CORRECTION TRAINING FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

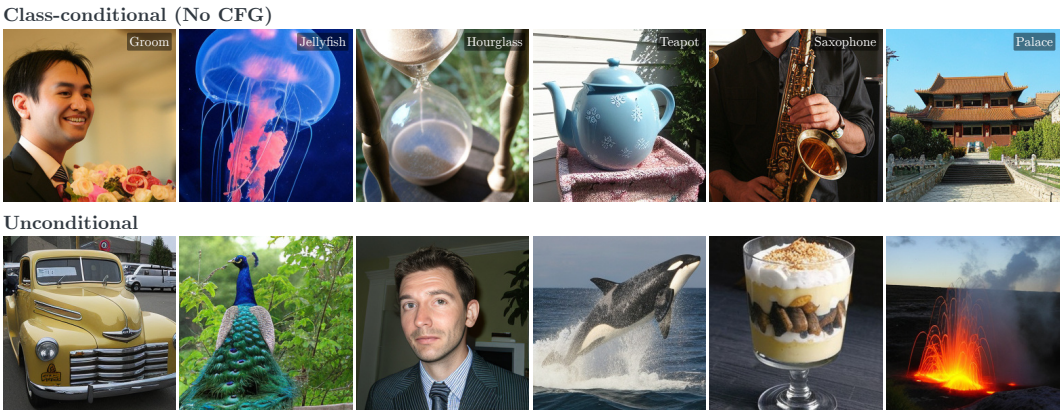


Figure 1: **ImageNet samples with correction.** *Without the use of guidance*, our correction method can achieve high-quality class-conditional (top) and *unconditional* (bottom) image samples.

ABSTRACT

Neural networks inevitably produce imperfect predictions, and accumulate errors in iterative sampling scenarios like generative diffusion models. In this paper, we introduce an iterative correction step for diffusion models. The corrector takes the base model's own prediction as input and is trained against the same ground-truth target as the base model. This makes correction an explicit learning task, with training inputs provided by the base model. During sampling, at each denoising step, we iteratively refine the denoiser's clean-sample prediction. Although the corrector is trained only on predictions from the base denoiser, we find that it can be applied repeatedly to its own outputs at inference time for even better results. This simple framework substantially improves sample quality while largely preserving diversity. We extensively evaluate our method across widely used architectures and training frameworks, including DiT, SiT, REPA, and JiT. Our method substantially improves upon previously reported FID scores for *unguided generation* and, notably, *unconditional generation*, where producing high-quality samples without guidance has been particularly challenging. Our method achieves performance competitive with classifier-free guidance (CFG), and combining correction with CFG further improves diversity and reduces oversaturation typical of CFG. Finally, we show that the additional inference cost of correction can be amortized by distilling the refined predictions back into the base diffusion model, retaining the quality gains without additional inference-time overhead. We hope that our findings encourage further efforts to move beyond diffusion models' long-standing reliance on guidance techniques.

1 INTRODUCTION

Diffusion models generate images through a sequence of denoising predictions (Ho et al., 2020; Song et al., 2020). At each sampling step, a neural network estimates clean data from a noisy input, and this estimate is used to update the sample. However, approximation errors in these predictions can accumulate to artifacts and incoherent structures. A common way to improve sample quality is classifier-free guidance (CFG) (Ho & Salimans, 2022), which, despite generating better quality samples, comes at the cost of reduced diversity and oversaturation.

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.

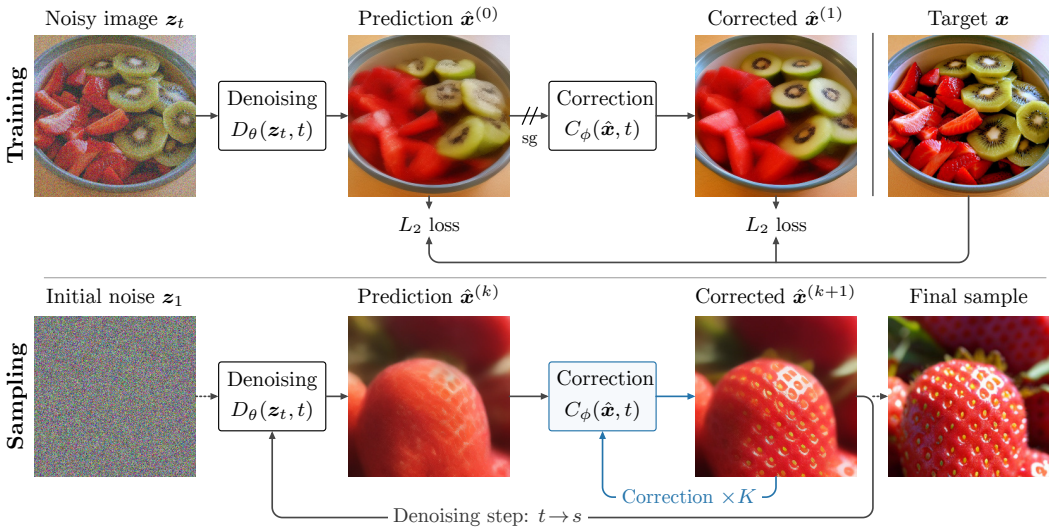


Figure 2: **Denoising and prediction correction.** During training, the denoiser and corrector predict the same clean target. Stop-gradient blocks the correction loss through the denoiser's prediction. During sampling, each denoising prediction is refined K times before the next sampling step.

In this paper, we take a more direct approach: refining the *model's own prediction*. We introduce a *corrector* that takes the base model's prediction as input and is trained against the same ground-truth target, as illustrated in Fig. 2. This makes correction an explicit learning task, with training inputs provided by the base model. During sampling, at each denoising step, we therefore iteratively correct the denoiser's clean-sample prediction. Although the corrector is trained only on predictions from the base denoiser, we find that it can be applied repeatedly to its own outputs at inference time without additional training. This iterative correction further improves sample quality, producing sharper images with more coherent structures.

We evaluate our method across class-conditional, unconditional, and text-to-image generation, using architectures and training frameworks including DiT (Peebles & Xie, 2023), SiT (Ma et al., 2024), REPA (Yu et al., 2024), and JiT (Li & He, 2026). Correction improves unguided sample quality across these settings with a relatively small amount of additional training. The gains remain when inference compute is matched, showing that they cannot be explained simply by increasing the sampling budget. In unconditional generation, correction also provides substantial improvements without any guidance signal. Comparisons with CFG on DiT and SiT show better quality at matched diversity, as measured by FID (Heusel et al., 2017) and recall (Kynkäänniemi et al., 2019). Qualitatively, correction show less oversaturation than CFG across the tested iterative correction. We present conditional and unconditional samples using correction in Fig. 1.

Finally, we show that corrector can be made as efficient as standard diffusion models while retaining its quality advantage. We distill corrected predictions back into the base diffusion model. The student learns to match predictions obtained through repeated correction and requires only one forward pass per sampling step. The resulting model retains much of the quality improvement while preserving the base model's architecture, parameter count, and inference cost. This allows the benefits of correction to be incorporated into the base model without additional correction passes during generation. We hope that the effectiveness, simplicity, and efficiency of corrector can help to bridge the long-standing gap between unconditional and conditional generation.

2 Learning to Correct Denoising Predictions

2.1 Background: Denoising and Sampling

Diffusion models learn to recover data from noisy observations (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020). Flow matching (Lipman et al., 2022; Liu et al., 2022b; Albergo et al., 2025) extends the diffusion paradigm by using a linear interpolation between noise and image. Specifically, given a clean sample $\boldsymbol{x} \sim p_{\text{data}}$ and independent noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$,

Algorithm 1 Training loss

```
# D: denoiser; C: corrector
# weight: correction loss weight
t = sample_time(x)  # per image
e = randn_like(x)
z = (1 - t) * x + t * e

pred = D(z, t)
refined = C(stopgrad(pred), t)

loss = mse(pred, x)
loss += weight * mse(refined, x)
```

Algorithm 2 Sampling with correction

```
# D: denoiser; C: corrector
# K: corrections per step
z = randn(shape)
for t, s in zip(times, times[1:]):
    pred = D(z, t)
    for _ in range(K):
        pred = C(pred, t)

    v = (z - pred) / t
    z = z + (s - t) * v
sample = z
```

the training input is

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

where $t = 0$ denotes data and $t = 1$ denotes noise. An $\boldsymbol{x}$-prediction network D_θ estimates the clean sample from $(\boldsymbol{z}_t, t)$. Its denoising objective is

$$\mathcal{L}_{\text{denoise}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t}\left[\|D_\theta(\boldsymbol{z}_t, t) - \boldsymbol{x}\|_2^2\right]. \tag{2}$$

At each sampling step, $D_\theta(\boldsymbol{z}_t, t)$ estimates the clean sample $\boldsymbol{x}$. The solver uses this prediction to construct the velocity $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t) = (\boldsymbol{z}_t - D_\theta(\boldsymbol{z}_t, t))/t$ for $t > 0$, and integrate from Gaussian noise at $t = 1$ toward $t = 0$. Sample quality is therefore affected by *modeling error*, arising from inaccuracies in the prediction $\hat{\boldsymbol{x}}^{(0)} = D_\theta(\boldsymbol{z}_t, t)$, and *numerical integration error*, arising from approximating a continuous sampling trajectory with a finite number of steps.

2.2 Modeling Error in Diffusion Sampling

A natural way to improve sample quality is to increase the accuracy of numerical integration, for example by increasing the number of sampling steps or using more advanced integration methods (Liu et al., 2022a; Zhang & Chen, 2022; Lu et al., 2022; Zhao et al., 2023; Lu et al., 2025). Once numerical integration error is small, additional computation offers diminishing returns, and sample quality can plateau because modeling error remains. Several studies have reported that gains from increasing the number of denoising steps quickly plateau (Liu et al., 2026; Ning et al., 2023) and that further increases can even degrade sample quality, attributing this degradation to error accumulation during sampling (Ning et al., 2023; Li et al., 2024a; Ning et al., 2024).

Another widely used strategy for improving sample quality is classifier-free guidance (CFG) (Ho & Salimans, 2022). Without CFG, generated samples often exhibit artifacts and incoherent structures, as illustrated in the top rows of Figs. 4 and 9. While highly effective, CFG can reduce sample diversity and produce oversaturated images. Other strategies also improve sample quality at the cost of additional inference compute, including alternative guidance techniques (Ahn et al., 2024; Karras et al., 2024) and search-based inference-time scaling methods (Ma et al., 2025; Singhal et al., 2025; Zhuo et al., 2025).

In this paper, we take a more direct approach to reducing modeling error: we train a corrector C_ϕ to refine predictions from the denoiser D_θ. Our aim is to generate high-quality samples without guidance by improving the predictions themselves.

2.3 Learning to Refine Model Predictions

Standard denoising training learns the mapping $\boldsymbol{z}_t \to \boldsymbol{x}$ by reconstructing images corrupted by a prescribed noise process. Given a noisy observation $\boldsymbol{z}_t$, the denoiser produces an initial prediction $\hat{\boldsymbol{x}}^{(0)}$, which may still contain inaccuracies. We introduce *prediction correction* as a reconstruction task that uses this prediction as its input. The corrector takes $\hat{\boldsymbol{x}}^{(0)}$ and predicts the same clean target $\boldsymbol{x}$, learning the mapping $\hat{\boldsymbol{x}}^{(0)} \to \boldsymbol{x}$ (Fig. 2).

When $\hat{\boldsymbol{x}}^{(0)}$ already captures much of the structure in $\boldsymbol{x}$, the initial denoising pass has already performed much of the reconstruction. The corrector can then build on this estimate and focus on the remaining inaccuracies. We refine $\hat{\boldsymbol{x}}^{(0)}$ by applying corrector C_ϕ again at the original time t:

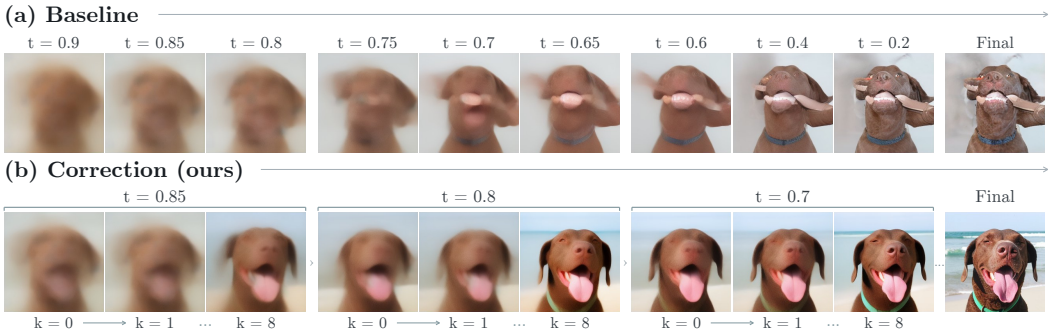


Figure 3: **Correction uses NFEs more effectively than denoising steps.** We show clean-sample predictions $\hat{x}$ along denoising trajectories without (a) and with (b) correction. At the same NFE budget, correction (b) produces sharper, more coherent samples than the baseline (a) with fewer steps. Each triplet in (b) shows the refined clean-sample prediction at fixed t.

