UNLOCKING FEW-STEP DIFFUSION FOR FAITHFUL PREVIEWS

Anonymous authors
Paper under double-blind review

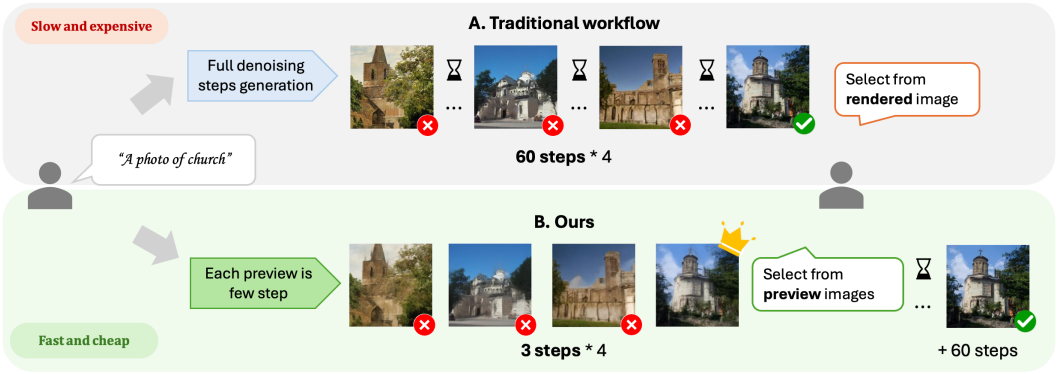


Figure 1: **Traditional vs. preview-then-refinement workflow.** Traditional generation requires full denoising steps (e.g. 60 steps) before selection, making rejected candidates expensive. Our enhancement module enables a faithful 3-step preview for cheap selection, with full generation performed only for selected candidates.

ABSTRACT

Sampling latency compounds in diffusion workflows, where users generate and discard many candidates before keeping one. Surprisingly, the poor outputs of standard few-step samplers do not reflect a lack of reconstruction capacity: by optimizing only the initial noise, frozen 3–4-step samplers can closely reproduce their corresponding full-step outputs. Building on this finding, we learn corrections to the initial noise and denoising updates using endpoint supervision, improving correspondence with full-step outputs generated from the same noise and prompt. The resulting previews allow users to screen candidates cheaply and reserve full-step generation for promising ones. Input correction also transfers across sampling budgets without retraining. Experiments show substantial improvements in reference fidelity, including 53–78% lower reconstruction MSE than retrained LD3 on unconditional benchmarks, alongside improved ranking preservation and candidate selection on SD1.5, SDXL, and FLUX.1-dev. Code and model checkpoints are available at `https://anonymous.4open.science/r/faithful-previews-anonymous-F2EB/`.

1 INTRODUCTION

Diffusion models generate high-quality images, but users often wait through round after round of sampling before finding the right one. Each image takes tens to hundreds of denoising steps to generate (Podell et al., 2024; Esser et al., 2024; Z-Image Team et al., 2026; Black Forest Labs et al., 2025), and that cost compounds over every round of the search. Ideally, users could quickly preview candidates and reserve full-step generation for the images they choose to keep, avoiding expensive sampling for candidates they would ultimately discard.

Such previews require *instance-wise correspondence*: the quick preview should be close to the full-step output when using the same initial noise and prompt. However, achieving this correspondence with only a few steps is challenging. For standard few-step solvers, coarse time discretization can

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

cause significant deviations from the full-step trajectory. For example, three- or four-step DDIM sampling often produces blurry images that differ substantially in appearance and structure from their full-step counterparts (for example, see the first and second column of Figure 3). Distilled few-step samplers can produce sharper images (Lin et al., 2024; Yin et al., 2024a;b), but improved sample quality alone does not ensure that these images match the corresponding full-step outputs. For example, DMD2 Yin et al. (2024a) removes the paired regression loss and improves generation quality through distribution-matching.

Our study was sparked by an observation from diffusion-based inverse problems (Wang et al., 2024; Jia et al., 2026). A diffusion prior that on its own generates blurry or out-of-domain samples can still recover a corrupted image very well through suitable optimization of the initial noise. While their objective differs from ours, their results hint at an unexplored possibility for few-step samplers:

The blurry outputs of naive few-step sampling may substantially underestimate what the frozen sampler can generate. With a suitable adjustment to the initial noise, few-step samplers may suffice to closely reproduce the corresponding full-step image.

We first test this possibility through oracle noise optimization with access to full-step reference images. Optimizing only the initial noise of frozen samplers reduces mean reconstruction MSE by 99.1–99.5% across four unconditional models with three-step sampling, and by 98.6% for four-step FLUX.1-dev. These results encourage us to learn a shared input corrector that predicts a residual adjustment to the initial noise. We train it to make the frozen few-step sampler's output from the corrected noise match the full-step reference generated from the original noise. Across unconditional diffusion benchmarks, our method achieves significantly higher fidelity to full-step references than evaluated baselines, such as LD3 (Tong et al., 2025). It also transfers to larger sampling budgets through simple rescaling, without retraining.

To extend this approach to large-scale text-to-image models, we explore more flexible trajectory corrections through low-rank adapters. Under controlled experiments, these adapters further improve instance-wise correspondence while keeping the pretrained base weights frozen.

Returning to our central preview application, corrected few-step samplers let users quickly screen candidates and reserve full-step generation for promising ones. Experiments on SD1.5, SDXL, and FLUX.1-dev show substantial overall gains in ranking preservation and candidate selection over evaluated distilled models, learned preview solvers, and probe-based predictors.

Our contributions are threefold. We reveal a substantial gap between raw sampling quality and reconstruction capacity: even when their raw outputs are severely degraded, frozen 3–4-step samplers can closely reproduce the corresponding full-step generations through oracle optimization of only their initial noise. Building on this finding, we develop corrections that improve instance-wise correspondence between few-step and full-step generation, with input correction also generalizing across sampling budgets without retraining. The resulting corrected samplers enable fast, faithful previews, allowing users to screen candidates cheaply and reserve full-step generation for promising ones. As an unexpected byproduct, we observe that cross-backbone transfer can also improve perceptual preference scores.

Related Works: ConsistencySolver (Wang et al., 2026) learns integration coefficients through reinforcement learning, whereas we train corrections using direct endpoint supervision. LD3 (Tong et al., 2025) learns sampling schedules with training-only noise perturbations. Diffusion Probe (Huang et al., 2026) and Probe-Select (Guo et al., 2026) predict final quality scores from early model features.

More broadly, other diffusion acceleration methods share our goal of computational efficiency but serve complementary purposes. These include improved numerical solvers (Lu et al., 2022) and distillation methods (Salimans & Ho, 2022; Song et al., 2023; Lin et al., 2024; Yin et al., 2024b;a). These approaches prioritize fast, high-quality generation, which alone does not ensure faithful previews. For example, DMD2 removes the loss for instance-wise matching to teacher outputs and uses distribution-level objectives to improve sample quality (Yin et al., 2024a).

Finally, prior work has shown that refining initial noise can improve generation and inverse problem solving (Eyring et al., 2024; Ahn et al., 2026; Wang et al., 2024; Jia et al., 2026). We instead target instance-wise fidelity to a full-step teacher for fast previews.

2 METHOD

We consider deterministic diffusion samplers that map initial noise z, typically drawn from a standard Gaussian, and an optional condition c to an image. Let F_{ref} denote a fixed full-step reference (or teacher) sampler and F_{few} a few-step student sampler. Our goal is to correct F_{few} so that its output closely matches $F_{\text{ref}}(z, c)$ for the same input (z, c). In most of our experiments, the two samplers are based on the same model family (e.g., Stable Diffusion 1.5 or SDXL), although our formulation also allows different model families.

Training objective. We denote the corrected few-step map by $\widehat{F}_\phi$, where ϕ represents the trainable correction parameters. For each input (z, c), we use the reference output as the training target and minimize the error between the two outputs:

$$\min_{\phi} \mathbb{E}_{z,c}\Big[\ell\Big(\widehat{F}_\phi(z, c), F_{\text{ref}}(z, c)\Big)\Big], \qquad (1)$$

where ℓ measures image reconstruction error.

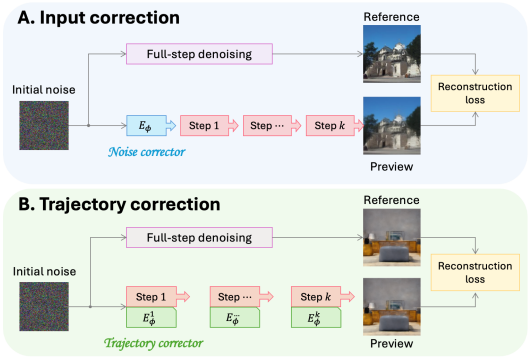


Figure 2: Input correction (A) and trajectory correction (B) for fast and faithful diffusion previews.

Training procedure. We sample inputs (z, c), compute the corresponding reference outputs $F_{\text{ref}}(z, c)$, and minimize the above objective over ϕ. Gradients are propagated through the complete few-step sampler, while the pretrained backbone remains frozen. Training requires only input–output pairs from the reference sampler. Losses, auxiliary terms, and implementation details are provided in Appendix A.

Parameterizing the correction. We consider two natural locations for correction: the initial noise and the intermediate denoising updates.

Input correction uses a lightweight network to predict a residual Δz from the initial noise z. We then add this residual to z and pass the corrected noise to the fixed few-step sampler:

$$z \longrightarrow z + \Delta z \longrightarrow F_{\text{few}}(z + \Delta z, c).$$

Intuitively, this residual adjusts the starting point in a learned direction that helps the few-step sampler better approximate the full-step output associated with the original noise.

Trajectory correction keeps the original initial noise and modifies the denoising updates through step-specific low-rank adapters (Hu et al., 2022). Each step has its own adapters, which add trainable low-rank updates to the frozen denoiser weights. The adapters are trained jointly to improve the final output of the complete sampling chain. Figure 2 illustrates the two correction strategies for a k-step sampler. For trajectory correction, we choose to supervise only the final outputs. This avoids the need for intermediate reference states and allows us to treat the reference sampler as a black box that provides input–output pairs.

Classifier-free guidance. In text-to-image diffusion, classifier-free guidance (CFG) combines conditional and unconditional denoiser predictions using a guidance scale s, which controls the strength of the text condition. Because changing s also changes the generation trajectory, we treat it as an additional input to the correction model. We encode s with Fourier features and use a learned projection to continuously modulate the correction. This conditioning is compatible with both input and trajectory correction.

Transfer across steps. Input correction learns a sample-dependent direction in the initial-noise space: adjusting the noise along this direction helps a few-step sampler match the full-step reference. We find that the learned direction remains useful at other sampling budgets without retraining the corrector. For a corrector trained with k sampling steps, we adapt it to a K-step sampler by scaling its predicted noise displacement by e.g. $\alpha_K = (k/K)^2$. For example, suppose a three-step corrector

predicts Δz such that $F_{3-\text{step}}(z + \Delta z) \approx F_{\text{ref}}(z)$. We can use the same direction for five-step sampling with input $z + (3/5)^2 \Delta z = z + 0.36\Delta z$, so that $F_{5-\text{step}}(z + 0.36\Delta z) \approx F_{\text{ref}}(z)$. This lets us reuse a trained noise corrector across sampling steps without retraining. We demonstrate the effectiveness of this approach in Section 3.2.

Inference and cheap preview. The corrected few-step sampler is itself a fast generative model. Because the corrected sampler preserves instance-wise correspondence with the full-step reference, we can naturally use its outputs as faithful previews of the reference images generated from the same inputs. Suppose a user wants to find an image they like for a prompt c. Instead of running the full-step sampler for every noise in a batch $\{z_i\}$ before deciding which image to keep, we generate corrected few-step previews and use them to select a candidate or discard the batch. If we select a candidate, we run the original full-step sampler from its original (z_i, c). Since the corrector is trained to minimize image reconstruction error, selection is not tied to a particular image metric. Users can apply a metric suited to their task or choose directly from the previews, keeping a human in the loop.

3 Instance-wise correspondence through few-step samplers

3.1 Hidden Reconstruction Capacity of Frozen Few-Step Samplers

Our approach is inspired by diffusion-based inverse problems (Wang et al., 2024; Jia et al., 2026). These methods optimize the initial noise of a frozen, few-step diffusion model (e.g., 3-4 step DDIM sampler) to reconstruct an image from corrupted observations. Even when a few-step sampler produces blurry images (see the first column of Figure 3), optimizing its input noise can yield accurate reconstruction. This leads us to ask whether the same idea can help generation: can we correct the initial Gaussian noise so that a few-step sampler produces an image close to its full-step counterpart?

As a proof of concept, we optimize the initial noise of frozen three-step DDIM samplers on CIFAR-10, LSUN-Church, CelebA-HQ, and LSUN-Bedroom. References use 1,000-step DDIM for CIFAR-10 and 20-step DDIM for the other three datasets. We evaluate 100 seeds per model and perform 200 Adam updates per image. Optimization starts from the reference image's original noise, with all model parameters frozen and no constraint on the optimized noise.

Across the four unconditional models, noise optimization raises mean PSNR from 14.62–17.00 dB to 37.41–39.98 dB. We additionally evaluate FLUX.1-dev on four fixed prompt–noise pairs, using a four-step ODE sampler, 28-step reference targets, and 2,560 optimization updates per image, with reference guidance 3.5. Mean PSNR increases from 14.54 to 37.49 dB (Table 1).

These target-aware experiments demonstrate that frozen few-step samplers can closely reproduce full-step outputs through input optimization. They motivate learning an input correction that approximates this capability without target access or per-image optimization at inference.

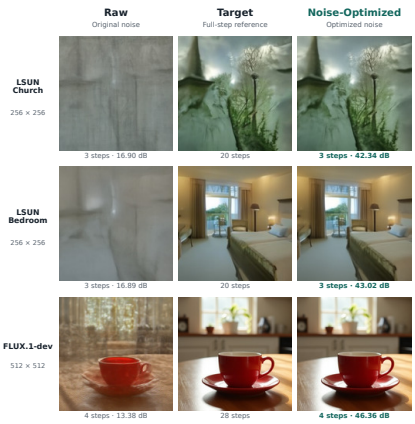


Figure 3: Noise optimization reveals the reconstruction capacity of frozen few-step samplers. Columns show raw outputs, full-step targets, and noise-optimized outputs.

3.2 Input Correction

We evaluate input correction on CIFAR-10 (Krizhevsky, 2009), LSUN-Church (Yu et al., 2015), and CelebA-HQ (Liu et al., 2015). For each dataset, the base sampler is 3-step DDIM (Song et al., 2021) with a frozen pretrained denoising network. The reference sampler is 1000-step DDIM for CIFAR-10 and 20-step DDIM for Church and CelebA-HQ. Given an initial noise sample z, the learned corrector predicts a residual Δz. We add this residual to z, rescale the result to preserve the original noise norm, and feed the corrected noise

Table 1: Frozen-model oracle noise optimization. PSNR (dB) is the mean ± sample SD over N images; few/ref denotes student/teacher steps.

