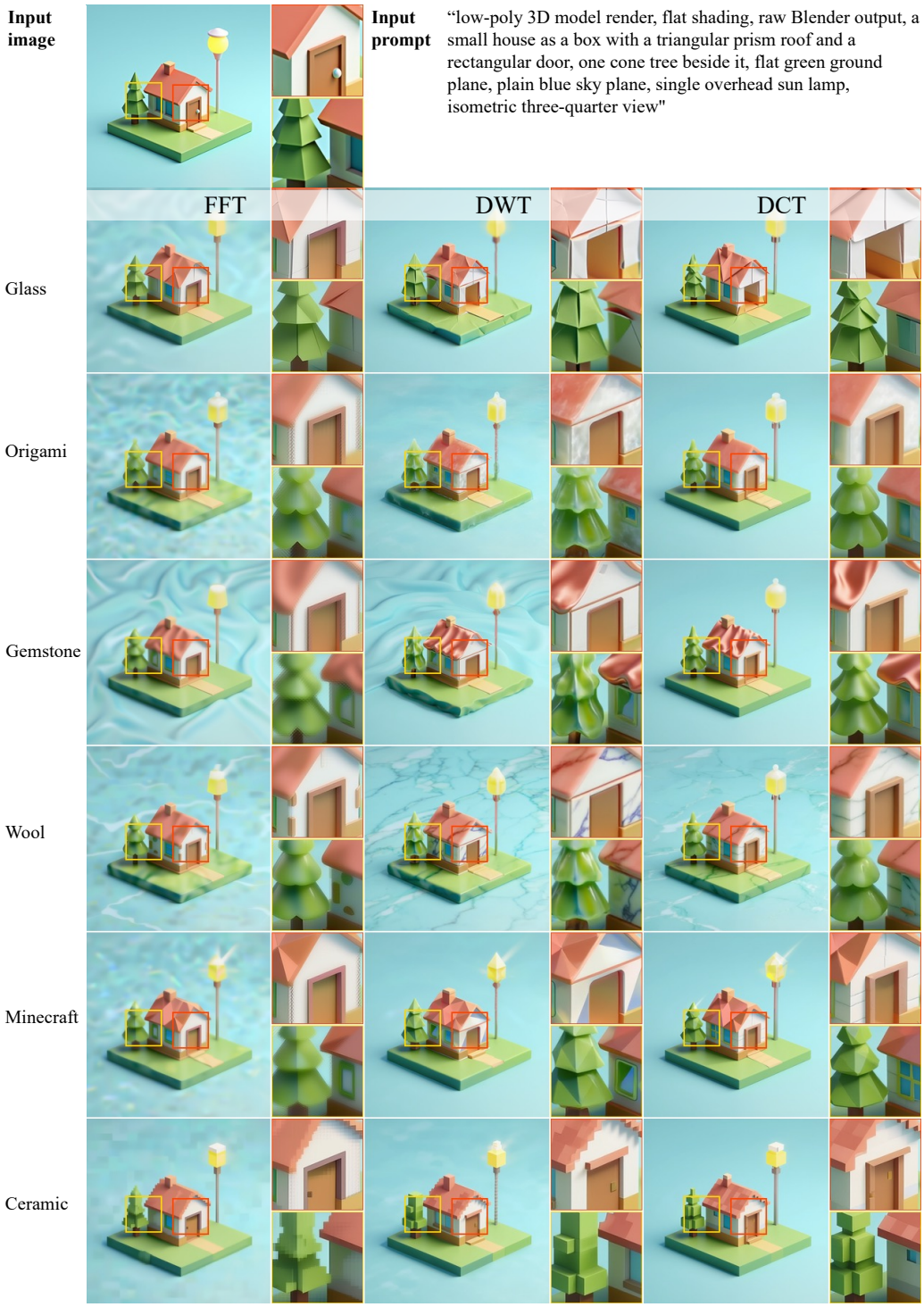

Figure 23: **Effect of T_Φ on image editing.** FFT–based editing leads to overly-smooth and hazy results; DCT- and DWT-based editing achieve similar editing quality.

H Broader Impacts

Spectral Progressive Diffusion is a principled method for improving the efficiency of pretrained image and video generation models, rather than a system targeted at a particular deployment domain. Its main positive impact is to reduce the compute, latency, and energy cost of generation, which can make experimentation and creative workflows more accessible and lower the environmental footprint of repeated generation. At the same time, because the method can accelerate existing generative models, we acknowledge that it could also lower the cost of producing synthetic media for undesirable uses such as disinformation, impersonation, or spam, depending on the pretrained model and release setting. We do not release new datasets or generation services, and any future release of fine-tuned adapters should follow the access terms, safeguards, and usage restrictions of the underlying pretrained models.

I Existing Assets and Licenses

Our experiments build on publicly available pretrained models, datasets, evaluation benchmarks, and software packages, including FLUX.1-dev [39], Z-Image [5], PixelGen [55], WAN [75], MS-COCO [46], GenEval [18], T2I-CompBench [28], VBench [31], and the evaluation metrics used in Sec. 5, including ImageReward [86], CLIP-IQA [82], and NIQE [57]. We cite the original sources for these assets and use them under their respective licenses and access terms. The named licenses and access terms are: FLUX.1-dev is released under the FLUX.1 [dev] Non-Commercial License; Z-Image, PixelGen, WAN 2.1, VBench, ImageReward, VChitect-T2V-Dataverse, and the PyIQA implementation used for NIQE are under the Apache License 2.0; GenEval and T2I-CompBench are under the MIT License; and CLIP-IQA is under the NTU S-Lab License 1.0. MS-COCO annotations are under Creative Commons Attribution 4.0 International (CC BY 4.0), while its images follow their original Flickr licenses; the public page for Aesthetic-Train-V2 does not list an explicit license, so we use it only under its stated access terms and do not redistribute the dataset.