$\hat{\boldsymbol{x}}^{(1)} = C_\phi(\hat{\boldsymbol{x}}^{(0)}, t)$. The corrected prediction directly estimates the clean sample $\boldsymbol{x}$. We train it with

$$\mathcal{L}_{\text{corr}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t}\left[\|C_\phi(\text{sg}(\hat{\boldsymbol{x}}^{(0)}), t) - \boldsymbol{x}\|_2^2\right], \tag{3}$$

where sg is the stop-gradient operation.

Starting from a pretrained denoiser, we jointly optimize $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoise}} + \lambda_{\text{corr}}\mathcal{L}_{\text{corr}}$, where λ_{corr} weights the correction task. The denoising term supervises the initial prediction, while the correction term supervises its refinement. The two tasks are trained jointly, with refinement inputs generated online by the current model. Algorithm 1 summarizes the loss computation in Python-style pseudocode.

2.4 Iterative Refinement at Inference

Training supervises only a single correction, $\hat{\boldsymbol{x}}^{(1)}$, of the initial denoising prediction $\hat{\boldsymbol{x}}^{(0)}$. At inference, however, the model can repeatedly refine its own outputs, further improving sample quality in our experiments. The output $\hat{\boldsymbol{x}}^{(k)}$ at the k-th correction iteration is another estimate of the same clean image. This estimate may still contain inaccuracies, motivating further applications of the learned refinement.

Starting from the denoising prediction, we apply K correction passes:

$$\begin{aligned} \hat{\boldsymbol{x}}^{(0)} &= D_\theta(\boldsymbol{z}_t, t), \\ \hat{\boldsymbol{x}}^{(k+1)} &= C_\phi(\hat{\boldsymbol{x}}^{(k)}, t), \qquad k = 0, \ldots, K-1. \end{aligned} \tag{4}$$

Fig. 3 illustrates how successive corrections refine intermediate predictions throughout the sampling trajectory. This provides a way to spend additional inference compute on improving the predictions at each time. At matched total inference cost, we find that allocating part of the budget to correction produces better samples than spending the full budget on additional integration steps.

3 Results

We evaluate correction on class-conditional and unconditional generation (Sec. 3.1) and text-to-image generation (Sec. 3.2). We also demonstrate that the base and correction models can share most parameters and that additional correction passes can be eliminated with distillation (Sec. 3.3). We detail our experimental settings in Appendix C.

3.1 ImageNet Generation

We evaluate our method on widely used transformer-based architectures, including DiT (Peebles & Xie, 2023), SiT (Ma et al., 2024), and JiT (Li & He, 2026). We initialize both the base denoiser and a separate corrector from publicly released checkpoints and jointly fine-tune them following Algorithm 1. Unless otherwise specified, we follow each baseline's training recipe and evaluation

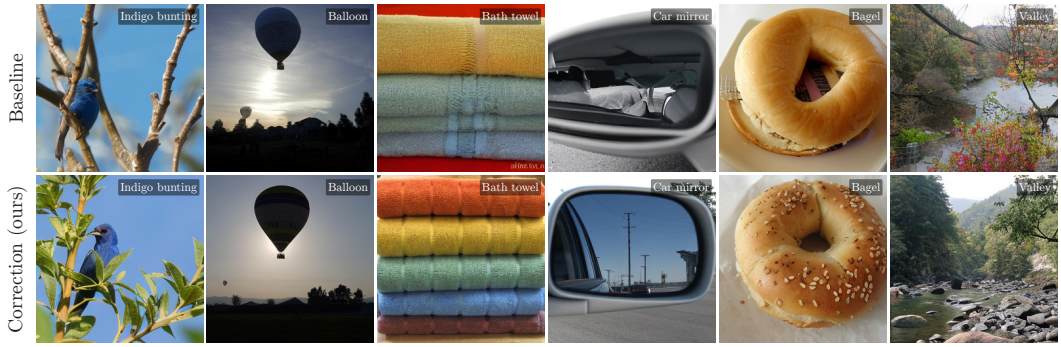


Figure 4: **Correction improves class-conditional sample quality.** We compare baseline (top) and correction (bottom), starting from the same noise in each column. Correction gives more coherent and detailed images for different classes and noises compared against the baseline.

Table 1: **Correction improves unguided class-conditional generation.** We compare FID-50K, precision, and recall on ImageNet-256 class-conditional image generation across 4 base models. We report consistent improvement on FID and precision over unguided baseline and it is competitive with CFG. All methods are evaluated at approximately 250 NFEs. See Tab. 14 for evaluation details.

DiT-XL/2	**FID** ↓	**Prec.** ↑	**Rec.** ↑
Unguided (baseline)	9.58	0.668	**0.676**
Corrective sampling	**2.07**	**0.811**	0.603
CFG (reference)	2.51	0.818	0.581

SiT$_{\text{REPA}}$-XL/2	**FID** ↓	**Prec.** ↑	**Rec.** ↑
Unguided (baseline)	5.53	0.708	**0.739**
Corrective sampling	**1.31**	**0.797**	0.703
CFG (reference)	1.35	0.803	0.699

SiT-XL/2	**FID** ↓	**Prec.** ↑	**Rec.** ↑
Unguided (baseline)	8.55	0.687	**0.669**
Corrective sampling	**1.92**	**0.812**	0.608
CFG (reference)	2.11	0.820	0.595

JiT-L/16	**FID** ↓	**Prec.** ↑	**Rec.** ↑
Unguided (baseline)	14.86	0.607	**0.672**
Corrective sampling	**2.96**	**0.766**	0.601
CFG (reference)	2.35	0.780	0.606

protocol. Number of function evaluations (NFE) under correction measures both denoiser steps and correction steps. We approximately match the total NFE by reducing the number of denoising steps as the number of correction iterations K increases.

Class-conditional generation. We present class conditional image samples in Fig. 4. Correction produces sharp, realistic, and diverse images, all *without applying any guidance*. The resulting samples exhibit realism competitive with that achieved by guidance-based methods such as classifier-free guidance (Ho & Salimans, 2022), perturbation guidance (Ahn et al., 2024) and autoguidance (Karras et al., 2024). See Appendix F.1 for more samples.

In Tab. 1, we evaluate generation using FID, precision, and recall (Heusel et al., 2017; Kynkäänniemi et al., 2019). Correction *significantly reduces FID at matched NFE* (DiT: 9.58 → **2.07**, SiT: 8.55 → **1.92**, and SiT$_{\text{REPA}}$: 5.53 → **1.31**.). This suggests that correction is a better allocation of inference compute than simply increasing denoising steps. These gains require little additional training: for SiT$_{\text{REPA}}$, 50K updates amount to just 1.25% of its 4M pretraining updates. Fig. 5 compares FID convergence with correction versus continued denoising. Fig. 6 shows how the number of correction iterations K affects the final samples and their fidelity–diversity trade-off. Correction effectively suppresses artifacts and improves object coherence, yielding lower FID and higher precision at the cost of some recall.

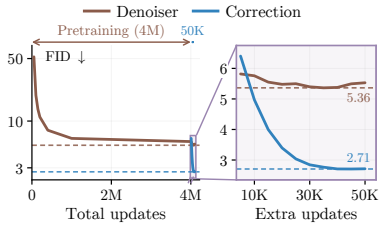
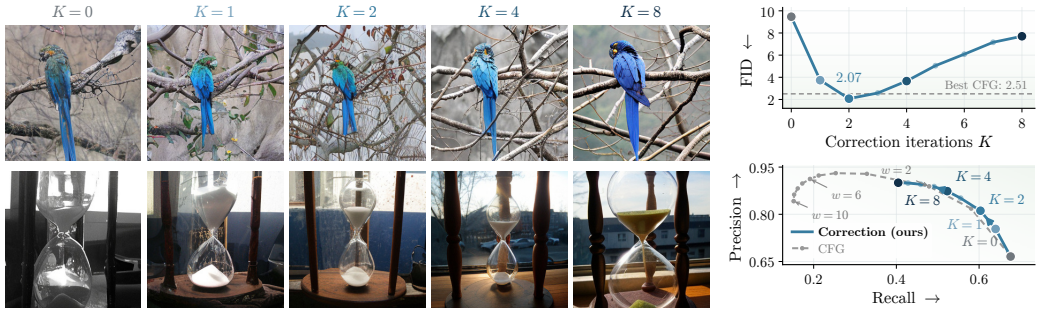


Figure 5: **Efficiency of correction training.** FID converges within relatively few corrector training steps.

Unconditional generation. Unconditional generation remains a challenging task since no class or text conditioning is available to constrain the outputs (Li et al., 2024b). We show that for unconditional generation, our prediction correction significantly improves image quality and coherence, all



(a) Effect of changing K (b) Generation quality and diversity

Figure 6: **Iterative correction provides a controllable refinement trajectory on ImageNet.** Left: Evolvement of the same samples over more correction steps. The main content remains the same while quality improves. Right: FID and precision–recall across K. Correction gives a controllable tradeoff between coverage and quality and lower minimum FID.



Figure 7: **Correction improves *unconditional* sample quality on ImageNet.** Without correction, generated samples exhibit artifacts and structural inconsistencies (top), while corrective sampling produces diverse and high-quality images (bottom). Each column share the same initial noise.

Method	FID ↓	Prec. ↑	Rec. ↑
Baseline	24.82	0.571	**0.699**
Correction ($K=1$)	17.80	0.620	0.679
Correction ($K=4$)	**5.37**	**0.782**	0.595
Autoguidance	9.37	0.699	0.630

Table 2: **Correction lowers unconditional FID without guidance.** 50K training updates; 250 NFE. Autoguidance uses $w=7$.

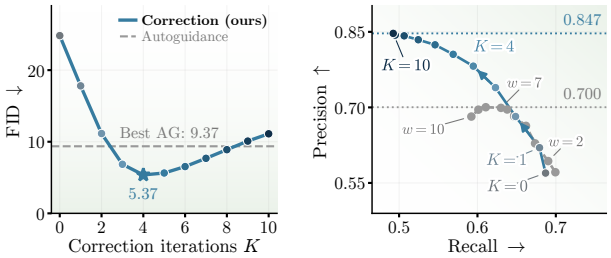


Figure 8: **Correction reaches lower FID and higher peak precision than autoguidance.** Dashed: best autoguidance FID. Dotted: each method's observed peak precision.

without requiring guidance. We present qualitative comparisons of corrected and uncorrected samples in Fig. 7. Correction removes structural inconsistencies and artifacts from the baseline samples while maintaining the main subject.

Quantitatively, we compare correction and autoguidance (Karras et al., 2024) in Tab. 2 on ImageNet unconditional generation. Correction with $K=4$ substantially improves FID over the baseline and the best evaluated autoguidance scale, while larger K trades recall for higher precision (Fig. 8).

3.2 Text-to-Image Generation

To assess our method on a more complex, practically relevant task, we evaluate text-to-image generation using NanoGen (Leng et al., 2026), a unified training framework for diffusion transformers. We use a pretrained DiT (Peebles & Xie, 2023) with a DDT head (Wang et al., 2026), operating in the latent space of the FLUX.2 VAE (Labs, 2025). We use the JourneyDB (Sun et al., 2023),



Figure 9: **Correction improves object structure in *text-to-image* generation.** Compared with no correction (top), correction (bottom) separates the violin bodies, clarifies the guitar and car geometry, and organizes cluttered interiors. All samples are without CFG; full prompts are in Appendix D.

Without CFG				*With CFG*			
Method	NFE	GenEval ↑	DPG-Bench ↑	Method	NFE	GenEval ↑	DPG-Bench ↑
Baseline	250	28.30	70.71	Baseline	250	68.08	83.38
Correction ($K = 6$)	245	**50.09**	**76.82**	**Correction** ($K = 2$)	246	**70.48**	**83.69**

Table 3: **Correction improves text-to-image alignment with and without CFG.** We observe consistent improvements on GenEval and DPG-Bench by applying correction over the baseline, both with and without CFG, evaluated at approximately matched NFE. See Table 14 for detailed settings.

Short-Caption, and Long-Caption subsets of the BLIP-3o pretraining mixture (Chen et al., 2025), with aspect-preserving preprocessing to 256×256. For the quantitative comparison, we jointly fine-tune the denoiser and corrector for 50K steps, adding a relatively small training cost compared with pretraining the base model.

Fig. 9 illustrates changes in object structure without guidance: correction separates the violin bodies, clarifies the guitar and car geometry, and organizes cluttered interiors. Quantitatively, Tab. 3 shows higher GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024) scores with correction at approximately matched NFE, first without CFG and then with correction applied before CFG. Both comparisons use the same-checkpoint baseline and DiffusionBench's native evaluation pipeline; sampling protocols are provided in Table 14, and additional guidance scales in Appendix C.3.