Model / data	N	few/ref	Raw	Noise-Optimized	Gain (dB)
DDPM / CIFAR-10	100	3 / 1000	17.00 ± 1.91	37.41 ± 1.89	+20.41
DDPM / LSUN-Church	100	3 / 20	14.96 ± 1.49	38.01 ± 3.15	+23.05
DDPM / CelebA-HQ	100	3 / 20	14.62 ± 1.65	39.14 ± 3.89	+24.52
DDPM / LSUN-Bedroom	100	3 / 20	15.52 ± 1.77	39.98 ± 3.09	+24.46
FLUX.1-dev	4	4 / 28	14.54 ± 1.95	37.49 ± 7.69	+22.95

into the unchanged 3-step DDIM sampler, as described in Section 2. Training and implementation details are provided in Appendix B.2.

We evaluate all methods on the same 100 held-out noise samples per dataset and compare their outputs with the corresponding teacher outputs generated from the same initial noises. We report MSE and PSNR for pixel fidelity, LPIPS for perceptual similarity, and SSIM for structural similarity. Metrics are computed at each model's native resolution and averaged over individual samples; in particular, the reported PSNR is the mean per-image PSNR.

Our baselines include the uncorrected 3-step DDIM sampler, 4-step DDIM, 4-step DPM-Solver++ (Lu et al., 2025), and two 4-step LD3 variants (Tong et al., 2025). The first LD3 variant applies the same publicly released four-step LSUN time-step schedule directly to our backbone, without retraining or schedule remapping. Meanwhile, we also learn a dataset-specific schedule on the target backbone using its corresponding teacher. Both LD3 variants require four denoiser evaluations. Our method instead uses three denoiser evaluations and one lighter corrector evaluation, denoted by $3 + 1$ in Table 2.

As shown in Table 2, input correction improves all four teacher-reconstruction metrics over every evaluated baseline on all three datasets. Relative to uncorrected 3-step DDIM, our method reduces MSE by 89.7%, 91.3%, and 95.8% on CIFAR-10, Church, and CelebA-HQ, respectively. Compared with retrained 4-step LD3, the corresponding MSE reductions are 78.0%, 53.2%, and 71.6%, accompanied by PSNR gains of 3.41–6.75 dB and LPIPS reductions of 15.8–44.3%. Thus, under this evaluation protocol, correcting the initial noise yields closer agreement with the teacher than the tested four-step samplers while retaining the original three-step DDIM sampler. Representative examples in Figure 4 illustrate these differences across all three datasets. Our outputs more closely match the teacher's object shape and color on CIFAR-10, architectural features on LSUN-Church, and facial features and fine details around the hairline on CelebA-HQ, whereas the few-step baselines tend to blur or alter these details.

Transfer across steps We investigate whether the correction direction learned for a three-step sampler generalizes to larger sampling budgets. As more denoising steps become available, we expect a smaller correction of the initial noise to be sufficient. We therefore reuse the learned correction Δz with a step-dependent scale $\alpha(K, p) = (3/K)^p$, and normalize $z + \alpha(K, p)\Delta z$ to the norm of the original noise z before sampling. This preserves the full correction at $K = 3$ and progressively attenuates it as K increases.

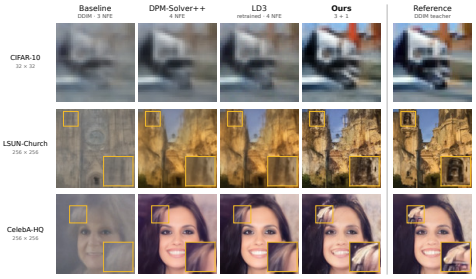


Figure 4: Representative examples of input correction on CIFAR-10, LSUN-Church, and CelebA-HQ. Columns show uncorrected 3-step DDIM, 4-step DPM-Solver++, 4-step LD3, and our method.

Using the frozen three-step correctors on LSUN-Church and CelebA-HQ, we evaluate $K \in \{3, 4, 6, 8\}$ and $p \in \{1.7, 2.0, 2.5\}$ without retraining. Each configuration uses the same 200 noises per dataset, with 20-step DDIM outputs as paired references. As shown in Table 5 in Appendix A.4 and Figure 5, every tested configuration improves PSNR over raw DDIM at the same denoising step count, while also reducing MSE and LPIPS and increas-

ing SSIM, HPS, and ImageReward Xu et al. (2023). These results demonstrate that the learned correction direction remains useful beyond its original sampling budget.

The exponent provides a simple control over the fidelity–quality trade-off. At each transferred budget, $p = 2.5$ achieves the highest PSNR, whereas $p = 1.7$ achieves the highest HPS. A larger exponent favors conservative correction for teacher fidelity, while a smaller exponent favors stronger correction for perceptual quality.

Additional ablations support the use of noise norm normalization, sample-specific correction directions, and reduced correction magnitudes when transferring to more sampling steps (Appendix C.6).

From input to trajectory correction. Input correction improves correspondence while leaving the sampling map unchanged. We next ask whether allowing corrections within the denoising updates can further improve fidelity. We compare both parameterizations on CIFAR-10 and SD1.5 under matched data, endpoint objectives, and training-update budgets. Trajectory correction improves PSNR over input correction from 26.60 to 29.79 dB on CIFAR-10 and from 20.60 to 23.25 dB on SD1.5. Both substantially outperform uncorrected three-step sampling. These results motivate exploring trajectory correction for text-to-image previews; the controlled comparison is detailed in Appendix C.3.

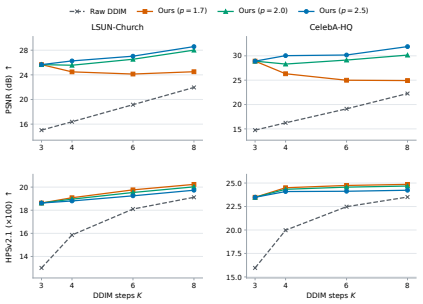


Figure 5: PSNR and HPS on LSUN-Church and CelebA-HQ as the number of DDIM steps increases. Dashed lines are raw DDIMs.

3.3 Trajectory Correction

We next evaluate trajectory correction on SD1.5 and SDXL (Rombach et al., 2022; Podell et al., 2024). We train a trajectory correction on three denoising steps for both models and evaluates on a disjoint set of 100 prompts from COCO val2017 (Lin et al., 2014), each paired with eight fixed noises. We evaluated at different CFG scales comparing with the full-denoising, 60 steps, teacher. We also optimized a 3-step timestamp using LD3 (Tong et al., 2025) for the Stable Diffusion family. For SD1.5, we additionally compare against the ConsistencySolver (Wang et al., 2026) pretrained solver.

Table 2 reports the SDXL and SD1.5 results at CFG scales 1, 3, and 5. Trajectory correction consistently outperforms all baselines across all four image-level similarity metrics and all evaluated guidance scales. Across SD1.5 and SDXL at CFG scales 1, 3, and 5, averaging per-group improvements against the best reported baseline for each metric, our method reduces LPIPS by 18.3%, increases SSIM by 18.7%, and improves PSNR by 2.32 dB. These gains persist as the CFG scale increases; however, stronger classifier-free guidance makes the denoising trajectory increasingly difficult to approximate under aggressive step compression, resulting in larger deviations from the full-denoising teacher. Accordingly, baseline similarity deteriorates at higher CFG scales, while trajectory correction substantially mitigates this degradation. Complete results including higher CFG comparisons are in Appendix C.

3.4 Transfer across backbones

We report an unexpected aesthetic benefit of cross-backbone input correction. Inspired by cross-domain inverse-problem results (Jia et al., 2026), we train a corrector through a frozen Bedroom backbone using CelebA teacher targets, then apply it unchanged to CelebA-HQ DDIM3. On 200 test samples, correction improves teacher-reconstruction PSNR from 14.78 to 22.82 dB. More surprisingly, the outputs exceed both the in-domain corrector and the full-step teacher in HPS, ImageReward, and PickScore Kirstain et al. (2023); for example, HPS increases from the teacher's 24.88 to 27.34. Figure 6 illustrates the resulting visual changes, with full results in Appendix C.4 and Figure 22. We hypothesize that backbone mismatch implicitly regularizes the correction toward perceptually preferred structure. Understanding the mechanism and generality of this effect is of independent interest for future work.

Table 2: Within each dataset/CFG group, metrics are reported in the order MSE ↓, PSNR ↑, LPIPS ↓, and SSIM ↑. Noise-correction results are averaged over 100 paired test noises per dataset. For input correction, $3+1$ denotes three denoiser evaluations and one lighter corrector evaluation. Trajectory correction is evaluated on SDXL and SD1.5. Bold denotes the best reported result within each group.

Input Correction MSE ($\times 10^{-3}$) ↓ / PSNR ↑ / LPIPS ↓ / SSIM ↑

Method	NFE	CIFAR-10				LSUN-Church				CelebA-HQ			
3-step DDIM	3	20.79	17.281	0.355	0.575	33.48	15.010	0.571	0.580	34.64	14.882	0.583	0.666
4-step DDIM	4	15.36	18.652	0.296	0.654	24.11	16.447	0.482	0.632	24.81	16.388	0.417	0.729
4-step DPM++	4	10.21	20.655	0.279	0.717	7.77	21.445	0.341	0.742	8.14	21.563	0.301	0.812
4-step LD3 (transferred)	4	24.58	17.494	0.274	0.732	14.16	18.708	0.500	0.712	14.50	18.731	0.409	0.736
4-step LD3 (retrained)	4	9.75	21.045	0.248	0.764	6.23	22.487	0.343	0.762	5.08	23.847	0.215	0.864
Ours	$3+1$	**2.15**	**27.790**	**0.138**	**0.919**	**2.91**	**25.899**	**0.277**	**0.856**	**1.44**	**29.054**	**0.181**	**0.917**

Trajectory Correction (SDXL) MSE ($\times 10^{-3}$) ↓ / PSNR ↑ / LPIPS ↓ / SSIM ↑

Method	NFE	CFG = 1				CFG = 3				CFG = 5			
3-step DDIM	3	16.394	18.097	0.580	0.566	26.468	15.965	0.643	0.548	34.866	14.733	0.676	0.518
3-step DPM++	3	10.358	20.284	0.609	0.518	18.285	17.800	0.572	0.628	33.657	15.162	0.589	0.616
3-step LD3	3	8.038	21.416	0.632	0.564	15.011	18.660	0.544	0.649	28.374	15.812	0.575	0.624
Ours	3	**4.139**	**24.382**	**0.431**	**0.727**	**9.010**	**21.015**	**0.451**	**0.741**	**13.687**	**19.154**	**0.477**	**0.727**

Trajectory Correction (SD 1.5) MSE ($\times 10^{-3}$) ↓ / PSNR ↑ / LPIPS ↓ / SSIM ↑

Method	NFE	CFG = 1				CFG = 3				CFG = 5			
3-step DDIM	3	25.003	16.338	0.581	0.382	31.575	15.350	0.626	0.386	40.469	14.234	0.661	0.384
3-step DPM++	3	17.456	18.034	0.572	0.430	20.485	17.292	0.546	0.509	30.023	15.558	0.576	0.516
3-step ConSolver	3	11.955	19.724	0.411	0.551	18.604	17.711	0.466	0.557	30.309	15.482	0.535	0.527
3-step LD3	3	16.742	18.249	0.781	0.428	19.521	17.569	0.635	0.510	29.887	15.640	0.615	0.511
Ours	3	**8.585**	**21.188**	**0.343**	**0.671**	**13.177**	**19.377**	**0.399**	**0.638**	**18.932**	**17.776**	**0.434**	**0.615**

4 Preservation of Candidate Rankings as Preview

Because the purpose of a "cheap preview" is to identify which candidates merit further computation, image-level reconstruction alone does not fully capture its utility. We therefore evaluate whether few-step previews preserve the relative quality ordering of candidate generations induced by their full-step teacher counterparts. For each prompt, eight noises define a common candidate pool. We independently score the few-step previews and the corresponding full-step teacher images, and measure how well the resulting candidate rankings agree. We consider automated scoring functions including PickScore, ImageReward v1.0, Aesthetic Score, CLIP, and HPS v2.1(Huang et al., 2026; Guo et al., 2026; Hessel et al., 2021; Wu et al., 2023). For the Stable Diffusion family, we use GPT-5.6-luna as an agent-based proxy for human preference in place of HPS v2.1, with details in Appendix B.4.

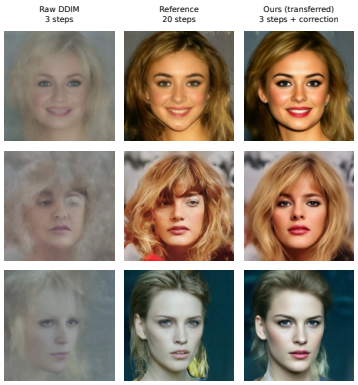


Figure 6: Bedroom-to-CelebA-HQ correction.

We study the Stable Diffusion family and FLUX.1-dev separately. For SD1.5 and SDXL, we compare against LD3 Tong et al. (2025); for SD1.5, we additionally compare against ConsistencySolver Wang et al. (2026). For FLUX.1-dev, we compare against probe-based methods (DiffusionProbe (Huang et al., 2026) and ProbeSelect (Guo et al., 2026)), ConsistencySolver, and distilled FLUX variants (TDD-FLUX and FLUX.1-schnell).

4.1 Stable Diffusion Family

For each scoring function, we report three statistics. Regret, defined as $s_{\text{oracle}} - s_{\text{selected}}$, which is the difference between the highest teacher score among the eight candidates and the teacher score of the selected candidate. Selected Score is the mean teacher score of the candidate selected by maximizing its preview score, directly measuring the quality achieved by the selection procedure.

Table 3: Candidate-ranking preservation on SDXL and SD1.5 across CFG scales. We compare DDIM, DPM++, LD3, and our method, in addition ConSolver for SD1.5. For PickScore, ImageReward, and agent, we report the selected teacher score ↑, regret ↓, and Spearman rank correlation ↑.

		SDXL Selected ↑ / Regret ↓ / Spearman ↑											
Metric	Method	CFG = 1			CFG = 3			CFG = 5			CFG = 7		
PickScore	3-step DDIM	19.642	1.377	0.016	21.724	0.915	0.009	22.162	0.855	-0.003	22.312	0.814	-0.011
	3-step DPM++	20.385	0.634	0.403	22.152	0.486	0.351	22.435	0.582	0.268	22.423	0.703	0.162
	3-step LD3	20.516	0.505	0.476	22.108	0.529	0.368	22.515	0.506	0.280	22.597	0.531	0.223
	Ours	**20.621**	**0.397**	**0.593**	**22.215**	**0.424**	**0.463**	**22.658**	**0.359**	**0.466**	**22.741**	**0.386**	**0.347**
ImageReward	3-step DDIM	-1.200	1.242	-0.030	0.274	0.701	-0.022	0.550	0.598	0.003	0.646	0.529	0.021
	3-step DPM++	-0.547	0.588	0.390	0.474	0.501	0.225	0.719	0.429	0.194	0.739	0.435	0.207
	3-step LD3	**-0.332**	**0.376**	0.522	0.616	0.358	0.356	0.745	0.402	0.265	0.849	0.326	0.254
	Ours	-0.363	0.404	**0.559**	**0.728**	**0.247**	**0.500**	**0.927**	**0.221**	**0.390**	**0.913**	**0.261**	**0.352**
Agent	3-step DDIM	57.670	28.340	0.113	73.890	16.240	0.004	76.680	14.300	0.035	76.730	14.520	0.025
	3-step DPM++	70.630	15.380	0.355	81.640	8.490	0.280	81.460	9.520	0.178	81.200	10.050	0.096
	3-step LD3	72.770	13.240	0.392	81.950	8.180	0.273	80.520	10.460	0.243	80.370	10.880	0.174
	Ours	**77.410**	**8.600**	**0.518**	**82.700**	**7.430**	**0.342**	**83.480**	**7.500**	**0.290**	**82.420**	**8.830**	**0.228**

		SD 1.5 Selected ↑ / Regret ↓ / Spearman ↑											
Metric	Method	CFG = 1			CFG = 3			CFG = 5			CFG = 7		
PickScore	3-step DDIM	19.573	0.848	0.128	20.830	0.882	0.007	21.421	0.614	0.075	21.412	0.764	-0.001
	3-step DPM++	19.857	0.564	0.369	21.111	0.600	0.324	21.404	0.632	0.247	21.583	0.593	0.119
	ConSolver	**20.025**	**0.396**	0.503	21.133	0.578	0.355	21.354	0.682	0.208	21.542	0.634	0.143
	3-step LD3	19.882	0.539	0.443	21.157	0.554	0.361	21.511	0.525	0.262	21.558	0.618	0.226
	Ours	19.975	0.446	**0.507**	**21.222**	**0.488**	**0.449**	**21.587**	**0.449**	**0.412**	**21.680**	**0.496**	**0.334**
ImageReward	3-step DDIM	-1.015	1.029	0.058	-0.078	0.811	0.060	-0.002	0.870	-0.009	0.128	0.799	0.019
	3-step DPM++	-0.573	0.587	0.408	0.173	0.560	0.330	0.228	0.641	0.192	0.365	0.562	0.171
	ConSolver	**-0.487**	**0.501**	**0.491**	0.295	0.439	0.339	0.263	0.605	0.211	0.367	0.561	0.136
	3-step LD3	-0.579	0.593	0.421	0.179	0.554	0.362	0.281	0.588	0.276	0.345	0.582	0.215
	Ours	-0.569	0.583	0.482	**0.381**	**0.352**	**0.444**	**0.391**	**0.478**	**0.355**	**0.524**	**0.403**	**0.278**
Agent	3-step DDIM	57.770	23.810	0.169	67.160	19.880	0.069	71.070	15.990	0.047	70.330	17.390	0.075
	3-step DPM++	66.030	15.550	0.387	73.130	13.910	0.319	73.660	13.400	0.279	72.510	15.210	0.120
	ConSolver	**69.810**	**11.770**	**0.473**	74.160	12.880	0.340	73.660	13.400	0.221	73.280	14.440	0.154
	3-step LD3	63.210	18.370	0.423	74.320	12.720	**0.373**	74.720	12.340	0.279	73.410	14.310	0.241
	Ours	68.680	12.900	0.450	**76.620**	**10.420**	0.357	**75.750**	**11.310**	**0.294**	**76.360**	**11.360**	**0.283**

Lastly, Spearman correlation measures the agreement between the candidate ranking induced by the few-step previews and that of the corresponding full-step teacher generations.

As reported in Table 3, our method consistently preserves teacher rankings across guidance scales and model families. Averaged over the three scoring metrics and four CFG scales in Table 3, Spearman correlation increases from 0.298 for ConSolver to 0.387 for our method on SD1.5, and from 0.319 for LD3 to 0.421 on SDXL. The agent-based evaluation shows a similar trend, with our previews exhibiting higher agreement with the teacher-sample rankings under a rubric that jointly considers prompt adherence, composition, visual quality, and artifacts. Visual examples are also provided in Figure 17, 18, and 19, which illustrate how the previews align with the corresponding teacher samples. Overall, our previews remain informative for candidate selection even under stronger guidance.

4.2 Image Selection on FLUX.1-dev

We extend our evaluation to the larger-scale FLUX.1-dev model to test whether cheap previews can identify promising candidates before full-step generation. We train a FLUX.1-dev student with trajectory LoRA adapters to generate four-step previews, using a 28-step reference sampler to provide training targets. Each preview is trained to match the reference image generated from the same prompt and initial noise. At inference, we score eight previews, select one candidate, and generate its final image with the original 28-step sampler.

We compare against two probe-based early quality predictors, a learned preview solver, and a distilled model. Diffusion Probe (Huang et al., 2026) applies frozen released weights to attention features extracted after six denoising steps, using a 20-channel input mapping selected in an earlier ImageReward comparison. Probe-Select (Guo et al., 2026) scores step-six transformer features with a FLUX-specific head trained on 35,332 image–score pairs. Both probes select one seed after six steps and continue its trajectory for the remaining 22 steps. We train a four-step FLUX.1-dev adaptation of ConSolver (Wang et al., 2026), optimizing its time-conditioned solver coefficients with PPO and a DINOv2 consistency reward against the 28-step reference, while keeping the generator

Table 4: FLUX.1-dev selection. HPS v2.1 and CLIP-L/14 cosine: $\times 100$. Bold: best non-oracle mean. NFE counts screening and final-generation denoiser calls, excluding decoding/scoring.

Method	ImageReward ↑	PickScore ↑	HPS v2.1 ↑	CLIP ↑	AES ↑	NFE ↓
Random	0.9483	22.9300	29.6458	25.7252	5.6658	28
Diffusion Probe	0.9731	22.9311	29.7302	25.7060	5.6749	70
Probe-Select	0.9442	22.9432	29.6611	25.7982	5.6709	70
ConSolver	1.0085	22.9818	29.8980	25.7976	5.6857	60
TDD-FLUX	1.0331	23.0183	29.9616	25.9744	5.7166	60
Ours	**1.1417**	**23.1869**	**30.3263**	**26.4693**	**5.7625**	60
Full best-of-8	1.3640	23.5260	31.4015	27.8651	5.9418	224

frozen. We also evaluate TDD-FLUX using the released TDD-ADV LoRA weights with four Euler steps without additional training.

Our method, ConSolver, and TDD-FLUX perform metric-specific selection: each scoring metric independently selects a candidate from the eight previews, and we evaluate the corresponding 28-step reference image. Probe training is itself metric-specific: predicting another metric requires a correspondingly supervised prediction head, whereas the same image previews can be rescored with different metrics without retraining.

We use 512 COCO val2017 prompts and eight fixed initial noise samples per prompt, with all methods selecting from the same candidate pool. Final candidates are generated with FLUX.1-dev at 512×512 resolution using 28 Euler steps and guidance embedding 3.5. We also report the expected score under uniform random selection and a full best-of-eight oracle that completes all eight images before selecting the best one for each metric. Further experimental details are given in Appendix B.5.

Table 4 shows that our preview-based selection achieves the highest non-oracle mean on all five metrics. Our method reaches 1.1417 ImageReward, compared with 1.0331 for TDD-FLUX and 1.0085 for ConSolver. The paired ImageReward gain over TDD-FLUX is 0.1086, with a 95% prompt-bootstrap confidence interval of $[0.0654, 0.1509]$. Thus, our corrected previews provide more effective candidate selection than the evaluated distilled model at the same sampling budget. All three four-step preview workflows require $8 \times 4 + 28 = 60$ NFE, compared with $8 \times 6 + 22 = 70$ for the probes and $8 \times 28 = 224$ for full best-of-eight.

Additional experiments show that our previews improve candidate selection over FLUX.1-schnell (another distilled model) and an equal-budget raw-sampling baseline, while generalizing to a higher resolution and PartiPrompts without retraining (Appendix C.5).

5 Discussion, Limitations, and Future Directions

Our results reveal underused reconstruction capacity in frozen few-step samplers, with learned input corrections remaining useful across sampling budgets and backbones. We apply these findings to fast previews for candidate selection, improving ranking preservation and selected-image scores over the evaluated baselines. By allowing users to screen previews and reserve full-step generation for selected candidates, this preview–select–render workflow can potentially reduce the time needed to obtain a satisfactory image.

Several limitations and open questions remain. Although our learned correctors substantially improve fidelity, they do not yet match the reconstruction accuracy achieved by target-aware oracle noise optimization, leaving considerable room to explore more effective correction methods. The generality of our empirical findings also warrants further evaluation on larger-scale diffusion models. Moreover, while transfer across sampling budgets and backbones shows promising results, broader experiments and mechanistic investigation are needed to understand when such transfer succeeds and which properties of the learned corrections enable it.

Future work could extend corrected few-step samplers to image editing and inverse problems, use lightweight corrections to help smaller models approximate larger teachers for resource-constrained deployment, and identify backbone–target mismatches that reliably improve fidelity or perceptual quality.

AI USE STATEMENT

AI tools assisted with literature review, manuscript editing, and code development. The authors implemented the core codebase, while AI tools helped coordinate parallel training runs and execute existing baselines. The authors take full responsibility for the content and results.

REPRODUCIBILITY STATEMENT

We provide implementations, evaluation scripts, and checkpoint download links in an anonymous repository at `https://anonymous.4open.science/r/faithful-previews-anonymous-F2EB`. The released input-correction and trajectory-correction checkpoints are hosted on Google Drive, with download links and loading instructions provided in the anonymous repository. The release supports oracle noise optimization on unconditional models, learned input correction and transfer across sampling budgets, trajectory correction with SD1.5, SDXL, and FLUX.1-dev, and preview-based candidate selection.

Appendix A details the correction objectives and parameterizations; Appendix B describes the experimental protocols and training settings; and Appendix C provides additional comparisons, ablations, and runtime measurements.

REFERENCES

Donghoon Ahn, Jiwon Kang, Sanghyun Lee, Jaewon Min, Minjae Kim, Wooseok Jang, Hyoungwon Cho, Sayak Paul, SeonHwa Kim, Eunju Cha, Kyong Hwan Jin, and Seungryong Kim. A noise is worth diffusion guidance. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=xEWooSOgaz`.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. 2025. URL `https://arxiv.org/abs/2506.15742`.

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 *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=FPnUhsQJ5B`.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. *Advances in Neural Information Processing Systems*, 37:125487–125519, 2024.

Huanlei Guo, Hongxin Wei, and Bingyi Jing. Toward early quality assessment of text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38410–38419, June 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf`.

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 *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Bukun Huang, Benlei Cui, Zhizeng Ye, Xuemei Dong, Tuo Chen, Hui Xue, Dingkang Yang, Longtao Huang, Haiwen Hong, and Jingqun Tang. Diffusion probe: Generated image result prediction using cnn probes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35926–35935, June 2026.

Jing Jia, Wei Yuan, Sifan Liu, Liyue Shen, and Guanyang Wang. Weak diffusion priors can still achieve strong inverse-problem performance. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=fdkSA4F0lN.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663. Curran Associates, Inc., 2023. doi: 10.52202/075280-1594. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/73aacd8b3b05b4b503d58310b523553c-Paper-Conference.pdf.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025. doi: 10.1007/s11633-025-1562-4. URL https://doi.org/10.1007/s11633-025-1562-4.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Vinh Tong, Trung-Dung Hoang, Anji Liu, Guy Van den Broeck, and Mathias Niepert. Learning to discretize denoising diffusion odes. In *International Conference on Learning Representations*, volume 2025, pp. 47244–47282, 2025.

Fu-Yun Wang, Hao Zhou, Liangzhe Yuan, Sanghyun Woo, Boqing Gong, Bohyung Han, Ming-Hsuan Yang, Han Zhang, Yukun Zhu, Ting Liu, and Long Zhao. Image diffusion preview with consistency solver. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43271–43280, 2026.

Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, and Ju Sun. DMPlug: A plug-in method for solving inverse problems with diffusion models. *Advances in Neural Information Processing Systems*, 37:117881–117916, 2024.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

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 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JVzeOYEx6d.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang, Ruikai Zhou, and Shilin Zhou. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. 2026. URL https://arxiv.org/abs/2511.22699.

A Correction Objectives and Method Details

A.1 Input Correction Details

The corrector predicts a residual $\Delta z = C_\phi(z)$. For the unconditional experiments, we normalize the corrected input to preserve the original noise norm:

$$\widetilde{z} = \|z\|_2 \frac{z + \alpha \Delta z}{\|z + \alpha \Delta z\|_2}, \qquad \widehat{x} = F_{\text{few}}(\widetilde{z}). \tag{2}$$

Normalization is performed independently for each sample, with a small denominator floor for numerical stability. We use $\alpha = 1$ during training. The reconstruction target is $F_{\text{ref}}(z)$, associated with the original noise. Gradients propagate through the complete few-step sampler, while its parameters remain frozen.

A.2 Trajectory Correction Details

Each denoising step uses an independent set of LoRA adapters. For a frozen weight matrix W, step i uses

$$W_i = W + \frac{a}{r} B_i A_i, \tag{3}$$

where r is the adapter rank and a is its scaling parameter. All adapters are trained jointly through the complete student rollout. Intermediate student states are not detached or replaced with reference states.

Training supervises both the final output and the terminal latent:

$$\mathcal{L} = \mathbb{E}\left[0.1 \log\left(\text{MSE}_{\text{RGB}} + 10^{-6}\right) + 0.05\, \text{MSE}_{\text{latent}}\right]. \tag{4}$$

The logarithm is applied per sample before batch averaging. RGB errors use continuous outputs of the frozen VAE decoder. This objective requires reference endpoint latents and images, but no intermediate reference states.

A.3 Classifier-Free Guidance Conditioning

For Stable Diffusion, we encode the guidance scale s using Fourier features and a learned projection, whose output modulates the correction. Training targets use the corresponding reference guidance scale. FLUX instead uses its native guidance embedding: the reference uses 3.5 and the trained student uses 1.0, without a separate unconditional branch.

A.4 Transfer Across Sampling Steps

For a corrector trained with k steps, we use $\alpha = (k/K)^p$ at inference with K steps. Scaling is applied before norm normalization. The corrector remains frozen. Our experiments use $k = 3$, $K \in \{3, 4, 6, 8\}$, and $p \in \{1.7, 2.0, 2.5\}$. This transfer procedure applies to input correction.