Combining CFG and correction. The two operations can be composed in either order: *CFG* $\rightarrow$ *correction* refines the guided prediction, whereas *correction* $\rightarrow$ *CFG* refines the conditional and null predictions separately before combining them. We favor the former for the guided qualitative examples and the main human-preference comparison based on our visual assessment of sample naturalness; the latter is used for the guided alignment benchmarks in Tab. 3. Neither order consistently leads on both human-rated quality and diversity across the evaluated settings; we provide the full comparison in Appendix E, Tab. 18.

Fig. 10 compares different combinations of CFG and correction. Stronger CFG alone can improve conditioning adherence and sample quality, but often at the cost of reduced diversity and oversaturated or stylized appearances. Lowering the guidance scale alleviates these effects, but may leave samples under-refined. We find that *applying correction jointly with CFG alleviates the diversity issue* while preserving the quality refinement. We observe that CFG alone produces visually similar samples, while combining CFG and correction leads to higher diversity (Fig. 10).

Unconditional generation on web images. Using the same model, we also present *unconditional* generation results in Appendix F.6. Surprisingly, correction improves realism while preserving both the diversity of web images and the overall structure of the corresponding uncorrected samples.



Figure 10: **Correction improves the diversity of CFG samples while preserving quality.** Columns compare no guidance ($w = 1$) and CFG ($w = 6$); rows compare no correction ($K = 0$) and correction ($K = 6$). All settings use the same eight seeds and number of denoising steps. Corrected samples show more varied bench shapes and viewpoints, with less oversaturation and a more natural appearance. Prompt: "a photo of a bench".

Human evaluation. To assess sample quality and diversity through human judgments, we conduct a preference study (Tab. 4) using 80 GenEval single-object prompts. Raters answer "Which image has better visual quality?" for individual images and "Which group is more diverse? Ignore image quality." for four-image groups. Raters prefer CFG + Correction over the baseline using DiffusionBench's default CFG scale (Leng et al., 2026) in more than 60% of responses. Evaluation details are provided in Appendix E.

Method	Quality ↑	Diversity ↑
Baseline	39.25%	39.75%
Correction	**60.75%**	**60.25%**

Table 4: **Correction is preferred for quality and diversity.** Released CFG ($w = 6$) versus CFG followed by correction ($w = 4$, $K = 8$).

3.3 Making correction efficient

Parameter-efficient correction through weight sharing. The denoiser and corrector can share a backbone while retaining separate lightweight input embedders, which contribute only a small fraction of the total parameters. On SiT-XL/2, this configuration achieves performance comparable to separate networks with roughly half the total parameters (Tab. 5). In the shared configuration, the backbone receives gradients from both the denoising and correction losses.

We also compare against a larger single denoiser with roughly matched parameter count, and find that iterative correction yields substantially larger gains than simply doubling the denoiser depth (Appendix B.4 Tab. 12).

Model	Params	FID ↓	K	NFE
Baseline	0.68B	8.55	0	250
Separate corrector	1.35B	**1.92**	2	249
Shared corrector	0.68B	2.14	3	248

Table 5: **A shared backbone retains similar correction quality.** Using shared layers from SiT-XL/2 on ImageNet with 100K steps leads to similar FID performance, suggesting that correction's advantage does not primarily come from using more parameters.

Model	Params	FID ↓	K	Steps	NFE ↓
Base	0.68B	8.62	–	250	250
Teacher	1.35B	**1.92**	2	83	249
Student	0.68B	2.04	–	250	250
Student	0.68B	2.26	–	83	**83**

Table 6: **Distillation achieves similar FID under much smaller NFE.** Distilling a SiT-XL/2 correction model into the base model leads to a similar FID performance with 3 times less NFE on ImageNet.

Distilling correction into the base model. Weight sharing reduces parameter overhead, but each correction iteration still requires an additional forward pass. We therefore distill corrected predictions into the original denoiser. The student is initialized from the teacher's base denoiser and trained

to match its predictions after k^* correction iterations:

$$\mathcal{L}_{\text{distill}} = w_{\text{distill}}(t)\|D_{\theta'}(x_t, t) - \hat{x}^{(k^*)}\|_2^2, \tag{5}$$

where k^* controls the number of teacher correction iterations. Tab. 6 reports results for class-conditional ImageNet-256 generation using SiT-XL/2 with $k^* = 2$. The student achieves comparable FID while reducing NFE by approximately 67%.

4 Discussion

It is not entirely clear why our simple method works as well as it does. Here we summarize our empirical findings on this question, with more details in Appendix A.

Lower gradient variance. We find that the corrector C_ϕ exhibits substantially lower raw gradient variance than the denoising baselines in our SiT_{REPA} comparison, while quickly improving generation quality (Fig. 12). These observations suggest that correction may be easier to optimize (App. A.1).

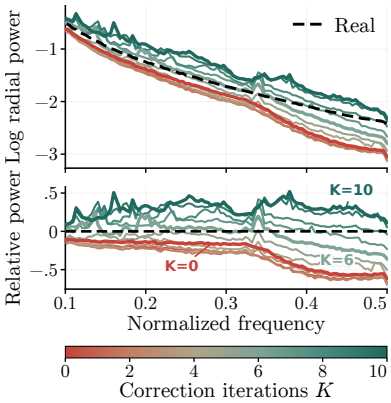


Figure 11: **Correction reduces the high-frequency deficit of generated samples.** Top: $\log_{10} P$. Bottom: $\log_{10}(P/P_{\text{real}})$. Dashed lines indicate real-image references.

Refinement in x-space. In a small-scale parameterization study, we observe a preference for refining predictions in x-space rather than v- or ϵ-space. We hypothesize that this may reflect the same principle underlying JiT's low-dimensional-manifold argument (Li & He, 2026), applied to the *input as well as the output* of C_ϕ. With x-predictions as both input and output, C_ϕ can focus its capacity on refining the low-dimensional structure of the data x, without needing to preserve the full-dimensional noise in v- and ϵ-space (App. A.1).

Repeated correction. The corrector learns from the denoiser's own predictions. In our toy study, the corrector sometimes specializes in particular errors, such as patch artifacts, which can be amplified under repeated application. When the denoiser evolves during training, its predictions may expose the corrector to a broader range of errors, helping it generalize to repeated refinement (App. A.2).

Sampling trajectory. Correction refines predictions before each sampler update, allowing local improvements to affect later states. Moderate correction also reduces the high-frequency deficit of baseline samples, while larger correction counts can overshoot the real-image spectrum (Fig. 11.)

5 Related work

Stacked generalization (Wolpert, 1992) trains a higher-level predictor on held-out predictions of base models to improve generalization. In diffusion models, self-conditioning (Chen et al., 2022) uses previous clean-data predictions as additional denoiser inputs, and recent concurrent works extend this idea to iterative refinement in language modeling and structured reasoning (Yoo et al., 2026; Helbling et al., 2026). We train an explicit corrector on denoiser predictions and reuse it across refinement iterations, without requiring the original noisy input. Idempotent Generative Networks (Shocher et al., 2024) also learn iterative mappings through an explicit idempotency objective. Our method retains the denoising objective and applies correction at each noise level to refine predictions before the sampling update.

6 Conclusion

We introduce correction training, which learns to refine a diffusion model's predictions using the original clean-data targets. Despite being trained for a single correction, the corrector supports repeated refinement at inference. Experiments on class-conditional and unconditional ImageNet generation and text-to-image generation demonstrate substantial improvements in sample quality. Correction can achieve competitive quality with guidance while better preserving diversity. Correction can also be distilled to remove inference overhead. Our analysis offers empirical insights into why correction works, but a theoretical understanding of its effectiveness and iterative behavior remains an open question for future research.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, refining hypothesis, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning dataset, support qualitative and thematic data analysis, interpreting results. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs. Additionally, we used generative AI tools for creating or modifying scientific figures or images, suggesting experimental parameters, creating or editing software code, creation of artifacts, summarizing or analyzing existing literature, brainstorming, sourcing/searching for information, editing a research paper to improve readability, identifying relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide pseudocode for our core algorithm and detailed training and evaluation configurations in main paper and Appendix C.1. We will publicly release our code and checkpoints to facilitate reproduction of our results.

REFERENCES

Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

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

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Alec Helbling, Andrey Bryutkin, Mauro Martino, Nima Dehmamy, and Hendrik Strobelt. Flow reasoning models: Scaling reasoning through iterative self-refinement. *arXiv preprint arXiv:2606.29150*, 2026.

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

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

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

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

Junwan Kim, Jiho Park, Seonghu Jeon, and Seungryong Kim. Better source, better flow: Learning condition-dependent source distribution for flow matching. *arXiv preprint arXiv:2602.05951*, 2026.

Myunsoo Kim, Donghyeon Ki, Seong-Woong Shim, and Byung-Jun Lee. Adaptive non-uniform timestep sampling for accelerating diffusion model training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2513–2522. IEEE, 2025.

Yuya Kobayashi, Masato Ishii, Yuhta Takida, Takashi Shibuya, and Yuki Mitsufuji. Spectral prior for reducing exposure bias in diffusion models. *arXiv preprint arXiv:2607.22091*, 2026.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Xingjian Leng, Jaskirat Singh, Zhanhao Liang, Ethan Smith, Martin Bell, Aninda Saha, Yuhui Yuan, and Liang Zheng. Diffusionbench: On holistic evaluation of diffusion transformers. *arXiv preprint arXiv:2606.24888*, 2026.

Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. In *International Conference on Learning Representations*, volume 2024, pp. 16816–16838, 2024a.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37: 125441–125468, 2024b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Yuxi Liu, Konpat Preechakul, Kananart Kuwaranancharoen, and Yutong Bai. The serial scaling hypothesis. In *International Conference on Learning Representations*, volume 2026, pp. 106809–106839, 2026.

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.

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

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. *arXiv preprint arXiv:2301.11706*, 2023.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 15167–15189, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2024.

Assaf Shocher, Amil Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei Efros. Idempotent generative network. In *International Conference on Learning Representations*, volume 2024, pp. 35440–35460, 2024.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Advances in neural information processing systems*, 36:49659–49678, 2023.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 40633–40642, 2026.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Jaehoon Yoo, Wonjung Kim, Floor Eijkelboom, Chanhyuk Lee, Nicholas M Boffi, Seunghoon Hong, and Jinwoo Kim. Self-conditioned flow map language models via fixed-point flows. *arXiv preprint arXiv:2607.00714*, 2026.

Meng Yu and Kun Zhan. Frequency regulation for exposure bias mitigation in diffusion models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10370–10378, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15329–15339. IEEE, 2025.

APPENDIX CONTENTS

A Discussion **14**
A.1 What differs when learning from model predictions? . . . 14
A.2 Why does single-step training support repeated correction? . . . 17
A.3 How does correction alter the sampling trajectory? . . . 17

B Design choices and alternative realizations **22**
B.1 Corrector interface . . . 22
B.2 Training configuration . . . 24
B.3 Inference configuration . . . 27
B.4 Alternative resource and training regimes . . . 27

C Implementation Details **27**
C.1 Training configurations . . . 27
C.2 Corrector parameterization . . . 27
C.3 Figure and Table Details . . . 29

D Full Prompts for Text-to-Image Examples **31**

E Correction vs. Classifier-Free Guidance. Comparison and its synergy **32**
E.1 Comparing Correction and CFG . . . 32
E.2 Combining CFG and Correction . . . 32
E.3 Human evaluation . . . 34

F More Samples **35**
F.1 Effect of Correction Depth . . . 35
F.2 SiT: Random-Class Generation . . . 36
F.3 SiT_{REPA}: Random-Class Generation . . . 37
F.4 JiT: Random-Class Generation . . . 38
F.5 Unconditional Generation . . . 39
F.6 Unconditional Generation from a Text-to-Image Model . . . 40

A DISCUSSION

The denoiser and corrector are trained against the same clean targets, yet correction can improve generation with a short additional-training budget. What makes correction so effective? To answer these questions, we first compare correction with continued and deeper denoising (Sec. A.1), and then examine repeated refinement (Sec. A.2).

A.1 WHAT DIFFERS WHEN LEARNING FROM MODEL PREDICTIONS?

Additional-training and capacity controls. We examine how quickly the corrector converges by comparing correction training with three baselines: continued training of the base denoiser and two

deeper denoisers with different initialization schemes. We use the pretrained SiT_{REPA}-XL/2 denoiser with 28 transformer layers from Tab. 1. For the deeper denoisers, we append 28 transformer layers before the final unembedding layer, matching the total number of transformer layers in the denoiser–corrector pair. The added layers are either randomly initialized or copied from the pretrained denoiser, with the latter matching our corrector initialization. We train all configurations for 50K steps and report their progress in Fig. 12. We use a single correction pass ($K = 1$) to isolate its benefit from the additional gains of iterative refinement.