B Experimental Setup and Implementation Details

Unconditional diffusion backbones. For CIFAR-10, we use pretrained unconditional DDPM backbones (Ho et al., 2020) with EMA weights. For LSUN-Church, CelebA-HQ, and LSUN-Bedroom, we use `google/ddpm-ema-church-256`, `google/ddpm-ema-celebahq-256`, and `google/ddpm-ema-bedroom-256`, respectively, all publicly available on Hugging Face.

Conditional diffusion backbones. For text-to-image generation, we use Stable Diffusion 1.5, SDXL, and FLUX.1-dev. Their pretrained checkpoints are available on Hugging Face as `stable-diffusion-v1-5/stable-diffusion-v1-5`, `stabilityai/stable-diffusion-xl-base-1.0`, and `black-forest-labs/FLUX.1-dev`, respectively.

B.1 Oracle Noise Optimization

For unconditional models, we optimize the original Gaussian noise through a frozen three-step DDIM sampler using 200 Adam updates per image and 100 seeds per dataset. Reference targets use 1,000-step DDIM for CIFAR-10 and 20-step DDIM for LSUN-Church, CelebA-HQ, and LSUN-Bedroom.

The FLUX oracle uses four prompt–noise pairs, four-step students, and 2,560 updates.

B.2 Input Correction Training

For LSUN-Church, we use a residual U-Net corrector, whereas for CelebA-HQ we additionally include attention blocks. Both work well. All students use DDIM timesteps $[666, 333, 0]$. Reference budgets are 1,000 steps for CIFAR-10 and 20 steps for Church and CelebA-HQ.

B.3 Trajectory Correction Training

For SDXL and SD1.5, we use a three-step sampler with timesteps $[666, 333, 0]$ and independent rank-64 LoRA adapters at each step. The CFG scale is encoded using Fourier features, followed by a learned linear projection with 64 outputs to produce continuous rank-wise gates:

$$g(s) = 1 + \tanh(W\gamma(s) + b)\,.$$

These gates attenuate or amplify the contribution of each LoRA rank. Training runs for 8,192 optimization steps on 4,000 prompts, each evaluated at five CFG scales. Testing uses 100 held-out prompts with eight seeds per prompt at each CFG scale.

The FLUX student uses four independent rank-64 adapter sets and 36,864 training pairs. We use AdamW with constant learning rate 10^{-4}, weight decay 10^{-4}, global batch size 64, gradient clipping at norm 5, and EMA decay 0.99. Training completes 1,548 updates; the EMA checkpoint at update 1,280 is selected using reconstruction PSNR on 128 validation pairs.

B.4 Agent-Based Visual-Quality Evaluation

We use a multimodal language model as an agent-based proxy for human visual preference. The evaluator considers prompt adherence, composition, perceptual quality, and visible artifacts. This evaluation complements conventional automated metrics and is not intended to replace a human study.

Evaluation protocol. For each text prompt and generation method, we collect the eight candidate images produced from eight fixed initial noises. All eight images are presented simultaneously to the evaluator in a single multimodal request. Before each request, we randomly permute their presentation order to reduce positional bias.

After shuffling, each image is assigned an opaque identifier that allows us to map the returned evaluation back to its original noise index. The identifiers do not reveal the generation method, sampling budget, guidance scale, original position, or noise seed. The evaluator receives only the original text prompt, the eight candidate images, and the evaluation instruction given below.

We use `GPT-5.6-luna` as the evaluator. A fresh conversation is created for every eight-image group, preventing information from being carried across prompts, methods, or candidate groups. The evaluator first assigns an independent score between 0 and 100 to every image and then produces a complete ranking of the eight candidates. Rank 1 denotes the best candidate.

Preview candidates and their corresponding full-step reference candidates are evaluated as separate eight-image groups. The image order is independently randomized for each group. The results are subsequently matched through their underlying noise indices, rather than through their presentation positions. Thus, the evaluator is not shown explicit preview–reference pairs and is not asked to perform direct image matching.

Evaluator instruction. We use the following instruction for all candidate groups without method-specific modification:

> You are a strict, repeatable visual-quality evaluator for text-to-image research. Score each candidate independently against the supplied prompt, then rank the candidates within this prompt. Do not reward a candidate for being more colorful, brighter, or more photorealistic unless that helps satisfy the prompt. Do not compare candidates across different prompts.
>
> Scoring rubric (0-100): 90-100: excellent prompt adherence, composition, visual quality, and no meaningful artifacts; 75-89: strong image with only minor omissions or defects; 50-74: recognizable but with noticeable prompt, composition, or quality problems; 25-49: major omissions, poor composition, or substantial artifacts; 0-24: largely unrelated, unusable, or severely corrupted.
>
> Use the full range when justified. Grade each image independently first; then rank by grade and visual judgment. Rank 1 is best. Break close ties by prompt adherence, then by visible image quality. Return only the requested JSON.

The required JSON response contains the opaque identifier, numerical score, and rank of every candidate. We verify that all eight submitted identifiers appear exactly once, that every score lies between 0 and 100, and that the returned ranks form a permutation of $\{1, \ldots, 8\}$.

B.5 FLUX.1-dev Image Selection

We use 512 COCO prompts and eight noises per prompt. Final images use 28 Euler steps at 512×512 resolution.

For Diffusion Probe, we use the released `base.pth` checkpoint without retraining, with a fixed mapping from FLUX attention features to its required 20-channel input. Using the released checkpoint with our feature mapping, Diffusion Probe achieves an ImageReward of 0.9731 on our 512×512 FLUX.1-dev benchmark, compared with 0.9483 for random selection. Exploratory retraining under our protocol did not yield a statistically significant improvement over this baseline: the highest score across four runs was 0.9785, a paired gain of 0.0054 (95% confidence interval $[-0.0392, 0.0506]$). The original study (Huang et al., 2026) also reports a modest ImageReward gain of 0.04, from 1.02 to 1.06, for FLUX.1-dev seed selection under its 1024×1024 setting with 25 denoising steps and ten candidates per prompt.

For Probe-Select, the public implementation we used was configured for SD3.5, and we did not find a directly usable FLUX checkpoint. We therefore train its released prediction-head architecture from scratch on FLUX features using 35,332 image–score pairs with ImageReward supervision, while keeping the FLUX backbone frozen. Thus, our comparison uses a frozen public checkpoint for Diffusion Probe and a locally trained FLUX adaptation of Probe-Select.

For ConSolver (Wang et al., 2026), we adapt the released flow-matching implementation, originally targeting FLUX Kontext, to four-step FLUX.1-dev text-to-image generation. We train its time-conditioned coefficient MLP, keeping the FLUX backbone, VAE, and DINOv2 reward model frozen. We train for 1,000 PPO updates, each using 16 stochastic rollouts of one prompt–noise pair. The reward is $50(1 + \cos(f_{\text{preview}}, f_{\text{reference}}))$, where f denotes the DINOv2-base CLS embedding and the reference is the 28-step Euler image generated from the same prompt and noise. We use AdamW with learning rate 10^{-4} and weight decay 10^{-3}, four PPO epochs per update, clipping threshold 0.2, entropy coefficient 0.01, and policy softmax temperature 0.01.

C Additional Experiments

C.1 Additional SD Family Results

Here we report the trajectory correction results for SD 1.5 and SDXL under high CFG values (7 and 9). Across both models, our method consistently achieves the best performance at both CFG settings, with substantially lower MSE and LPIPS and higher SSIM. The improvements are particularly pronounced at CFG = 9, demonstrating that trajectory correction remains effective and robust even under the increased difficulty induced by stronger classifier-free guidance.

Trajectory Correction (SDXL)		MSE ($\times 10^{-3}$) ↓ / PSNR ↑ / LPIPS ↓ / SSIM ↑							
Method	NFE	CFG = 7				CFG = 9			
3-step DDIM	3	42.634	13.846	0.775	0.485	50.090	13.141	0.790	0.456
3-step DPM++	3	51.495	13.301	0.630	0.574	69.432	11.980	0.671	0.530
3-step LD3	3	44.219	13.856	0.628	0.583	59.412	12.543	0.675	0.549
Ours	3	**18.263**	**17.873**	**0.505**	**0.709**	**22.513**	**16.922**	**0.531**	**0.692**

Trajectory Correction (SD 1.5)		MSE ($\times 10^{-3}$) ↓ / PSNR ↑ / LPIPS ↓ / SSIM ↑							
Method	NFE	CFG = 7				CFG = 9			
3-step DDIM	3	49.510	13.324	0.689	0.377	58.440	12.570	0.710	0.366
3-step DPM++	3	42.248	14.037	0.610	0.504	55.536	12.803	0.644	0.487
3-step ConSolver	3	43.357	13.879	0.592	0.496	56.324	12.700	0.637	0.470
3-step LD3	3	44.489	13.852	0.632	0.487	59.591	12.514	0.666	0.460
Ours	3	**24.290**	**16.636**	**0.460**	**0.596**	**29.337**	**15.758**	**0.483**	**0.579**

C.2 Extended Transfer Across Sampling Steps

We evaluate whether input correctors trained with three-step DDIM remain effective at larger sampling budgets without retraining. For $K \in \{3, 4, 6, 8\}$, we scale the predicted displacement by $\alpha = (3/K)^p$, with $p \in \{1.7, 2.0, 2.5\}$, and normalize the corrected input to the original noise norm. All configurations use the same 200 noise samples per dataset, with paired 20-step DDIM outputs as references. As shown in Table 5, every tested correction improves all six metrics over raw DDIM at the same step count. At transferred budgets, $p = 2.5$ yields the lowest MSE and LPIPS and the highest PSNR, whereas $p = 1.7$ consistently achieves the highest HPS. These results show that the learned correction remains useful beyond its training budget, with its strength controlling a trade-off between reference fidelity and preference scores.

Table 5: Transfer across steps on 200 samples per dataset. HPS: HPSv2.1 $\times 100$; IR: ImageReward. Best values per dataset, K, and metric are bold.

Metric order: MSE ↓	PSNR ↑	LPIPS ↓	SSIM ↑	HPS ↑	IR ↑						
K	Method	LSUN-Church						CelebA-HQ					
3	Raw DDIM	0.0336	15.024	0.571	0.577	13.010	-1.985	0.0355	14.738	0.592	0.658	15.958	-1.250
	Ours (all p)	**0.0031**	**25.661**	**0.287**	**0.850**	**18.624**	**-0.976**	**0.0015**	**28.909**	**0.183**	**0.915**	**23.482**	**-0.096**
4	Raw DDIM	0.0245	16.386	0.489	0.626	15.859	-1.330	0.0254	16.242	0.425	0.720	19.976	-0.488
	Ours ($p = 1.7$)	0.0039	24.486	0.244	0.864	**19.081**	-0.770	0.0026	26.318	0.160	0.904	**24.499**	**-0.135**
	Ours ($p = 2.0$)	0.0031	25.562	0.235	**0.872**	18.957	-0.782	0.0017	28.309	0.146	0.922	24.333	-0.142
	Ours ($p = 2.5$)	**0.0027**	**26.262**	**0.233**	0.869	18.803	**-0.767**	**0.0012**	**30.035**	**0.137**	**0.930**	24.094	-0.142
6	Raw DDIM	0.0131	19.157	0.348	0.738	18.103	-0.776	0.0133	19.114	0.282	0.806	22.481	-0.309
	Ours ($p = 1.7$)	0.0044	24.132	0.210	0.874	**19.762**	**-0.502**	0.0035	24.988	0.156	0.897	**24.743**	**-0.099**
	Ours ($p = 2.0$)	0.0026	26.532	0.183	**0.891**	19.535	-0.541	0.0014	29.130	0.120	0.937	24.564	-0.123
	Ours ($p = 2.5$)	**0.0023**	**27.030**	**0.181**	0.883	19.246	-0.605	**0.0011**	**30.168**	**0.106**	**0.943**	24.124	-0.131
8	Raw DDIM	0.0070	21.954	0.259	0.819	19.119	-0.627	0.0065	22.255	0.191	0.874	23.508	-0.231
	Ours ($p = 1.7$)	0.0041	24.506	0.186	0.886	**20.235**	**-0.416**	0.0036	24.920	0.144	0.902	**24.856**	**-0.131**
	Ours ($p = 2.0$)	0.0020	28.011	0.142	**0.915**	20.021	-0.462	0.0012	30.150	0.098	0.950	24.679	-0.140
	Ours ($p = 2.5$)	**0.0016**	**28.585**	**0.137**	0.912	19.731	-0.493	**0.0007**	**31.893**	**0.077**	**0.961**	24.232	-0.155

C.3 Controlled Comparison of Input and Trajectory Correction

Objective and comparison. We investigate whether correcting the denoising updates can improve instance-wise fidelity beyond correcting only the initial noise. We compare raw three-step DDIM, input correction, and trajectory correction on CIFAR-10 and SD1.5. Within each model, the learned methods use the same pretrained backbone, paired data, reference sampler, three-step grid, endpoint objective, minibatch order for each training seed, and optimization-update budget.

Table 6: Controlled comparison under matched data, endpoint supervision, and training-update budgets. MSE is multiplied by 10^3; PSNR is in dB. Params counts trainable parameters, and Calls counts U-Net evaluations. The input arm uses one additional full U-Net corrector. References use 1,000 steps for CIFAR-10 and 100 steps for SD1.5. Results use 1,000 CIFAR-10 test inputs or 64 SD1.5 prompts with four noises each.

Method	Calls	Params (M)	MSE ↓	PSNR ↑	LPIPS ↓	SSIM ↑
CIFAR-10						
Raw DDIM3	3	0	21.18	17.29	0.149	0.578
Input correction	$3+1$	1.266	2.86	26.60	0.025	0.901
Trajectory correction	3	3.788	**1.66**	**29.79**	**0.011**	**0.944**
SD1.5, CFG 1						
Raw DDIM3	3	0	20.05	17.33	0.801	0.402
Input correction	$3+1$	1.606	9.70	20.60	0.408	0.570
Trajectory correction	3	4.783	**5.57**	**23.25**	**0.189**	**0.770**

Correction parameterizations. Input correction uses a second copy of the pretrained backbone with frozen base weights, rank-8 adapters, and a trainable zero-initialized output head. It predicts a residual adjustment to the initial noise; the corrected noise is normalized to the original per-sample norm before entering the unchanged three-step sampler. Trajectory correction uses a separate rank-8 adapter set at each of the three denoising steps, keeping the original initial noise and pretrained base weights fixed. Adapters target attention and convolutional projections on CIFAR-10 and attention projections on SD1.5. Both corrections are initialized to recover raw sampling and trained through the complete student rollout without intermediate-state supervision.

Data and sampling. CIFAR-10 uses a 1,000-step DDIM reference and student timesteps $[666, 333, 0]$. We use 4,096 training pairs, 256 validation pairs, and 1,000 held-out test noises, with 4,096 updates at batch size 32. SD1.5 uses a 100-step DDIM reference at CFG 1 and the native trailing three-step grid $[999, 666, 332]$. Training uses 2,048 pairs and 2,048 updates at batch size four; validation uses 128 pairs. Testing uses 64 held-out prompts with four new noises each.

Training and checkpoint selection. Both methods minimize RGB MSE on CIFAR-10. On SD1.5, both minimize

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B}\left[0.1\log(m_{x,i}+10^{-6}) + 0.05 m_{h,i}\right], \tag{5}$$

where $m_{x,i}$ is continuous RGB MSE on the nominal $[-1, 1]$ scale and $m_{h,i}$ is terminal-latent MSE. The pretrained VAE weights remain frozen while gradients through its input are retained. This auxiliary term requires the reference terminal latent.

All runs use AdamW with weight decay 10^{-4}, gradient clipping at norm 5, and EMA decay 0.99. A 64-update validation screen between learning rates 10^{-4} and 3×10^{-4} selects 3×10^{-4} for both methods on both models. We train two seeds per learned configuration. Validation PSNR selects among current and EMA weights evaluated every 256 updates; test data is not used for checkpoint or hyperparameter selection.

Results. Table 6 reports mean per-image metrics, averaged over the two training seeds for learned methods. MSE and PSNR use continuous RGB without clipping or quantization; reported MSE is normalized to the $[0, 1]$ scale. LPIPS uses AlexNet at native resolution with RGB clipped to $[-1, 1]$. SSIM uses clipped $[0, 1]$ RGB with Gaussian weighting.

Both correction methods improve all four fidelity metrics over raw sampling. Trajectory correction further improves PSNR over input correction by 3.19 dB on CIFAR-10 and 2.65 dB on SD1.5, with

paired 95% bootstrap intervals of $[3.03, 3.35]$ and $[2.47, 2.84]$. It also reduces LPIPS from 0.025 to 0.011 and from 0.408 to 0.189, respectively. Bootstrap intervals use 5,000 paired resamples of CIFAR inputs or SD prompts after averaging the two trained models; they quantify test-input uncertainty conditional on those models. The results demonstrate effective correction at both locations, with trajectory correction achieving higher fidelity under this training protocol.

Computational cost and interpretation. The parameterizations differ in trainable parameter count and implementation cost. Input correction adds one full U-Net call; trajectory correction operates within the existing three calls.

C.4 Cross-Backbone Transfer

We transfer a frozen noise corrector trained with a Bedroom diffusion backbone to CelebA-HQ DDIM3, without retraining or test-time scale selection. All methods are evaluated on the same 200 noise seeds at 256×256 resolution, which are disjoint from Table 5. The corrector was trained using cached CelebA 20-step DDIM teacher targets, so this experiment evaluates cross-backbone transfer with target-domain supervision.

Table 7: Cross-backbone transfer from Bedroom to CelebA-HQ. Results are mean ± SD over 200 samples. NFEs denote denoiser plus corrector forward passes. HPS denotes HPSv2.1 $\times 100$; IR denotes ImageReward. Preference scores use the fixed text "a portrait photo of a person" for all unconditional outputs.

(a) Preference scores

Method	NFEs	PickScore ↑	HPS ↑	IR ↑
Raw DDIM3	3	19.03 ± 0.45	16.53 ± 2.22	-0.98 ± 0.69
Bedroom-backbone corrector + DDIM3	$3+1$	$\mathbf{20.37} \pm 0.54$	$\mathbf{27.34} \pm 1.77$	$\mathbf{0.54} \pm 0.36$
CelebA corrector + DDIM3	$3+1$	19.90 ± 0.62	24.04 ± 2.13	0.10 ± 0.50
DDIM20 teacher	20	19.89 ± 0.61	24.88 ± 2.13	0.10 ± 0.51

(b) Agreement with the DDIM20 teacher

Method	MSE ($\times 10^{-3}$) ↓	PSNR ↑	LPIPS ↓	SSIM ↑
Raw DDIM3	36.26 ± 14.81	14.78 ± 1.86	0.58 ± 0.06	0.65 ± 0.08
Bedroom-backbone corrector + DDIM3	5.43 ± 1.53	22.82 ± 1.20	0.27 ± 0.06	0.81 ± 0.04
CelebA corrector + DDIM3	$\mathbf{1.32} \pm 0.70$	$\mathbf{29.35} \pm 2.18$	$\mathbf{0.17} \pm 0.05$	$\mathbf{0.92} \pm 0.03$

As shown in Table 7, the transferred corrector improves PickScore, HPS, and ImageReward over raw DDIM3 by approximately 1.34, 10.81, and 1.52, respectively, using three denoiser evaluations and one corrector evaluation. In particular, it achieves higher mean preference scores than the evaluated CelebA corrector and DDIM20 teacher. Agreement with the teacher improves over raw DDIM3 by 8.04 dB in PSNR, with an approximately 53.4% reduction in LPIPS. The CelebA corrector trained in Section 3.2 achieves closer teacher agreement across all four fidelity metrics. Therefore, with the same target, the in-domain corrector achieves much higher teacher fidelity, while transferring a corrector trained with an out-of-domain backbone achieves much higher aesthetic preference scores.

C.5 Additional FLUX Evaluations

Comparison with FLUX.1-schnell. We compare our method with a distilled few-step generator to investigate whether distillation provides an effective alternative for faithful previews. We evaluate four-step FLUX.1-schnell on 256 prompts with eight seeds each, using the same prompts and initial noises as the FLUX.1-dev reference. Although Schnell achieves higher ImageReward on the previews themselves, our method produces substantially closer matches to the 28-step reference, achieving 18.954 dB PSNR versus 11.357 dB. This advantage also improves candidate selection: selecting seeds by preview ImageReward yields final reference-image scores of 1.1361 with our method versus 0.9514 with Schnell (Figure 7). These results show that our method outperforms the evaluated distilled baseline in both preview fidelity and downstream candidate selection.

Budget-allocation ablation. We conduct a small budget-allocation ablation to investigate how a fixed computation budget can be divided between candidate count and preview steps. On the same

256 prompts, we compare eight corrected four-step previews with four raw eight-step previews. Including the final 28-step run, both workflows require 60 NFE. Our method achieves a selected-image ImageReward of 1.1361 versus 1.0573, a paired gain of 0.0788 (95% CI $[0.0292, 0.1308]$). These results show that screening more candidates with our corrected previews improves selection over spending more denoising steps on fewer uncorrected previews at the same budget.

Transfer across resolutions. We study whether the learned correction generalizes to higher-resolution generation without retraining. We apply the frozen corrector trained at 512×512 directly at 768×768, retaining its four-step time grid. On 64 prompts with eight seeds each, our method achieves 18.45 dB PSNR, compared with 13.96 dB for a raw control with the same guidance and time grid and 15.35 dB for native eight-step Euler. Selected-image ImageReward is 1.247, compared with 1.125 and 1.154, respectively. These results show that the correction remains effective at a higher resolution, improving both reference fidelity and ImageReward-based candidate selection without additional training.

Transfer to PartiPrompts. We investigate whether the learned correction generalizes to a different prompt collection without retraining. We evaluate the same frozen checkpoint on 128 PartiPrompts with eight seeds each at 512×512. Our previews achieve 19.082 dB PSNR and 0.380 LPIPS, compared with 14.052 dB and 0.542 for native four-step Euler. ImageReward-based selection improves the final-image score from 1.3232 to 1.4102 and Spearman rank correlation from 0.079 to 0.357. These results show that the correction remains effective beyond the training prompt source, improving both reference fidelity and candidate selection on PartiPrompts.

C.6 Ablation Studies

We evaluate frozen input correctors on 200 fixed noise samples per dataset, using 20-step DDIM outputs as references.

Noise norm normalization. We investigate whether preserving the original noise norm contributes to reconstruction fidelity. Removing normalization reduces three-step PSNR from 26.58 to 23.59 dB on LSUN-Church and from 29.35 to 26.64 dB on CelebA-HQ. Visually, removing norm normalization produces blurrier outputs with less distinct fine details (Figure 8). These results support normalizing the corrected noise to retain the original input norm.

Sample-specific correction direction. We investigate whether the improvement depends on the correction direction predicted for each input. We reverse the residual or replace its direction with that of another sample, preserving the original residual magnitude and applying noise norm normalization. Both controls yield lower PSNR than raw three-step DDIM (Figure 9). This indicates that the gains depend on the sample-specific correction direction.

Attenuation across sampling steps. We test whether a input correction learned for three-step sampling should be reduced in magnitude when reused for eight-step sampling. We apply the frozen three-step corrector to eight-step DDIM, comparing full-strength correction ($\alpha = 1$) with attenuation ($\alpha = (3/8)^{2.5}$), while retaining norm normalization. We evaluate both settings on the same set of 200 noise samples per dataset that are disjoint from previous experiments. Attenuation improves PSNR from 11.50 to 29.09 dB on LSUN-Church and from 11.78 to 31.94 dB on CelebA-HQ (Figure 10). These results show that retaining the full correction can substantially degrade fidelity, supporting a smaller correction at larger sampling budgets.

C.7 Runtime and Computational Cost

We measure inference time on a single NVIDIA GH200 GPU, averaging 32 measurements per setting after warm-up with CUDA synchronization. Three-step DDIM, our input-corrected sampler, and four-step DDIM take 87.6, 102.2, and 116.8 ms on LSUN-Church, and 90.5, 109.4, and 121.4 ms on CelebA-HQ, respectively. Thus, input correction costs less than an additional denoising step on these two datasets.

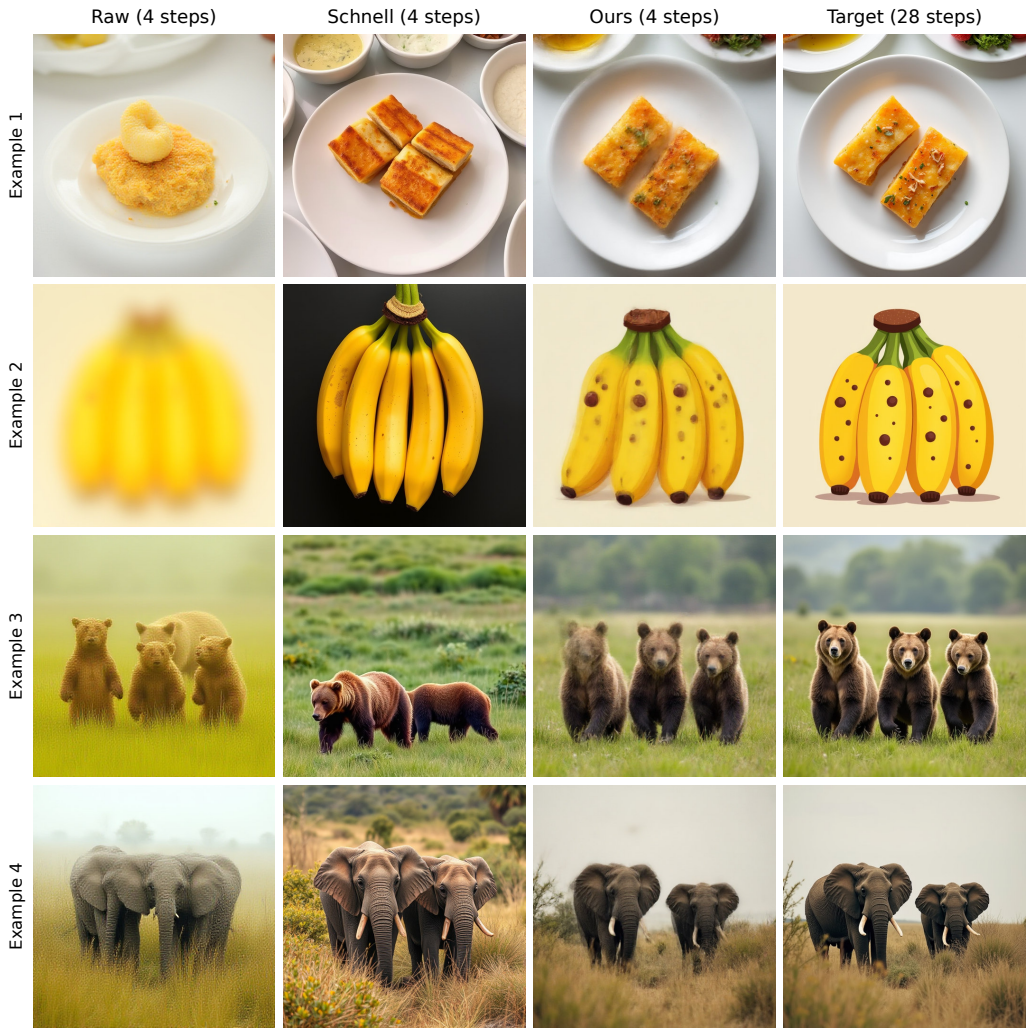


Figure 7: Comparison of FLUX.1-schnell and our method as previews of FLUX.1-dev. Columns show raw four-step Euler, Schnell, our corrected four-step sampler, and the 28-step reference. Although Schnell achieves higher ImageReward on the previews themselves, our method provides greater fidelity to the reference and better candidate-selection performance.

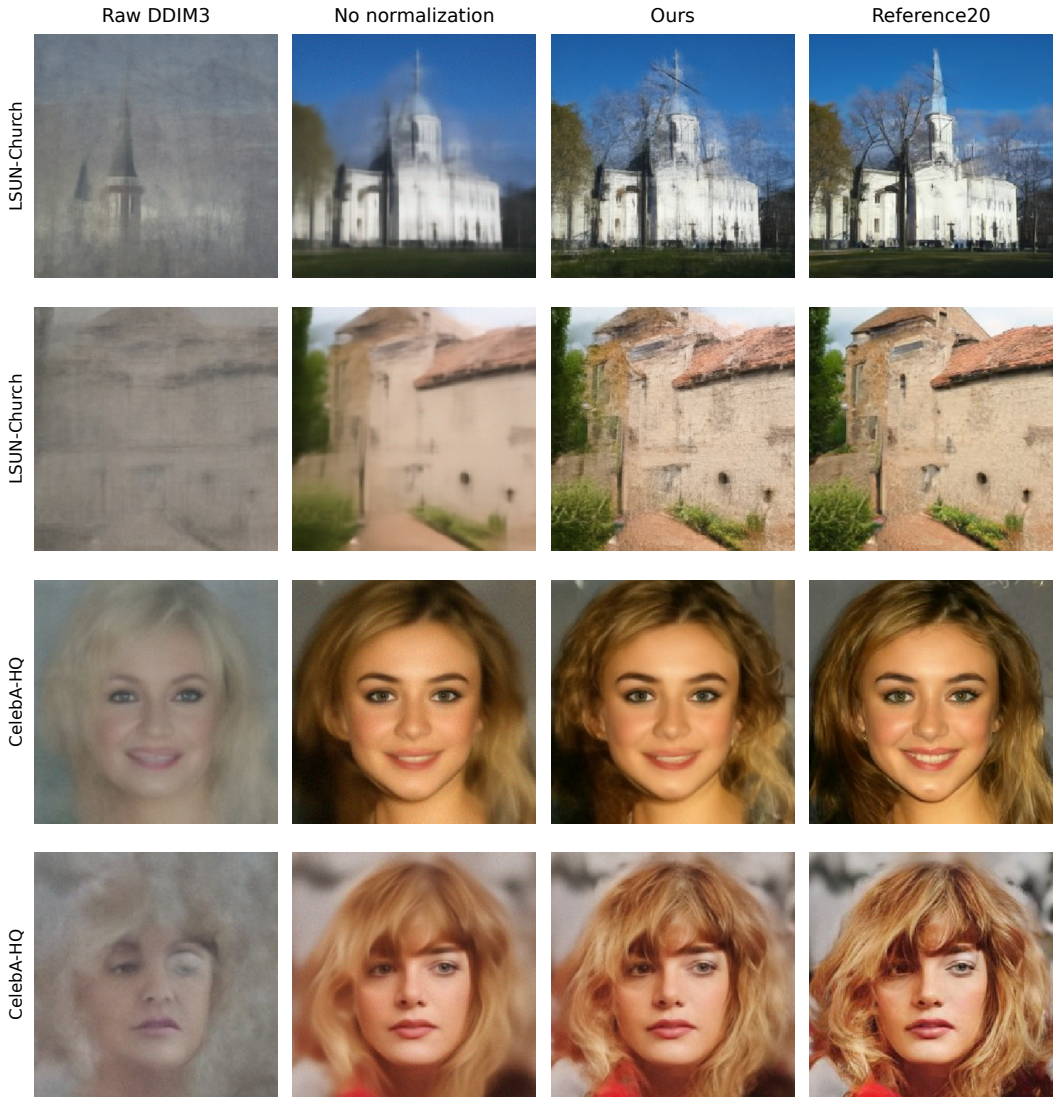


Figure 8: Noise norm normalization ablation. Columns show raw three-step DDIM, correction without normalization, complete correction, and the 20-step reference.