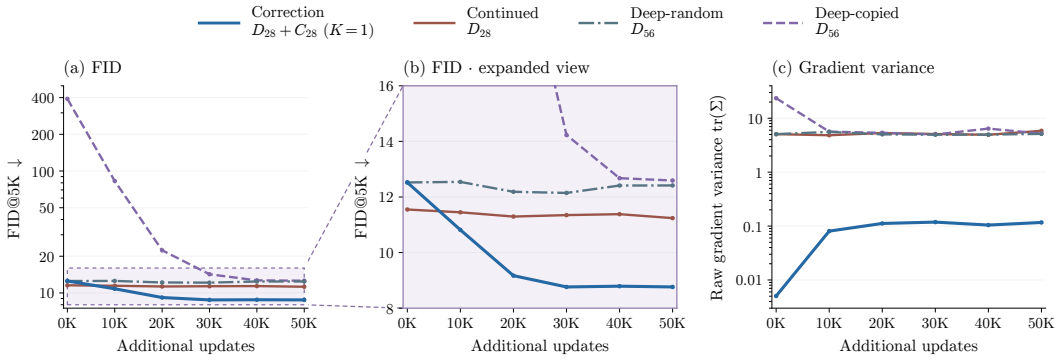


Figure 12: **Correction improves generation quality beyond continued denoiser training or parameter-matched deeper denoisers.**

As can be seen in Fig. 12 (a) and (b), correction improves FID substantially faster over the 50K-update training window we used, whereas continued training and both deeper-denoiser controls improve much less. Importantly, this gain appears with only a *single correction pass*: despite using the same total sequential transformer depth as the 56-layer denoisers, the $D_{28} + C_{28}$ correction model achieves substantially better generation quality. These results suggest that *additional training or model depth alone cannot explain the gains*, pointing to an advantage of learning to refine an already structured prediction over learning directly from noisy observations.

Correction exhibits lower absolute gradient variance. Fig. 12 (c) provides one possible optimization-level explanation. Gradients induced by the correction objective exhibit approximately 50 times lower variance than those induced by the denoising objective. We observe the same qualitative trend on a second architecture, JiT-L/16, where the measured variance decreases from 2.492×10^{-3} for denoising to 1.440×10^{-3} for correction (Fig. 13). Together with the faster FID improvement above, this raises a natural question: what differences between the denoiser and corrector account for their different learning behavior?

One possibility is that the denoiser finds it harder to learn stochastic clean targets from noisy observations than the corrector does from predictions already aligned with those targets. We probe the role of target stochasticity by replacing the denoiser's sampled targets with their analytic conditional means. We then examine whether correction's effectiveness depends on using clean-sample predictions in x-space as both input and output.

Network	Variance ($\times 10^{-3}$)
Denoiser	2.492
Corrector	**1.440**

Figure 13: **Correction has lower gradient variance under training-time sampling.** Gradient variance is measured over the feature-processing parameters of JiT-L/16, excluding timestep-embedding and class-conditioning parameters, using 8K examples with random t drawn from the training-time sampling distribution.

Target uncertainty alone does not explain the gain. A noisy observation z_t can be consistent with several clean samples, which introduces significant stochasticity in the denoising objective. For example, consider the extreme case of $t = 1$, where the input is pure Gaussian noise. The sampled noise $\boldsymbol{z}_1 = \boldsymbol{\epsilon}$ can be mapped to any data sample $\boldsymbol{x}$, and vice-versa. This randomness in the input-output mapping can cause per-sample gradients to conflict with one another resulting in high-variance gradients. In fact, reduc-

ing this variance through optimal transport and other data-noise couplings is an active area of research (Pooladian et al., 2023; Kim et al., 2025; 2026).

To isolate the effect of variance due to a stochastic target in standard denoiser training, we conduct an experiment on a small scale dataset (243 64×64 RGB faces, see Fig. 15). In it, we train a denoiser use a deterministic target: the exact conditional mean and compare it with another standard denoiser trained on a stochastic target. If target variance hurts optimization, we would expect worse results from the stochastic-target denoiser.

We train two identical four-layer SiT denoisers from scratch, using either a stochastic target $\boldsymbol{x}$ or the analytic target $\boldsymbol{x}_{\text{analytic}}$ defined as

$$w_i(\boldsymbol{z}_t, t) = \exp\left(-\frac{\|\boldsymbol{z}_t - (1-t)\boldsymbol{x}_i\|_2^2}{2t^2}\right), \qquad \boldsymbol{x}_{\text{analytic}}(\boldsymbol{z}_t, t) = \frac{\sum_{i=1}^N w_i(\boldsymbol{z}_t, t)\,\boldsymbol{x}_i}{\sum_{i=1}^N w_i(\boldsymbol{z}_t, t)}. \tag{6}$$

$$\mathcal{L}_{\text{analytic}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\,\boldsymbol{\epsilon},\,t}\left[\lambda(t)\,\|D_\theta(\boldsymbol{z}_t, t) - \boldsymbol{x}_{\text{analytic}}(\boldsymbol{z}_t, t)\|_2^2\right]. \tag{7}$$

We use $\lambda(t) = [d\,\max(t, 0.05)^2]^{-1}$ (velocity space weighting (Li & He, 2026)), where $d = 3 \cdot 64^2$ accounts for pixel normalization. Tab. 7 reports similar validation loss and mean gradient noise scale (GNS) across the two runs, despite the lower raw gradient variance with the analytic target. Additionally, structural accuracy does not improve consistently between the two runs either. Thus, removing the stochasticity in the target alone was insufficient for a consistent generation benefit in this experiment.

Table 7: **Removing target uncertainty does not consistently improve generation.** On the face toy dataset, sample quality, gradient noise scale, and structural validity remain largely unchanged, while gradient variance decreases. We report two-seed means after 10K updates. Validation loss is measured against a common exact-mean target, and gradient statistics include all parameters under the training-time timestep distribution. "Structure" denotes the percentage of 512 generated samples per seed that pass a partial check of eye and nose counts and placement.

Training target	Val. loss ↓	Variance	$\|\bar{g}\|$	GNS	Structure ↑
Sampled x	0.0068	0.066	0.047	31.08	47.66%
Analytic mean	0.0070	0.056	0.042	31.36	42.87%

Effect of correction parameterization. A second possible difference is the representation presented to the corrector. Unlike the denoiser, which operates on the noisy state $\boldsymbol{z}_t$, our default corrector receives the denoiser's clean-sample prediction $\hat{\boldsymbol{x}}^{(0)}$. To probe whether this representation matters, we run a from-scratch CIFAR-10 parameterization sweep using small SiT models. Fig. 14 compares same-space correctors whose inputs and outputs are parameterized in $\boldsymbol{x}$, $\boldsymbol{\epsilon}$, or $\boldsymbol{v}$ space.

We further investigate the advantage of x-space correction discussed in Sec. 4. In this setting, $\boldsymbol{x}$-space correction gives the lowest FID among the tested same-space variants. Notably, even when the base denoiser predicts in $\boldsymbol{\epsilon}$- or $\boldsymbol{v}$-space, converting its prediction to $\boldsymbol{x}$-space before correction improves FID. While the preferred parameterization may vary across architectures and datasets, these results indicate that the representation presented to the corrector is an important design choice.

Empirical picture. Taken together, these experiments suggest that correction defines a meaningfully different learning problem from direct denoising. Its advantage cannot be explained solely by additional training or model capacity, and reducing target uncertainty alone does not reproduce the gain. We also observe lower absolute gradient variance during correction training and find that the representation presented to the corrector materially affects performance. One possible interpretation is that a denoiser must reconstruct clean data directly from a noisy observation, whereas the corrector operates on a model prediction in which substantial image structure has already been resolved and can focus on refining the remaining errors. Together, these observations support the view that learning to refine an already structured model prediction provides a distinct and effective alternative to learning the denoising mapping directly.

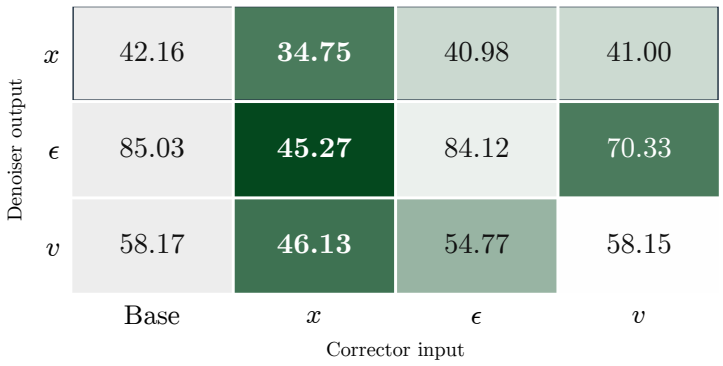


Figure 14: **Effect of correction parameterization on CIFAR-10.** In this controlled sweep, x-space input and output achieve the lowest FID among the tested same-space correctors.

A.2 Why does single-step training support repeated correction?

Proposal update rate and repeated correction. To probe one factor that may affect repeated application, we vary how quickly the proposal generator evolves during training. On the synthetic face dataset, we train a four-layer denoiser and separate four-layer correctors from scratch for 10K updates. Each corrector receives the detached clean prediction of a denoiser whose parameters follow an EMA, $\bar{\theta}_n = \beta\bar{\theta}_{n-1} + (1-\beta)\theta_n$, where n indexes training updates. At evaluation, all correctors are applied to the same base denoiser, isolating differences induced by the training-time proposal schedule.

Fig. 15 shows that repeated correction is sensitive to how its training proposals are generated. With very slow proposal updates ($\beta = 0.9999$), patch-pattern artifacts already appear after one correction and become more pronounced under repetition. In contrast, with a moderately evolving proposal generator ($\beta = 0.9$), increasing from $K = 1$ to $K = 3$ improves the mean structural pass rate from 83.6% to 89.1%; raw proposals ($\beta = 0$) show a similar, though weaker, improvement. These observations suggest that exposure to an evolving proposal distribution may help the corrector generalize beyond the single correction seen during training.

A.3 How does correction alter the sampling trajectory?

Refinement within a denoising step. Fig. 3 provides a qualitative view of how correction changes intermediate clean-sample predictions during sampling. Without correction, structural errors in an intermediate prediction can persist into subsequent denoising steps because the prediction is immediately used to update the noisy state. Correction instead refines the prediction at fixed t before this update. In the examples in Fig. 3, successive corrections sharpen object structure and suppress visible artifacts within a single denoising step, so the subsequent sampler state is constructed from an improved prediction.

Frequency-domain effects of correction. To further characterize the changes induced by correction, we examine the frequency content of generated samples. Prior work (Yu & Zhan, 2025; Kobayashi et al., 2026) has reported lower subband energy in reverse-process samples than in forward-corrupted real data. This motivates examining whether correction recovers frequency components that are underrepresented in baseline samples.

We compare the power spectra of samples generated with $K = 0, \ldots, 10$ correction iterations against those of real images. As shown in Fig. 11, the baseline samples exhibit a spectral deficit, particularly at high frequencies. Increasing the number of correction iterations reduces this deficit, with $K = 6$ yielding a spectrum that approximately matches the real-image reference. Further iterations can raise both low- and high-frequency power above the reference, indicating that the effect depends on the correction count.

Figures 16–18 show the individual correction passes for three examples from SiT_{REPA}-XL/2. Both trajectories use the same initial noise and 15 Euler denoising steps: the baseline and $K = 8$ sampler use 15 and 135 network evaluations, respectively.

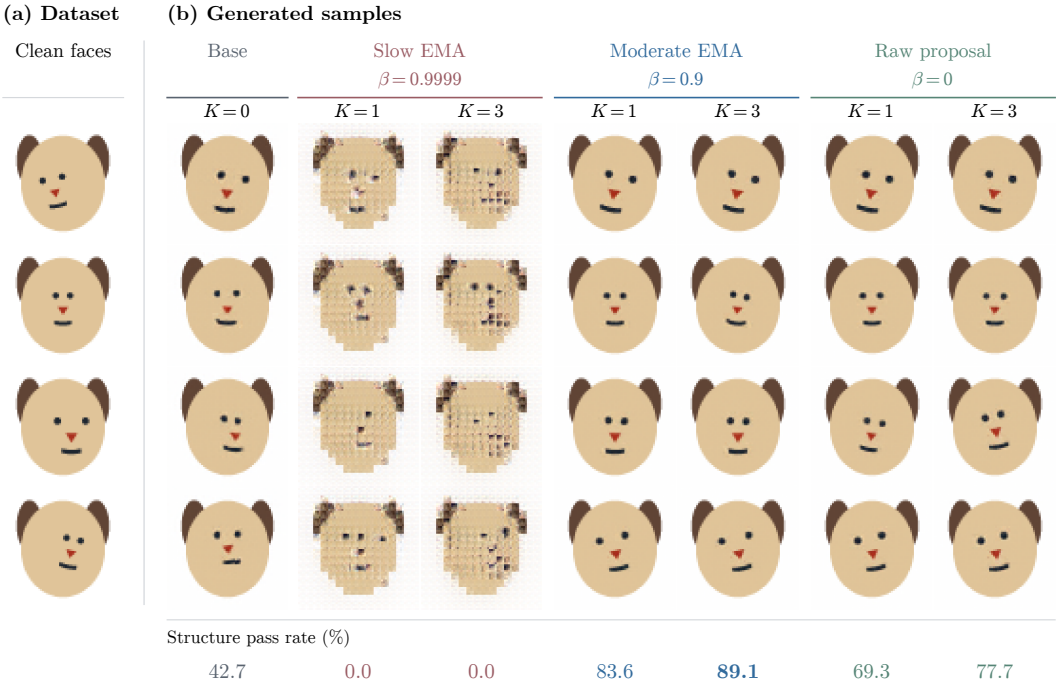


Figure 15: **Very slow proposal updates can amplify patch artifacts under repeated correction.** Correctors trained with different proposal EMA decays are applied to the same base denoiser. Numbers mean the partial facial-structure pass rate over 1.5K samples.