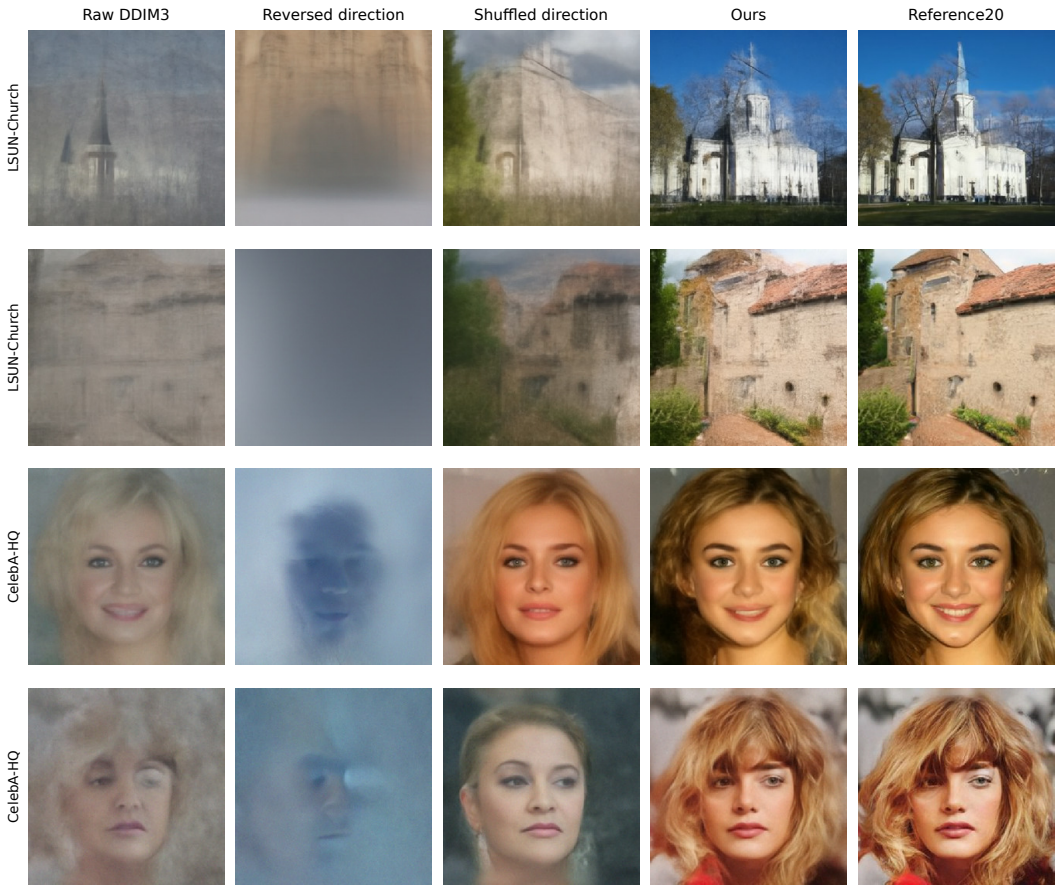


Figure 9: Correction direction ablation. Columns show raw three-step DDIM, reversed correction, another sample's correction direction rescaled to the original residual norm, complete correction, and the 20-step reference.



Figure 10: Attenuation when transferring a three-step corrector to eight-step DDIM. Columns show raw eight-step DDIM, correction with $\alpha = 1$, correction with $\alpha = (3/8)^{2.5}$, and the 20-step reference.

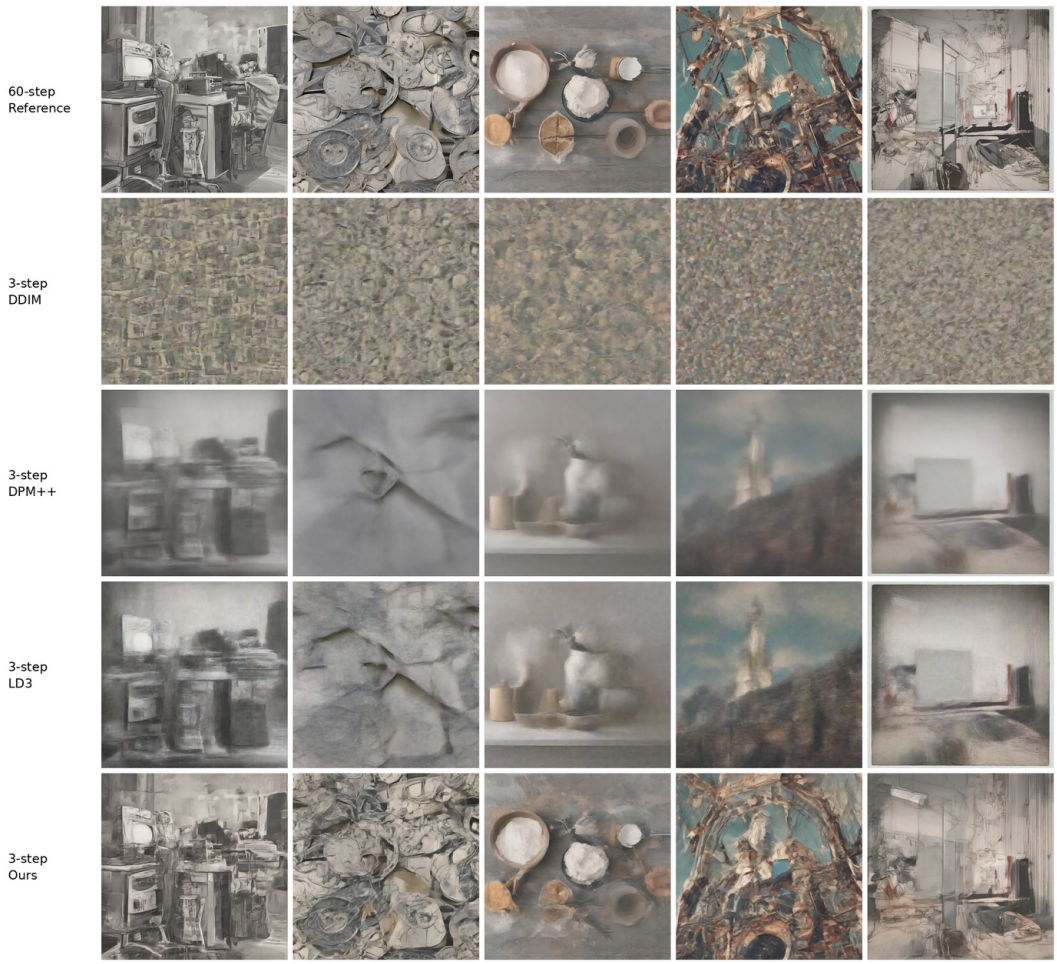


Figure 11: Visual comparison of 60-step Reference images and 3-step DDIM, DPM++, LD3, and Ours generations using SDXL at CFG 1.

For FLUX.1-dev at 512×512 and batch size eight, four-step denoising takes 1.739 s without trajectory correction and 1.740 s with correction, indicating negligible inference overhead at the same NFE.

Each denoising step uses 104.60 M LoRA parameters. The four independent adapter sets contain 418.38 M parameters in total, equivalent to 3.52% of the frozen 11.90 B-parameter FLUX transformer.

D Additional Visualizations

D.1 Oracle Noise Optimization

See Figure 20.

D.2 Transfer Across Sampling Steps

See Figure 21.

D.3 Cross-Backbone Transfer

See Figure 22.

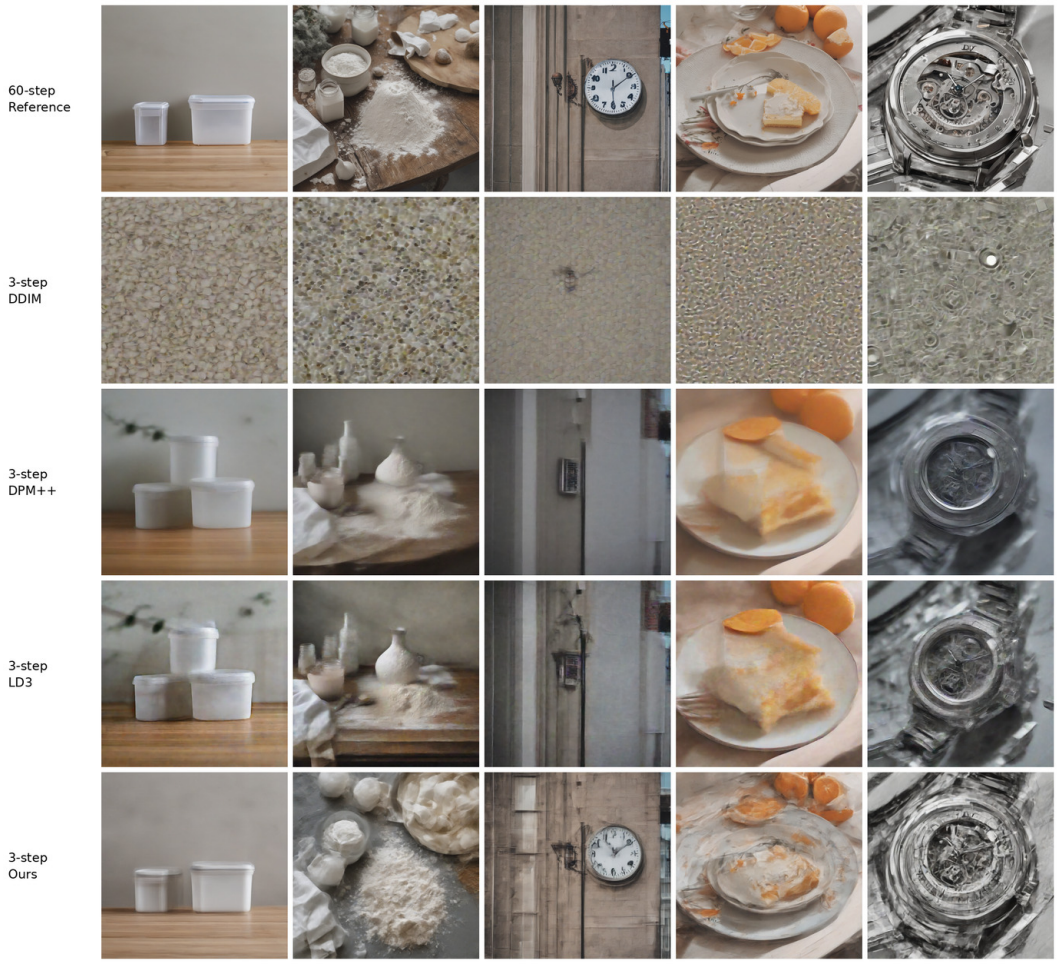


Figure 12: Visual comparison of 60-step Reference images and 3-step DDIM, DPM++, LD3, and Ours generations using SDXL at CFG 3.

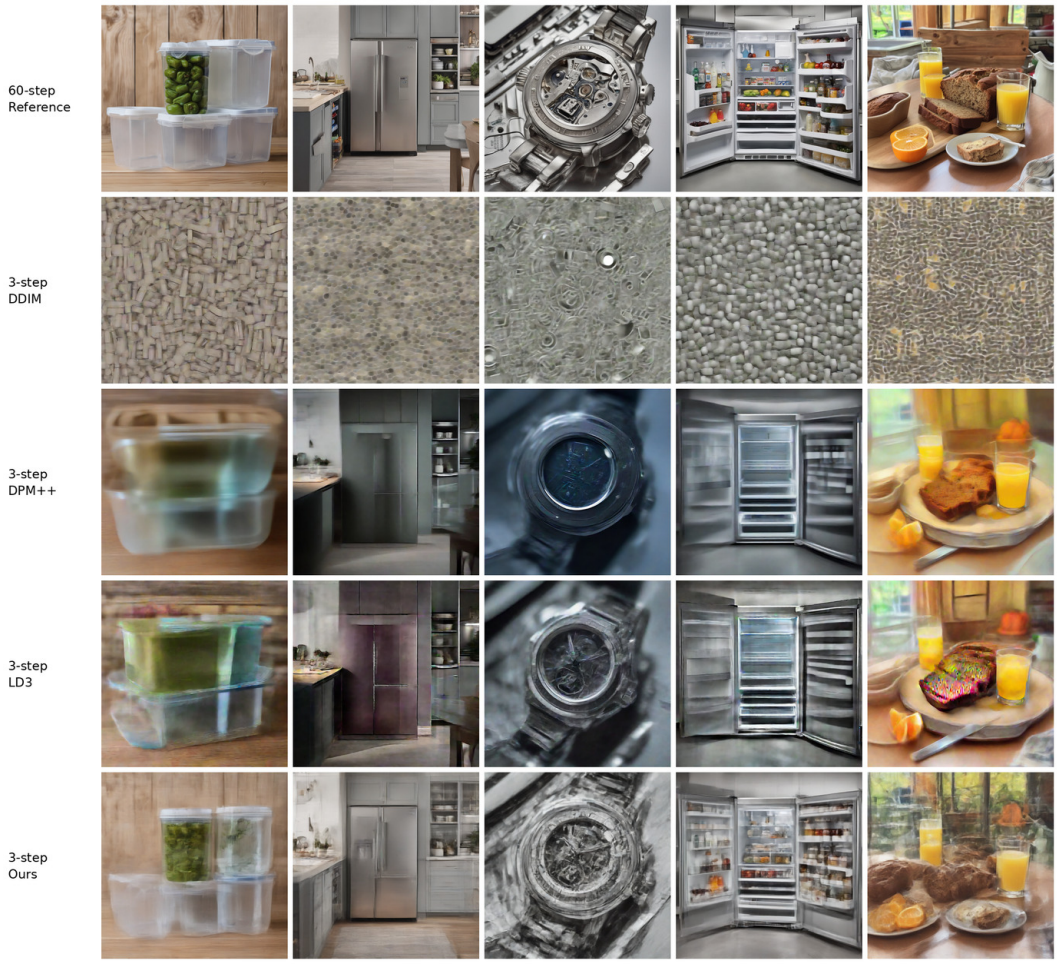


Figure 13: Visual comparison of 60-step Reference images and 3-step DDIM, DPM++, LD3, and Ours generations using SDXL at CFG 5.