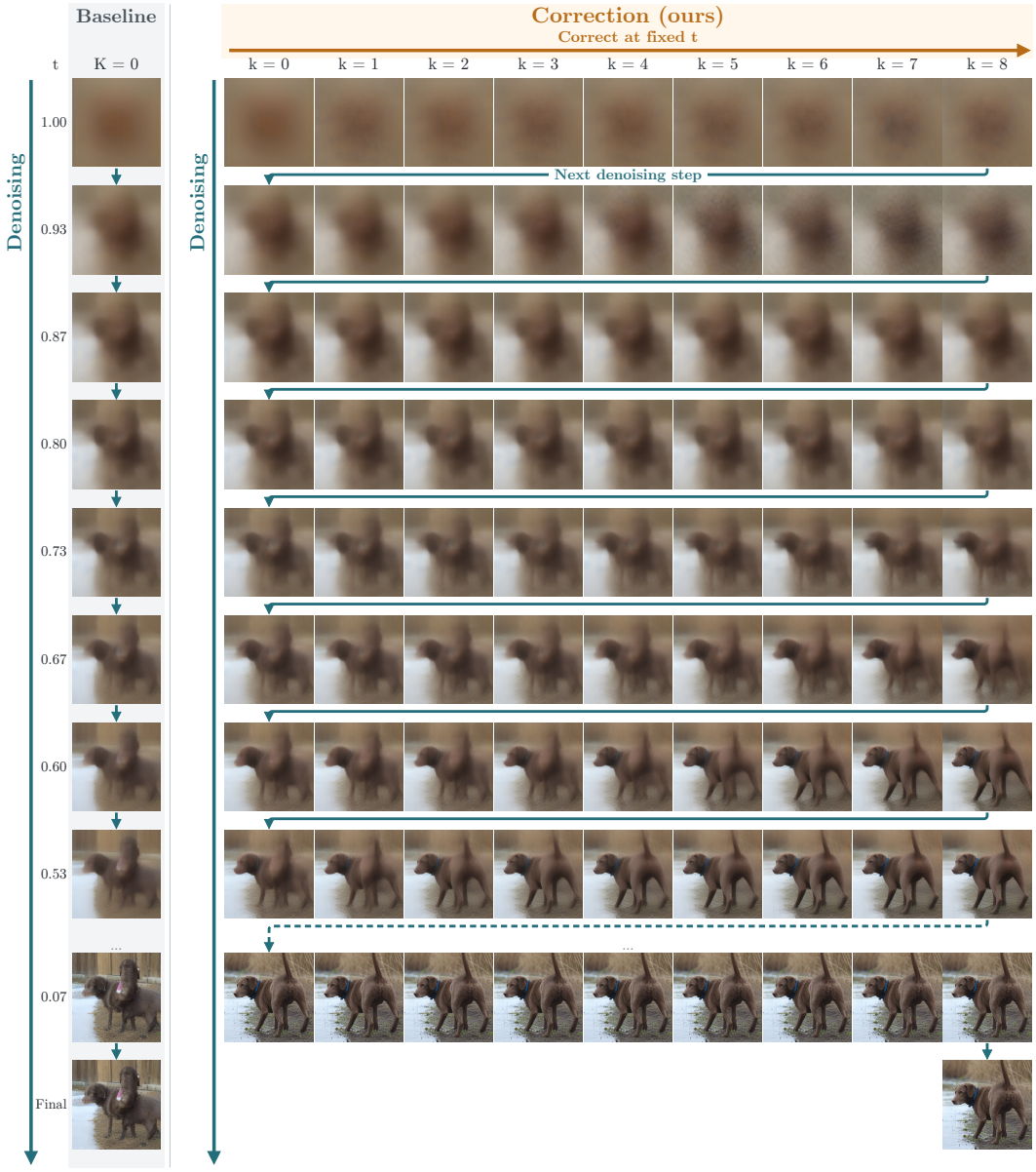


Figure 16: **Correction refines intermediate predictions before denoising continues.** Baseline and correction start from the same noise and use 15 denoising steps. Each row shows all eight correction passes at fixed t; later denoising rows are omitted. The bottom row shows the actual final samples.

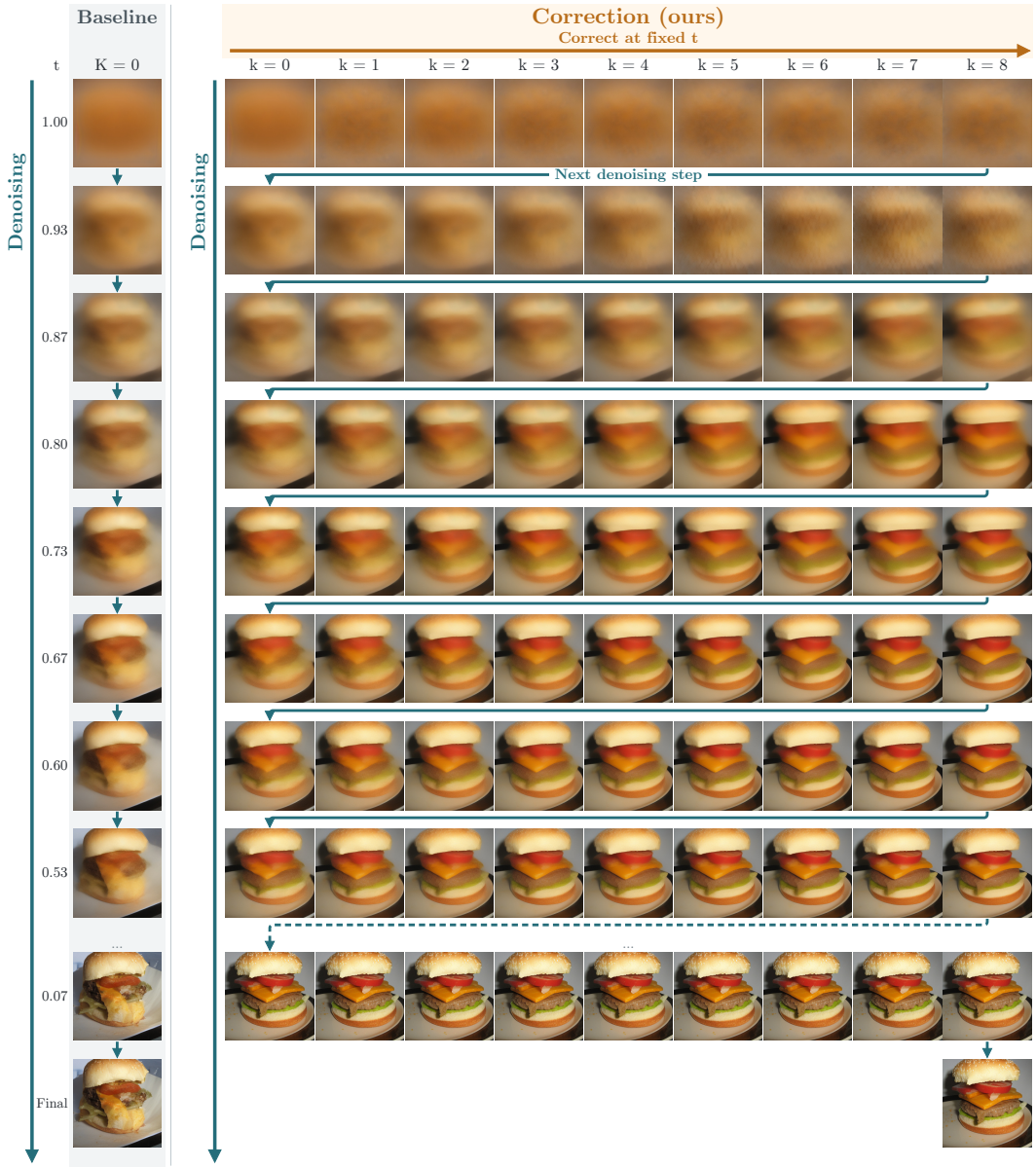


Figure 17: **Repeated correction sharpens the structure of intermediate predictions.** Baseline and correction start from the same noise and use 15 denoising steps. Each row shows all eight correction passes at fixed t; later denoising rows are omitted. The bottom row shows the actual final samples.

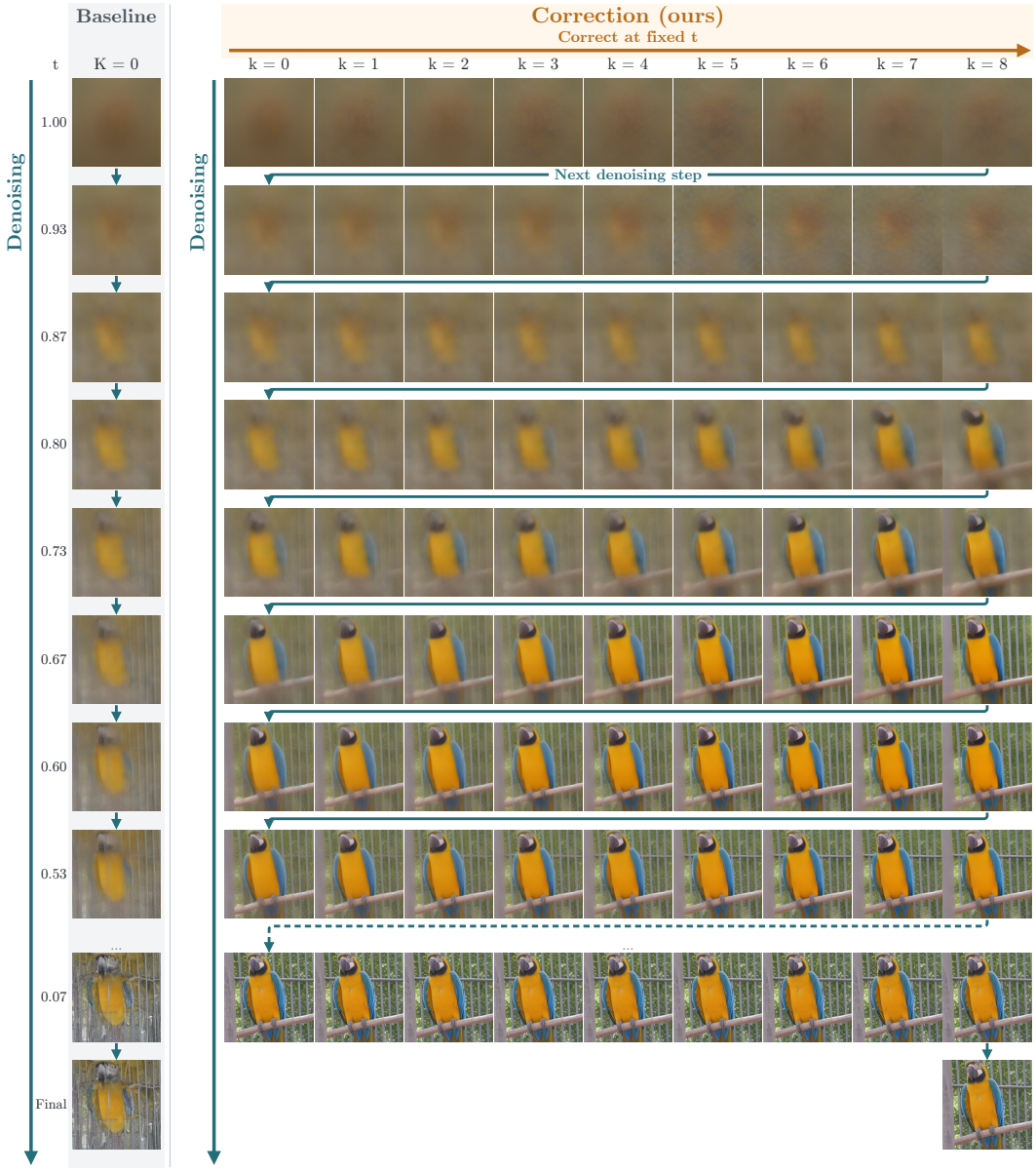


Figure 18: **Correction refines intermediate predictions before denoising continues.** Baseline and correction start from the same noise and use 15 denoising steps. Each row shows all eight correction passes at fixed t; later denoising rows are omitted. The bottom row shows the actual final samples.