Figure 14: Visual comparison of 60-step Reference images and 3-step DDIM, DPM++, LD3, ConSolver, and Ours generations using Stable Diffusion 1.5 at CFG 1.

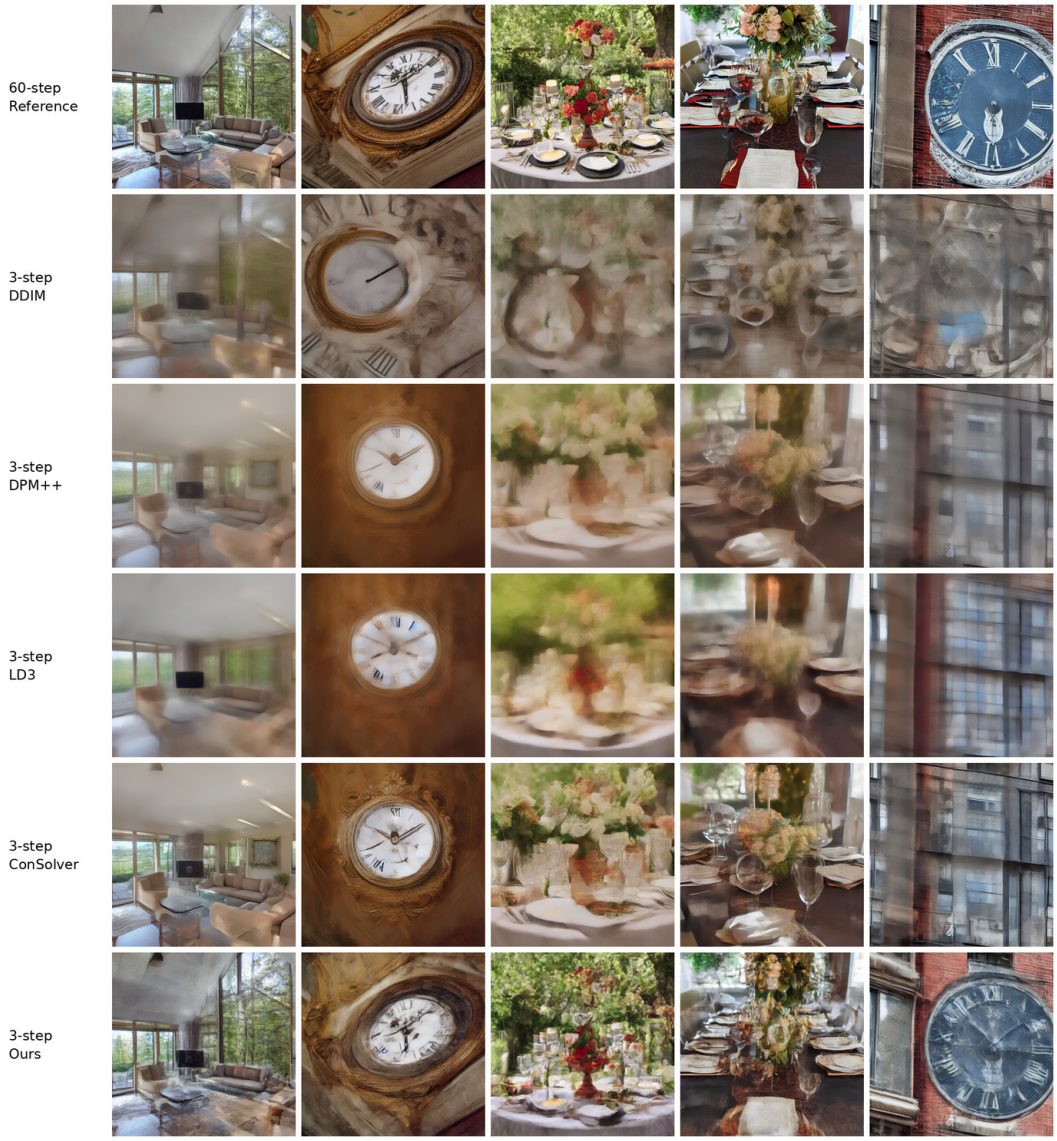


Figure 15: Visual comparison of 60-step Reference images and 3-step DDIM, DPM++, LD3, ConSolver, and Ours generations using Stable Diffusion 1.5 at CFG 3.

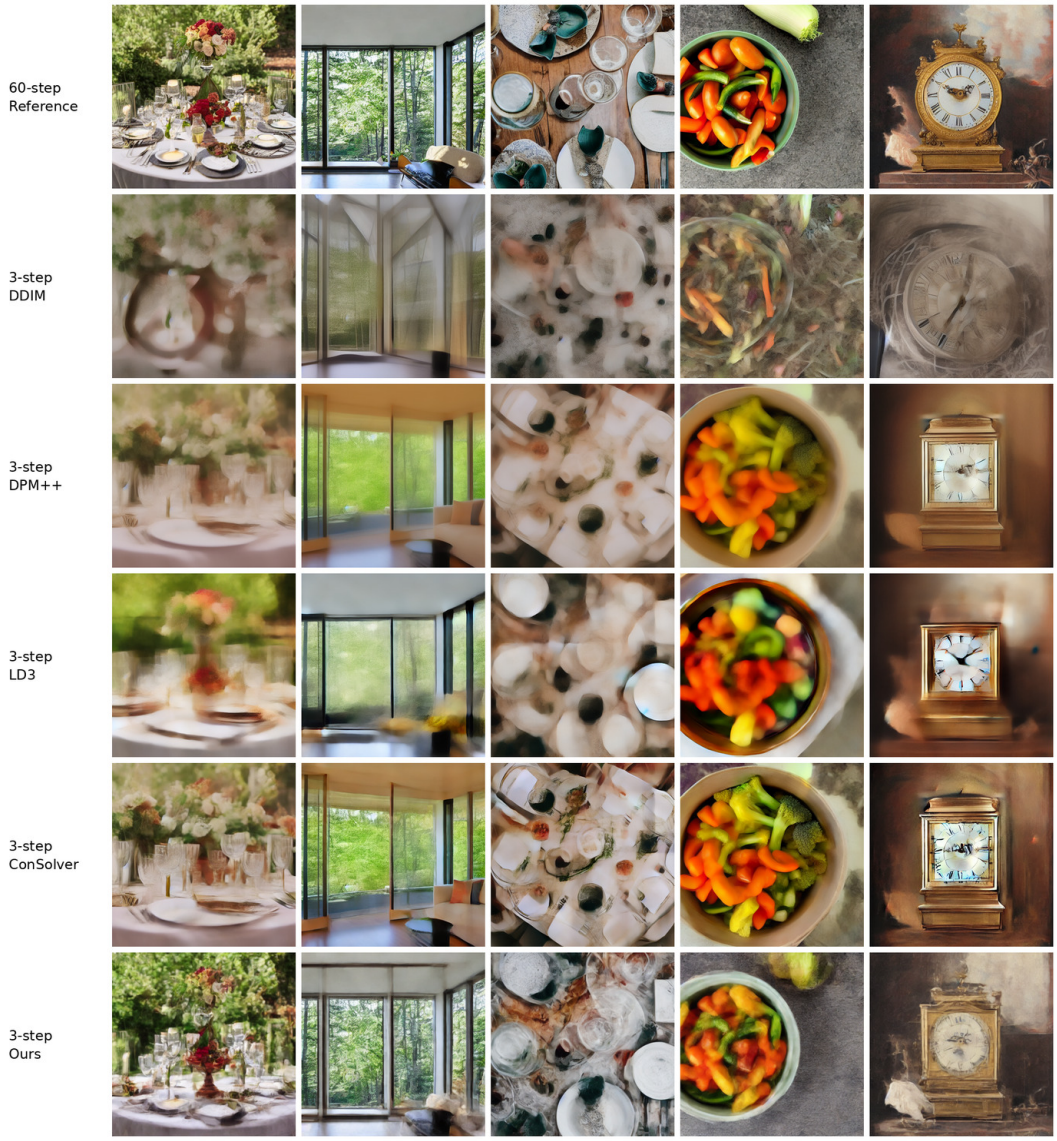


Figure 16: Visual comparison of 60-step Reference images and 3-step DDIM, DPM++, LD3, ConSolver, and Ours generations using Stable Diffusion 1.5 at CFG 5.

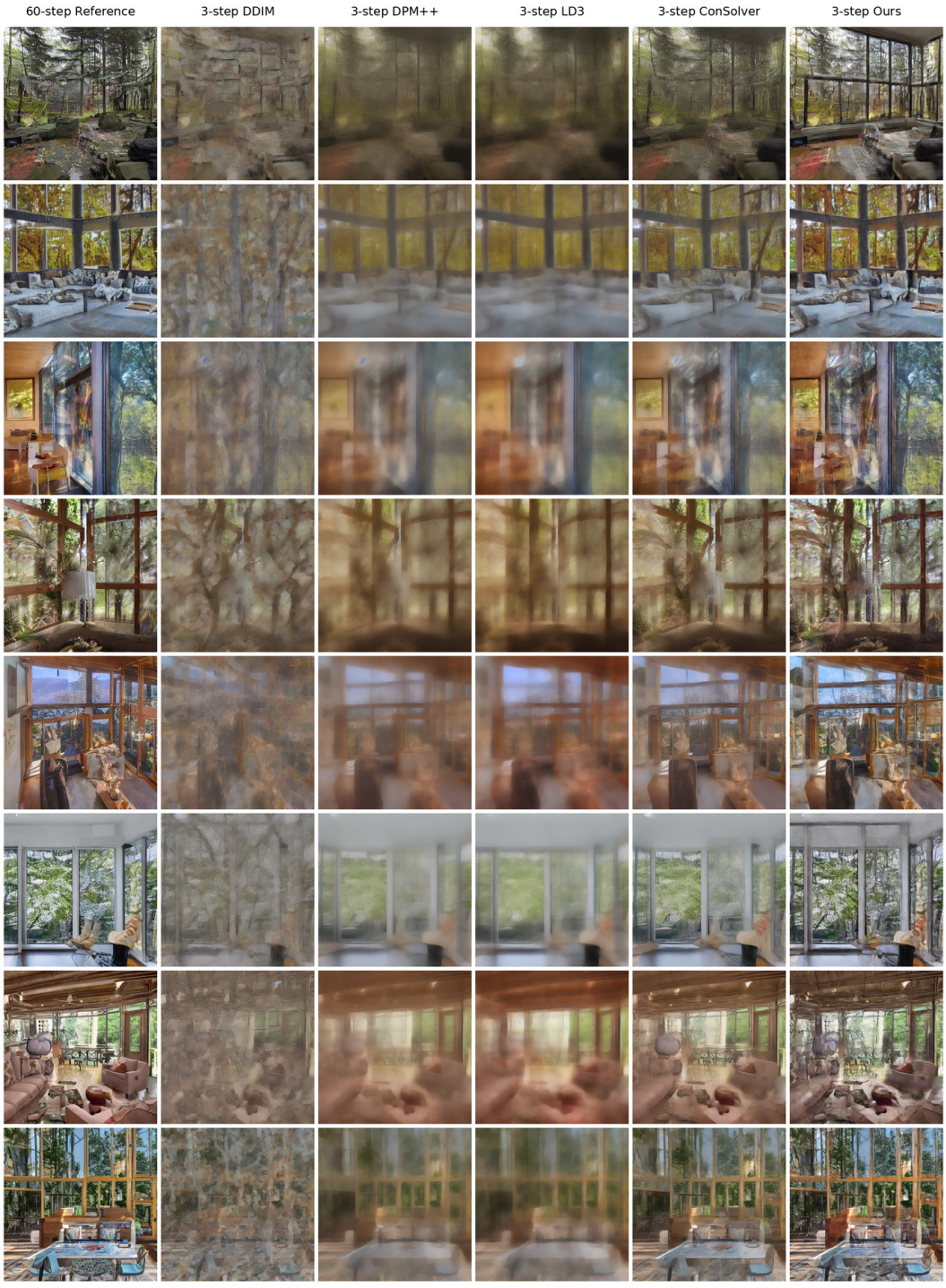


Figure 17: Comparison of 8 seed for the prompt “A large glass window in a living room.”

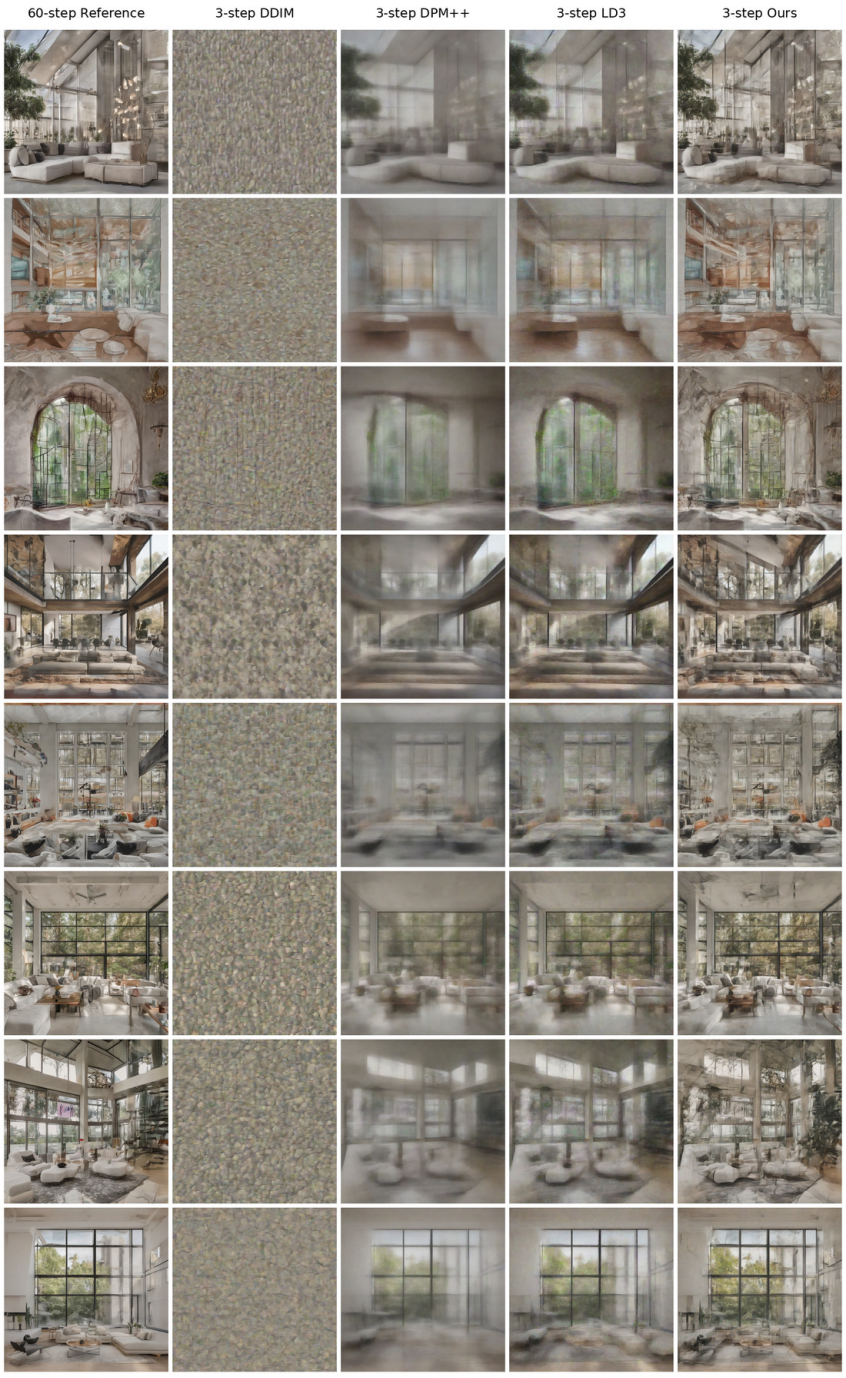


Figure 18: Comparison of 8 seed for the prompt “Comfortable, modern living room overlooking a wooded area”.

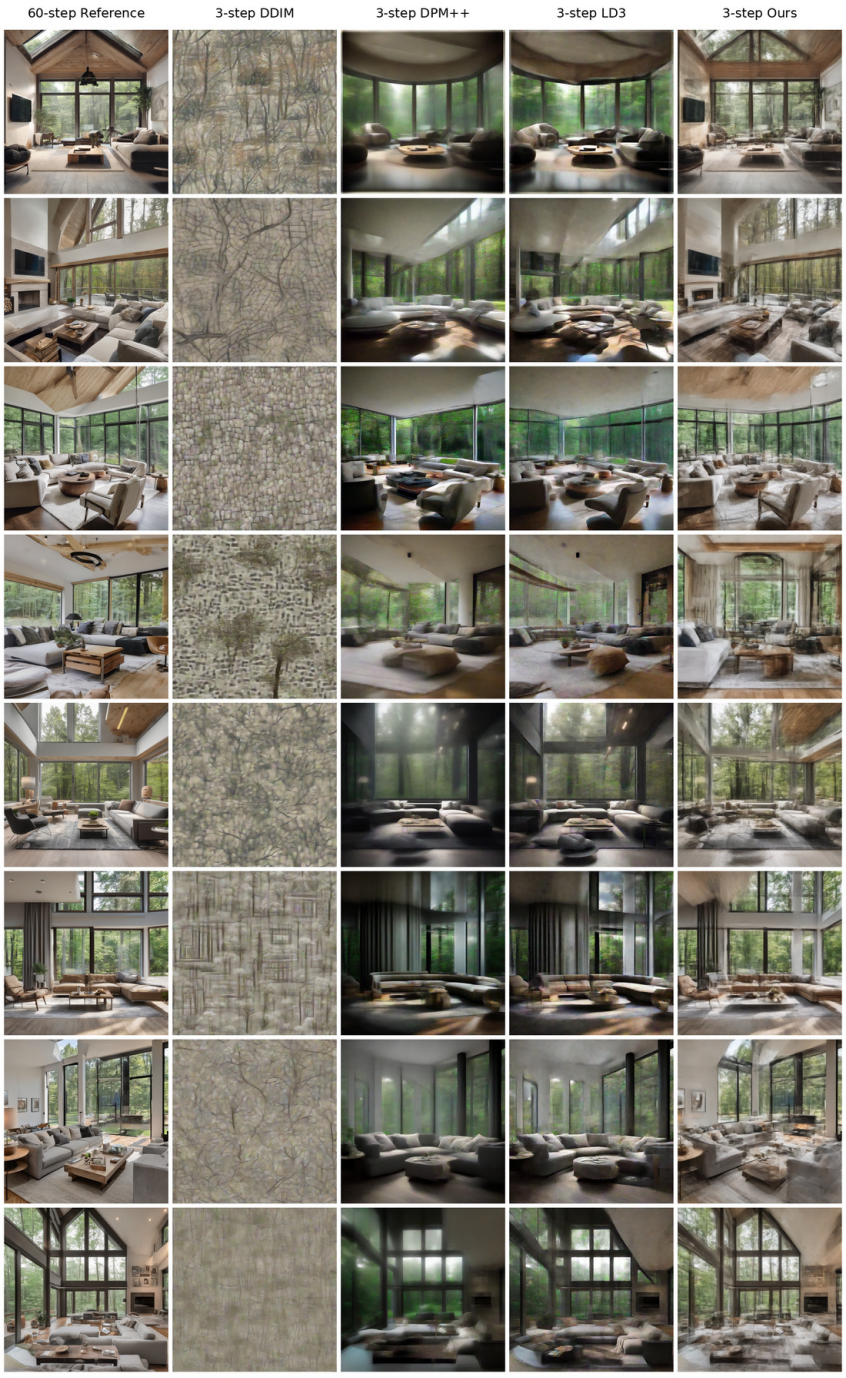


Figure 19: Comparison of 8 seed selection for the prompt “A living room with windows looking out onto a forest.”

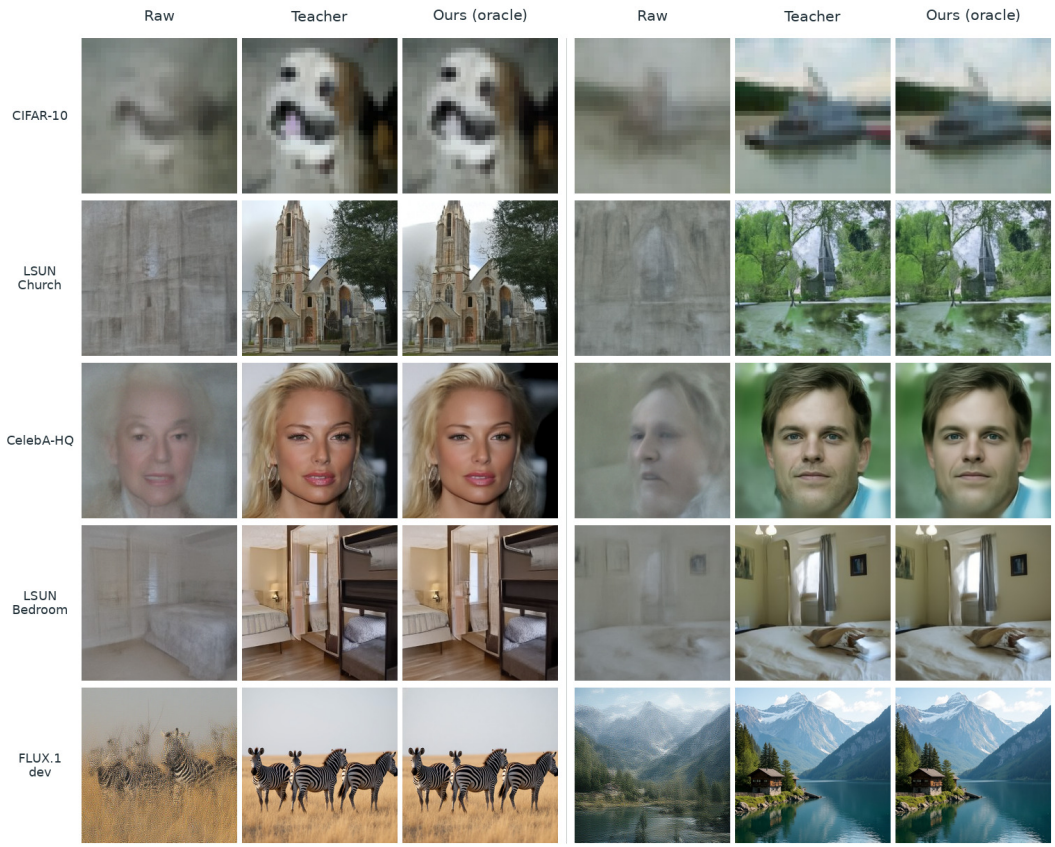


Figure 20: Additional oracle noise examples.

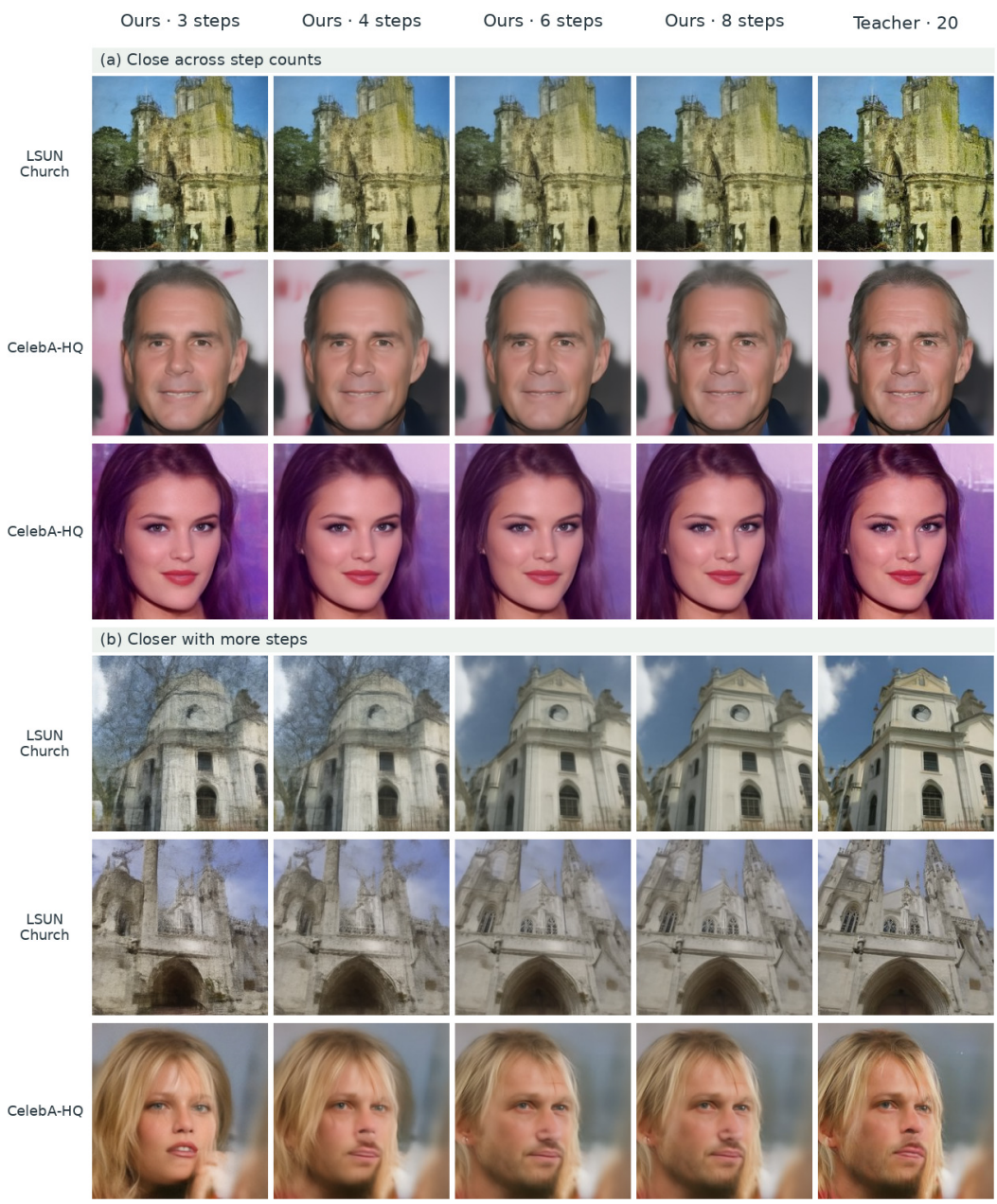


Figure 21: Selected examples of input-correction transfer across sampling steps. Top: examples remaining close to the teacher across step counts. Bottom: examples approaching the teacher as the step count increases. All outputs reuse the same frozen three-step corrector with a fixed exponent of 2.5 and noise norm normalization; each row shares the same original noise.

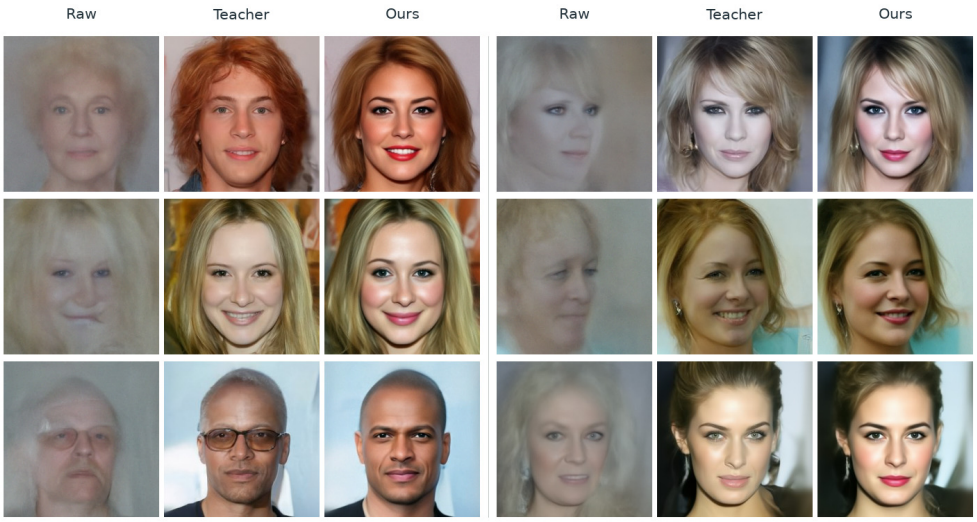


Figure 22: Additional cross-backbone correction examples on CelebA-HQ. Each triplet shows raw three-step DDIM, the paired 20-step teacher, and three-step DDIM with the transferred input corrector. The corrector was trained through a Bedroom backbone using CelebA teacher targets.