B Design choices and alternative realizations

A practical realization of correction training. In this section, we examine the design choices in our framework. The default configuration used in the main paper is a simple implementation compatible with pretrained models: we use a separate corrector initialized from the base denoiser, omit the noisy input x_t, retain the timestep-conditioning pathway to preserve the original architecture, and apply stop-gradient to prevent the correction loss from updating the denoiser. Many other combinations of these design choices are possible within our correction training framework. Instead of exhaustively evaluating all combinations, we vary one component at a time from the default configuration to understand how each choice affects performance.

B.1 Corrector interface

Access to the noisy state x_t from corrector. Prior approaches (Chen et al., 2022) to learning from model predictions often also provide the original input. In our diffusion setting, this corresponds to giving the corrector access to x_t in addition to the denoiser's prediction. Tab. 8 shows that, on JiT, correction without x_t achieves comparable FID to correction with x_t, while requiring no architectural changes to the denoiser.

The SiT comparison in Fig. 19 further shows how this choice affects the precision–recall trade-off. Here, p-only denotes conditioning on the proposal $\hat{\boldsymbol{x}}$ alone, whereas $p + x_t$ provides both the proposal and the noisy input. Proposal-only correction achieves higher precision at slightly lower recall. One possible explanation is that, without access to the original noisy input, the corrector is less constrained by it and can modify the proposal more aggressively.

Variant	K=0	K=1	K=2	K=3	K=4	K=5
Proposal only ($\hat{x}$)	24.63	17.96	14.07	**13.35**	14.27	16.06
With x_t conditioning	24.72	18.22	14.84	13.79	14.67	16.58

Table 8: **Proposal-only correction performs comparably to x_t-conditioned correction.** Unguided FID50K on JiT-B/16 at approximately 250 NFE.

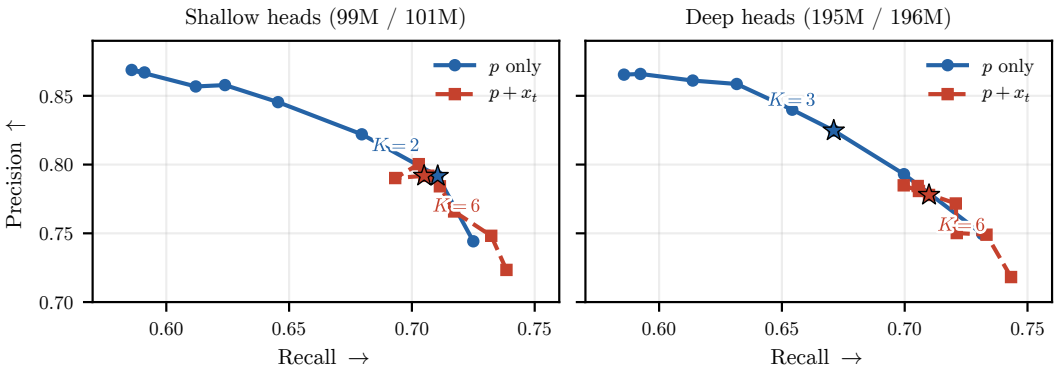


Figure 19: **Proposal-only correction reaches higher peak precision.** SiT on ImageNet with approximately capacity-matched correction heads, sweeping $K = 1$–8. Stars mark each arm's lowest-FID setting.

Timestep conditioning. By default, we provide the true timestep to reuse the pretrained model's timestep-conditioning pathway. However, Tab. 9 shows that using the true timestep or a constant time embedding yields similar final FID in our residual-correction setting. This finding may not generalize to other settings, including correctors trained from scratch: the pretrained pathway associates t with the noise level of the original noisy input, which does not directly describe the errors in the clean-sample prediction received by the corrector.

Corrector time input	FID ↓	Precision ↑	Recall ↑
True timestep	2.825	0.763	0.721
Constant time embedding	2.895	0.745	0.734

Table 9: **Explicit time conditioning has little effect on FID in this setting.** We compare true and fixed timestep inputs to the corrector on SiT-REPA at $K = 1$, using FID-50K at 250 NFE. The denoiser uses the true timestep in both cases.

Prediction-space versus hidden-state refinement. We compare correction in output space (pixels or latents) with refinement of the network's hidden states. Output-space correction benefits from repeated application, whereas the hidden-state variants do not achieve comparable gains (Tab. 10). Increasing the number of refinement iterations during training does not improve sample quality in either setting. Fig. 20 presents qualitative comparisons at $K = 5$.

On the other hand, removing stop-gradient from the base denoiser's prediction achieves a lower best FID across the tested correction counts, but retaining it yields more consistent gains at larger K. Without stop-gradient, samples become visually simpler under repeated correction, as shown in Fig. 20(b).

Stop-gradient	K_{train}	Inference-time iterations K					
		0	1	2	3	4	5
Correcting model predictions $\hat{x}$							
✓	1	24.63	17.96	14.07	13.35	14.27	16.06
✗	1	29.08	17.38	12.88	19.25	24.64	28.92
✓	1–5	24.66	21.02	21.68	22.51	23.30	24.67
Iterating over hidden states h							
✓	1	24.63	24.78	35.73	102.84	261.61	339.93
✗	1	24.19	24.63	39.85	104.94	233.81	326.03
✓	1–5	24.66	24.99	25.01	25.12	25.35	25.47

Table 10: **Output-space iteration enables effective repeated correction.** Component ablations on JiT-B/16, evaluated by unguided FID50K at approximately 250 NFE.

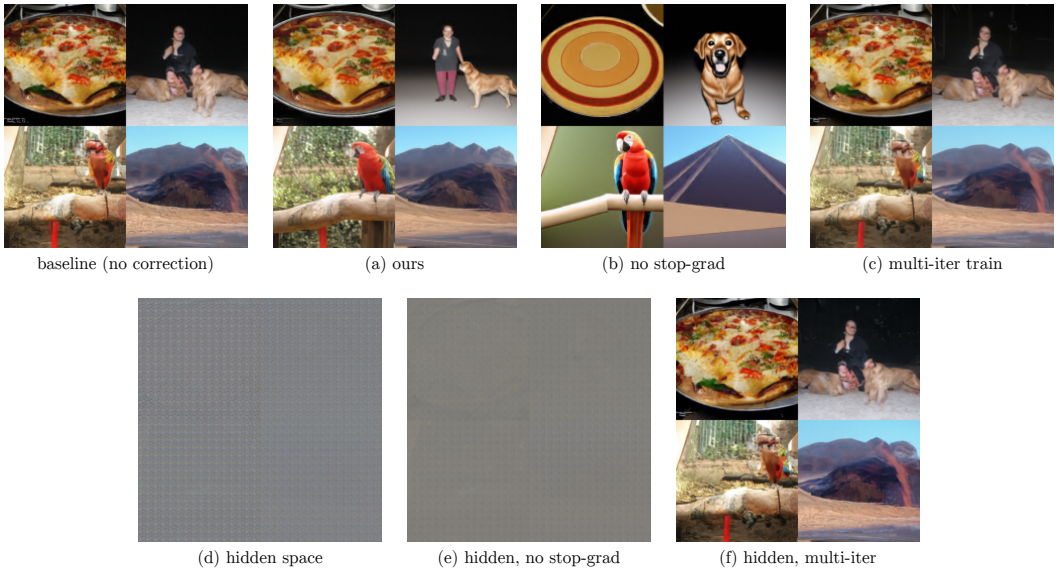


Figure 20: **Output-space correction remains effective under repeated application.** Qualitative comparisons of the component ablations in Tab. 10 at $K = 5$, using the same initial noise and class labels.

B.2 Training configuration

Stop-gradient versus end-to-end optimization. We further evaluate the effect of stop-gradient on additional pretrained models (Fig. 21). Unlike the earlier comparison in Sec. B.2, the variants achieve similar best FID here. Full stop-gradient (our default) nevertheless leads to less degradation at larger correction counts K.

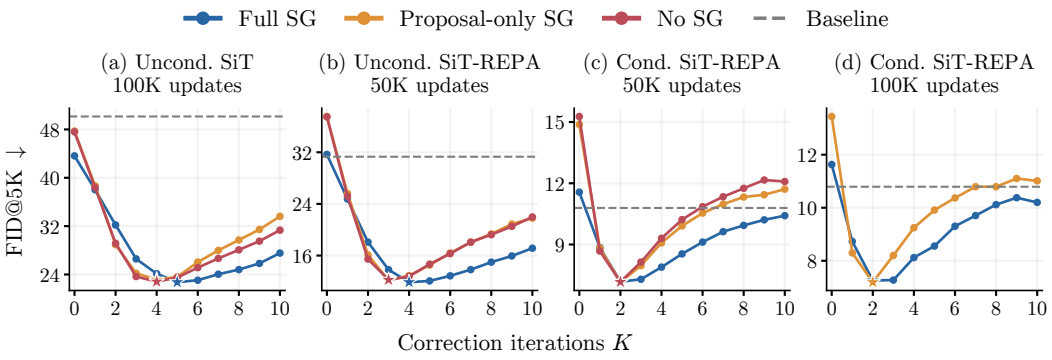


Figure 21: **Full stop-gradient degrades more slowly at large correction counts in these comparisons.** FID@5K versus K, using EMA weights, the full correction interval, and at most 250 NFE. SG denotes stop-gradient; full SG blocks both proposal and base-combination paths.

Continued versus frozen denoiser. We briefly discussed the effect of continued denoiser training in Sec. A.2. Here, we compare freezing the JiT denoiser with jointly training it under the denoising objective. Fig. 22 shows that freezing remains competitive at small K, whereas continued denoiser training yields better FID at larger K.

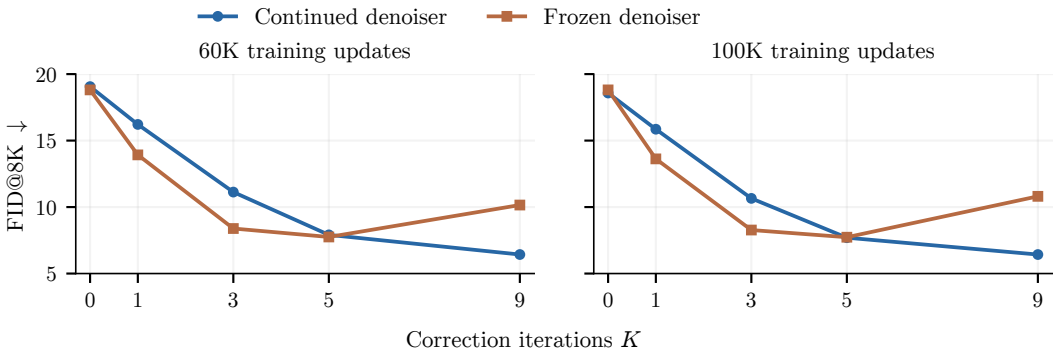


Figure 22: **Frozen-denoiser correction is competitive at small K but deteriorates earlier.** JiT-L/16 FID@8K with a continued or frozen denoiser, using EMA weights, no CFG, early-interval correction, and at most 250 network evaluations.

Qualitative behavior under repeated correction. The frozen-denoiser examples include both successful structural refinement and pronounced color saturation (Fig. 23).

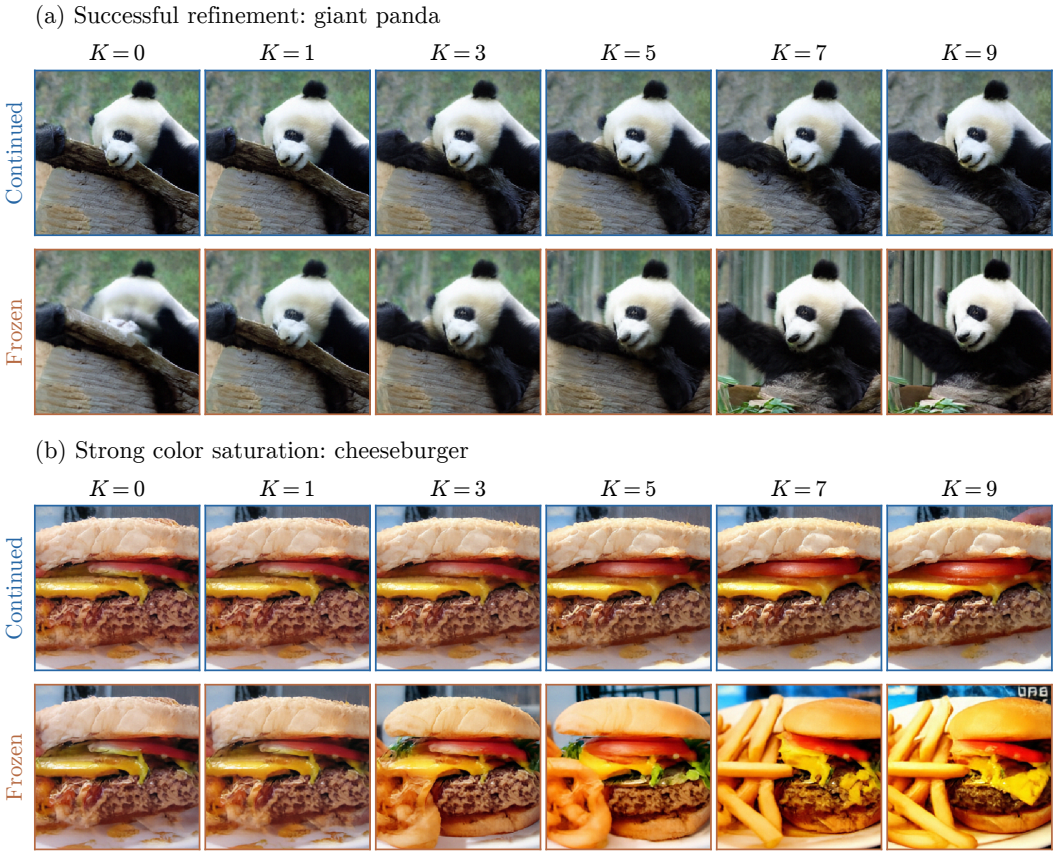


Figure 23: **A frozen denoiser can support successful refinement, but repeated correction can also oversaturate samples.**

Single- versus multi-correction training. Single-correction training transfers effectively to repeated output-space correction and outperforms the tested variable-K_{train} alternative in JiT (Fig. 24).

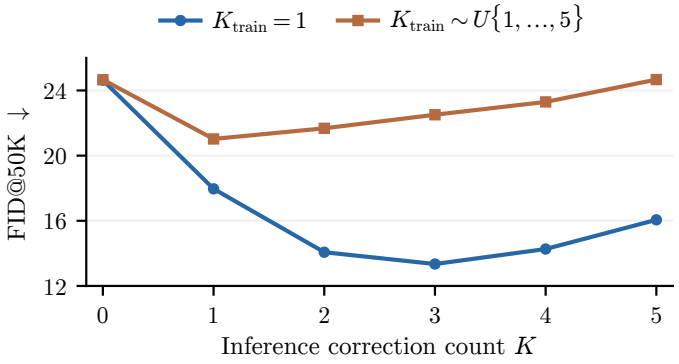


Figure 24: **Single-correction training performs better than the tested multi-correction alternative.** JiT-B/16 with stop-gradient, trained for 100K updates using either one correction or a uniformly sampled count from 1 to 5.

Proposal update rate. Proposal EMA controls how quickly the corrector's inputs change, independently of stop-gradient; Fig. 25 reports finite-budget sensitivity, not converged performance. For pretrained models, the continued/raw and frozen settings compared above both support useful correction: selected examples show less saturation with continued updates, while a fixed denoiser simplifies training and can yield pronounced, qualitatively CFG-like structural refinement (Fig. 23).

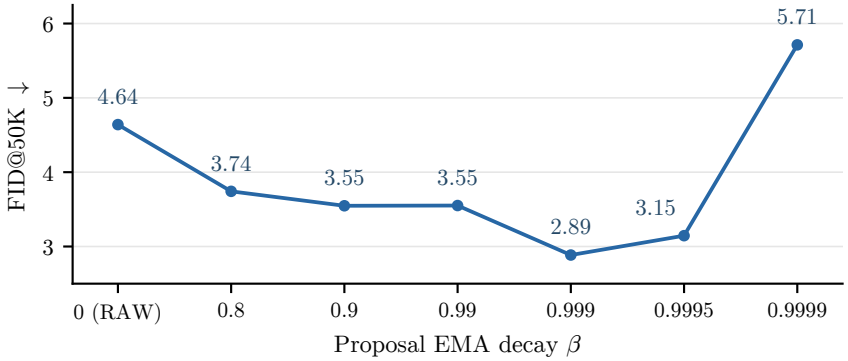


Figure 25: **Intermediate proposal smoothing gives the lowest FID at the evaluated checkpoint.** From-scratch LightningDiT D4+C8 on ImageNet at 200K updates, with evaluation EMA fixed at 0.9999. Each point selects the best FID50K over $K = 1, \ldots, 8$; all select $K = 3$. Decays are equally spaced categories.

B.3 Inference configuration

Correction count K. Increasing K improves FID up to a setting-dependent optimum, after which further correction can degrade it (Tab. 10; Fig. 21).

Correction interval. Restricting JiT correction to early steps and reallocating the saved calls to denoising improves FID at a fixed sampling budget (Tab. 11).

Start r_{start}	End r_{end} 0.8	0.6	0.4	0.2	0.0
1.0	7.70	**6.79**	7.72	9.05	11.85[full]
0.8	—	11.16	11.12	12.59	13.76
0.6	—	—	14.54	15.09	17.94
0.4	—	—	—	17.10	20.05
0.2	—	—	—	—	19.96

[full] Full trajectory ($1 \to 0$).

Table 11: **Allocating correction to early steps yields the lowest FID at a fixed sampling budget.** JiT-L/16 with $K = 5$, evaluated by FID8K. Rows and columns specify the start and end noise levels, decreasing from noise to data.

B.4 Alternative resource and training regimes

Shared backbone. The shared-backbone SiT variant retains similar correction quality with substantially fewer parameters (Tab. 5), although its shared parameters receive gradients from both losses.

From-scratch capacity allocation. A 14-layer denoiser plus a 14-layer corrector reaches similar FID with one correction and lower FID with repeated correction than the tested 28-layer denoiser baselines at matched block-evaluation budget (Tab. 12), a different question from the pretrained expansion controls in Appendix A.1.

Model	Params. (M)	Sampling	Steps	Transformer Block Evaluations	FID ↓
D_{28}	675.3	Unguided	240	6,720	8.93
D_{28}	675.3	CFG ($w = 5$)	120	6,720	6.17
D_{14} + C_{14}	681.1	Correction ($K = 1$)	240	6,720	8.74
D_{14} + C_{14}	681.1	Correction ($K = 4$)	96	6,720	**5.56**

Table 12: **Repeated correction outperforms deeper denoising and the tested CFG scales at a matched transformer-block budget.** Models are trained from scratch for 80K updates on ImageNet-256; FID uses 8,000 samples. CFG and correction use the full sampling interval. We report the best CFG scale from $w \in \{1, \ldots, 10\}$.

C Implementation Details

C.1 Training configurations

Common backbone settings. We retain each pretrained model's backbone and original conditioning. The separate corrector uses an independent copy of the denoiser backbone; latent-space models retain their original VAE, while JiT operates directly on RGB images. Training and quantitative sampling settings are summarized in Tables 13 and 14.

C.2 Corrector parameterization

Algorithm 1 uses x-prediction. For an ϵ- or v-predicting denoiser, we first convert its output to a clean-sample proposal $\hat{\boldsymbol{x}}^{(0)}$ using the model's noise schedule, following the x-space input choice discussed in Sec. 4. The corrector can then predict x, ϵ, or v. In our main pretrained-model experiments, we preserve the pretrained output parameterization, including ϵ- and v-prediction heads.

Table 13: **Iterative refinement uses correctors trained with a single correction pass.** Training settings for the main ImageNet and quantitative text-to-image comparisons. We follow each model's original training recipe as closely as possible. Updates count additional training, not pretraining; D and C denote separate denoiser and corrector networks. Sampling settings are reported separately.

Setting	DiT-XL/2	SiT-XL/2	SiT$_{\text{REPA}}$-XL/2	JiT-L/16	T2I (DiT + DDT)
Data and initialization					
Training data	ImageNet	ImageNet	ImageNet	ImageNet	BLIP3o mixture
Image resolution			256×256		
Denoising space	SD-VAE latents	SD-VAE latents	SD-VAE latents	RGB pixels	FLUX.2-VAE latents
Denoiser init.	Released	Released	Released	Released	202K base
Corrector init.	Independent copy of the pretrained denoiser				
Additional updates	100K	100K	50K	100K	50K
Optimization					
Global batch size	256	256	256	1,024	1,024
Optimizer	AdamW; weight decay 0				
Adam (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.95)$	$(0.9, 0.95)$
Denoiser LR	10^{-4}	10^{-4}	10^{-4}	5×10^{-5}	Linear decay
Corrector LR	10^{-4}	10^{-4}	10^{-4}	5×10^{-5}	10^{-4}
Warmup	None	None	100 updates	5 epochs	No new warmup
LR after warmup	Constant	Constant	Constant	Constant	D: linear; C: const.
EMA decay	0.9999	0.9999	0.9999	0.9999/0.9996	0.9995
Training objective and correction					
Prediction (D/C)	ϵ/ϵ	v/v	v/v	x/x	v/v
Corrector image input	Detached clean-sample prediction; original timestep and conditioning retained				
Output combination	Gated ϵ	Gated v	Gated v	Direct x	Gated v
Gate initialization	0	0	0	—	0
Correction passes	$K_{\text{train}} = 1$				
Loss weights $(D : C)$	$1 : 1$; correction loss has no gradient path into the separate denoiser				

Noise and velocity contain a noise-dependent component that a clean sample alone does not determine. Rather than requiring the corrector to reconstruct this component without access to $\boldsymbol{z}_t$, we retain the base prediction through a learned linear blend. Let $\boldsymbol{q}$ denote the target in the chosen output space and $\hat{\boldsymbol{q}}^{(0)}$ the base prediction. Our full stop-gradient formulation is

$$\hat{\boldsymbol{q}}^{(1)} = (1 - \boldsymbol{\gamma}) \odot \mathrm{sg}(\hat{\boldsymbol{q}}^{(0)}) + \boldsymbol{\gamma} \odot C^q_{\boldsymbol{\phi}}\Big(\mathrm{sg}(\hat{\boldsymbol{x}}^{(0)}), t\Big), \tag{8}$$

$$\mathcal{L}_{\text{corr}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t}\left[\lambda_q(t)\left\|\hat{\boldsymbol{q}}^{(1)} - \boldsymbol{q}\right\|_2^2\right], \tag{9}$$

where $C^q_{\boldsymbol{\phi}}$ outputs in q-space, $\odot$ is elementwise multiplication, and $\lambda_q(t)$ retains the model's loss weighting. The gate $\boldsymbol{\gamma}$ is learned per channel and initialized to zero. DiT and plain SiT use a scalar shared across channels. Direct x-prediction is also a viable, simpler alternative (App. A.1). We retain the gated form for compatibility with pretrained models and its slightly better empirical performance, although our framework does not require this parameterization.

Gradient paths. Full stop-gradient (SG) uses both stop-gradients in Eq. 8. Proposal-only SG removes the stop-gradient on $\hat{\boldsymbol{q}}^{(0)}$. No SG removes both. As discussed in App. B.2, the preferred path depends on K. We use Full SG by default: it avoids backpropagation through the proposal denoiser, achieves competitive FID, and is more robust to large correction counts in our comparisons.

Table 14: **The main quantitative comparisons use approximately 250 network evaluations per image.** Sampling settings for Tables 1 and 3. K is the number of corrections per active step; intervals use noise time t (1: noise, 0: data). NFE includes correction and both CFG branches. Guided T2I applies correction before CFG.

Method	CFG w	K	CFG interval	Correction interval	Steps	NFE
DiT-XL/2; DDPM						
Time steps: uniformly respaced discrete indices, $999 \to 0$						
Baseline	1	0	—	—	250	250
CFG	1.5	0	$1 \to 0$	—	125	250
Correction	1	2	—	$1 \to 0$	83	249
SiT-XL/2; Euler–Maruyama SDE						
Time steps: linear $1 \to 0.04$, followed by a final mean step to 0						
Baseline	1	0	—	—	250	250
CFG	1.5	0	$1 \to 0$	—	125	250
Correction	1	2	—	$1 \to 0$	83	249
SiT$_{\mathrm{REPA}}$-XL/2; Euler–Maruyama SDE						
Time steps: linear $1 \to 0.04$, followed by a final mean step to 0						
Baseline	1	0	—	—	250	250
CFG	1.8	0	$0.7 \to 0$	—	148	250
Correction	1	5	—	$0.7 \to 0.5$	123	248
JiT-L/16; Heun ODE (final Euler step)						
Time steps: linear $1 \to 0$						
Baseline	1	0	—	—	125	249
CFG	2.4	0	$0 < t < 0.9$	—	63	250
Correction	1	9	—	$1 \to 0.8$	45	242
T2I (DiT + DDT); Euler ODE						
Time steps: $t = su/[1 + (s-1)u]$, $s = \sqrt{8}$, with u linear $1 \to 0$						
Baseline	1	0	—	—	250	250
Correction	1	6	—	$1 \to 0$	35	245
CFG	4	0	$1 \to 0$	—	125	250
CFG + Correction	4	2	$1 \to 0$	$1 \to 0$	41	246

Table 15: **Unconditional ImageNet settings.** Sampling settings for Table 2, using SiT$_{\mathrm{REPA}}$-XL/2.

Method	**Setting**	**Train steps**	**Denoising steps**	**NFE**
Baseline	–	50K	250	250
Correction	$K = 1$	50K	125	250
Correction	$K = 4$	50K	50	250
Autoguidance	$w = 7$	50K	125	250

C.3 Figure and Table Details

Unconditional generation (Table 2; Figure 8). We evaluate SiT$_{\mathrm{REPA}}$-XL/2 with the null class after 50K updates, using EMA weights, 50,000 images, and the native Euler–Maruyama sampler. Correction detaches both the proposal and the base prediction. We sweep $K = 0, \ldots, 10$ using $\lfloor 250/(K+1) \rfloor$ denoising steps, and autoguidance $w = 1, \ldots, 10$ using the released pretrained guide at 250 NFE. The table reports the lowest-FID settings from these sweeps, in addition to $K = 1$. Sampling settings are summarized in Table 15.

Text-to-image examples (Figure 9). We compare $K = 0$ and $K = 6$ using the same 50K-update EMA checkpoint, without CFG, and 50 Euler steps (50 and 350 NFE). Paired images share their prompt and recorded rank seed. Full prompts are listed in Table 17.

With CFG ($w = 2$)			*With CFG* ($w = 6$)		
Method	GenEval ↑	DPG-Bench ↑	Method	GenEval ↑	DPG-Bench ↑
Baseline	54.57	80.11	Baseline	71.70	84.44
Correction ($K = 2$)	61.17	80.82	**Correction** ($K = 2$)	**72.65**	**84.55**
Correction ($K = 4$)	63.61	**81.11**	**Correction** ($K = 4$)	71.52	84.38
Correction ($K = 6$)	**63.74**	80.79	**Correction** ($K = 6$)	69.53	83.42

Table 16: **Correction's alignment gains depend on guidance strength and iteration count.** Additional guidance scales use the same checkpoint as Table 3, but fix 50 Euler steps rather than NFE. Correction is applied before CFG. Bold marks the best score within each block.

Guidance and correction (Figure 10). We compare $w \in \{1, 6\}$ and $K \in \{0, 6\}$ at 50 Euler steps. The selected prompt is *a photo of a bench*, with all eight seeds retained per setting. CFG is applied before correction.

Parameter sharing (Table 5). We compare plain SiT-XL/2 models after 100K updates, without CFG, using FID50K. The shared model uses separate input embedders and shares the transformer blocks and output head. Both separate and shared correctors detach the proposal and base prediction, but shared parameters receive gradients from both losses. The baseline is a separate denoising-only continuation. Baseline, separate-corrector, and shared-corrector settings use $(K, \text{steps}) = (0, 250), (2, 83), (3, 62)$, respectively.

Distillation (Table 6). We freeze the separate SiT-XL/2 teacher at EMA100K and initialize the student from its denoiser. The student minimizes velocity MSE against the teacher's $K = 2$ prediction, without a ground-truth loss, for 100K updates on class-conditional ImageNet. We sample Gaussian noise and uniform time, using batch size 256, AdamW with learning rate 10^{-4}, 100 warmup updates, zero weight decay, and EMA decay 0.9999. We evaluate the final EMA with FID@50K using the native SiT Euler–Maruyama sampler and ADM evaluator.

Power spectra (Figure 11). We average radial power over 100 ImageNet-256 samples per setting for $K = 0, \ldots, 10$ at approximately 250 NFE. The panels show $\log_{10}$ power and $\log_{10}(P_{\text{sample}}/P_{\text{real}})$ over normalized frequencies 0.1–0.5.

D FULL PROMPTS FOR TEXT-TO-IMAGE EXAMPLES

Table 17: **Full prompts for the text-to-image examples in Figure 9.**

Figure column	Full prompt
1. Green guitar	In the center of a dimly lit stage, a vintage green guitar with a glossy finish leans gently against a classic black amplifier. The polished wooden flooring of the stage reflects the warm, golden hue from a single spotlight above, which casts an inviting glow over the instrument. A microphone stand with a chrome finish stands to the left of the guitar, poised for the evening's performance.
2. Vintage car	a vintage blue Porsche 356 is captured mid-turn on a winding asphalt road, its polished surface reflecting the bright sunlight. The car's classic design is accentuated by its rounded headlights and sleek bodywork. Alongside the road, a low stone wall partially covered in moss provides a contrast to the well-maintained vehicle.
3. Sunlit kitchen	A spacious kitchen with a large wooden table at the center, cluttered with an assortment of pots, pans, and cooking utensils. The table is surrounded by matching wooden chairs, and the walls are lined with shelves filled with spices and kitchenware. Sunlight streams in through a window, casting a warm glow on the array of cookware.
4. Three violins	Three exquisitely crafted violins, with their elegant F-shaped sound holes and smooth, glossy wooden surfaces, rest gently beside a grand piano. The piano's polished ebony finish reflects the soft, gentle curves of the violins that lie on a velvet-lined case. Behind them, the piano's open lid reveals its intricate strings and hammers, inviting a symphony of sounds to be played.
5. Rabbit and seashell	On the soft, warm sand of the beach, a fluffy white rabbit with rounded ears is caught in a curious moment, gently placing its paw on the ribbed surface of a pink scallop shell. The scallop, slightly open, reveals its smooth interior contrasting with its coarse outer texture, while hues of pink and orange from the setting sun reflect off its surface. There's a tranquil ocean backdrop with the gentle ebbing of the tide, and the fading daylight casts a golden glow over the scene, highlighting the rabbit's soft fur and the shell's subtle color.
6. Victorian library	A voluminous library, bathed in soft, muted blue hues, stretches majestically across a spacious Victorian-style living room. Intricately carved wooden bookshelves, filled with an array of books of various sizes and colors, line the walls and a grand fireplace sits as the room's focal point. The vray render showcases hyperrealistic textures, from the plush, patterned rug beneath the polished wooden tables, to the delicate fabric of the antique armchairs scattered throughout the space.

E CORRECTION VS. CLASSIFIER-FREE GUIDANCE. COMPARISON AND ITS SYNERGY

E.1 COMPARING CORRECTION AND CFG

Fig. 26 compares correction and CFG on ImageNet using the same checkpoint and initial noise. In these selected examples, increasing guidance produces more similar close-up compositions and stronger colors, whereas correction retains more varied poses and backgrounds. These observations concern the displayed samples; they do not by themselves establish distributional mode collapse.

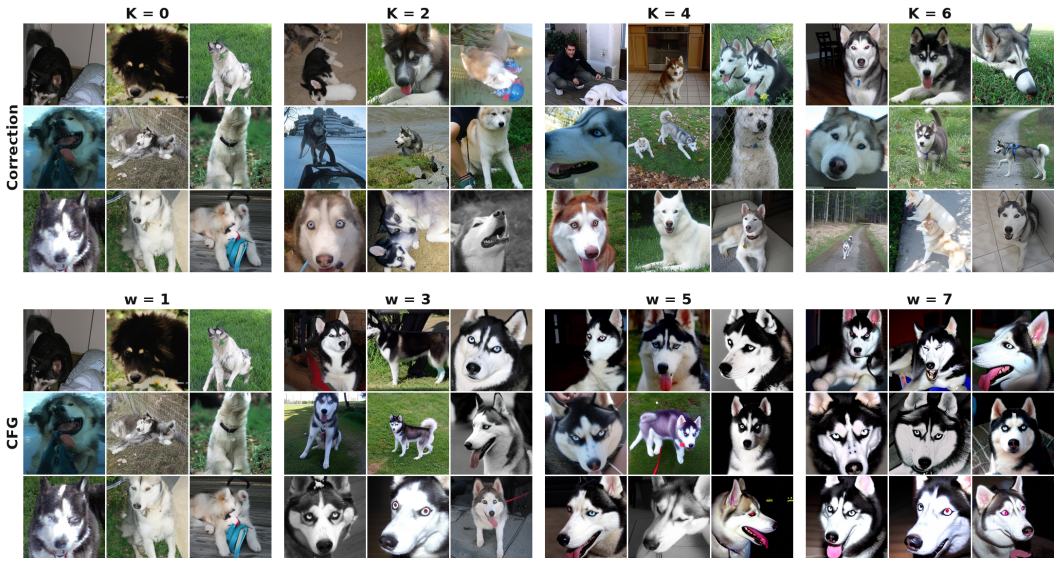


Figure 26: **Correction and guidance produce different visual trade-offs.** Matched-noise samples from SiT_{REPA}-XL/2 as correction depth K (top) or guidance scale w (bottom) increases, at approximately matched NFE.

Fig. 27 quantifies the trade-offs on DiT. At approximately matched sampling compute, correction reaches a lower minimum FID than the best evaluated CFG setting. The precision–recall trajectories also show the limit of this comparison: CFG reaches higher precision at lower recall, so correction does not establish a dominating frontier.

Following Sadat et al. (2024), we also measure mean HSV saturation against a real-image reference. Stronger CFG raises saturation above the reference, whereas correction remains near or below the unguided level. This ImageNet measurement complements the text-to-image examples; the latter illustrate refinement at weaker guidance rather than measuring a reduction in saturation caused by correction.

E.2 COMBINING CFG AND CORRECTION

Combination order. CFG and correction can be applied in either order. Applying CFG first combines the conditional and null predictions before the corrector refines the guided prediction using the conditional text. Applying correction first refines both branches independently and then combines them with CFG. We use CFG $\rightarrow$ correction for the guided qualitative examples and the main human-preference comparison because we prefer its visual appearance, while correction $\rightarrow$ CFG is used for the guided alignment results in Tab. 3.

Table 18 compares quality and diversity preferences at matching w, K across the two orders. Among the evaluated guided settings, correction before CFG more often yields higher quality preference, while correction after CFG more often yields higher diversity preference. Thus neither order consistently leads on both criteria.

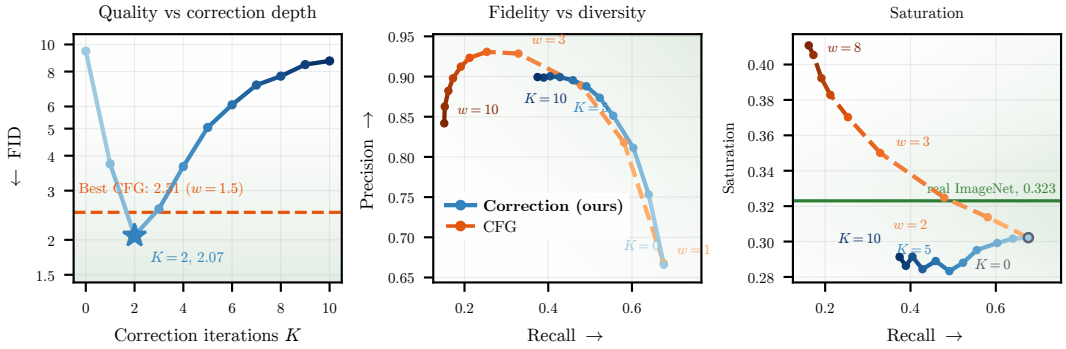


Figure 27: **Correction improves FID with a different precision–recall and saturation trade-off.** DiT-XL/2 at approximately matched NFE. The dashed orange line marks the best evaluated CFG FID; the green line marks real-image saturation.

Table 18: **Neither correction order consistently leads in both quality and diversity.** Full sweep of candidate vote shares (%) against released CFG ($w = 6$, $K = 0$); higher is better. Bold marks the setting reported in Table 4. Each entry contains 800 responses across 80 prompts. The two orders were rated separately against the baseline.

		CFG → Correction		Correction → CFG	
w	K	Quality	Diversity	Quality	Diversity
1	2	33.38	55.38	37.25	54.50
1	4	49.38	57.25	47.50	59.38
1	6	55.75	55.75	52.38	60.75
1	8	54.13	56.38	56.75	63.88
2	2	48.50	60.50	55.38	59.13
2	4	55.25	59.25	65.00	59.00
2	6	57.25	58.75	64.25	61.50
2	8	60.25	59.38	65.75	59.50
4	2	54.63	61.25	61.63	52.25
4	4	58.25	59.50	67.00	57.50
4	6	61.75	55.38	66.88	55.63
4	8	**60.75**	**60.25**	60.50	56.63
6	2	57.13	54.38	59.50	49.63
6	4	59.63	52.75	61.00	53.75
6	6	57.38	58.38	63.63	54.00
6	8	58.63	58.38	59.75	58.13

E.3 HUMAN EVALUATION

Table 4 reports unweighted preference votes on all 80 GenEval single-object prompts, with 10 judgments per prompt (800 per metric). Quality uses one image per method and diversity uses four-image sets. We first applying CFG and then correcting the guided prediction using the conditional text. Both methods use 50 Euler steps.

Table 18 provides the full results for both correction orders over $w \in \{1, 2, 4, 6\}$ and $K \in \{2, 4, 6, 8\}$, highlighting the setting reported in the main text. Each setting is compared separately against the same released baseline. At $w = 1$, the two orders use identical images, so differences reflect repeated human ratings rather than an order effect. Figure 28 illustrates the questions and stimuli for the main table's CFG-then-correction setting ($w = 4$, $K = 8$).



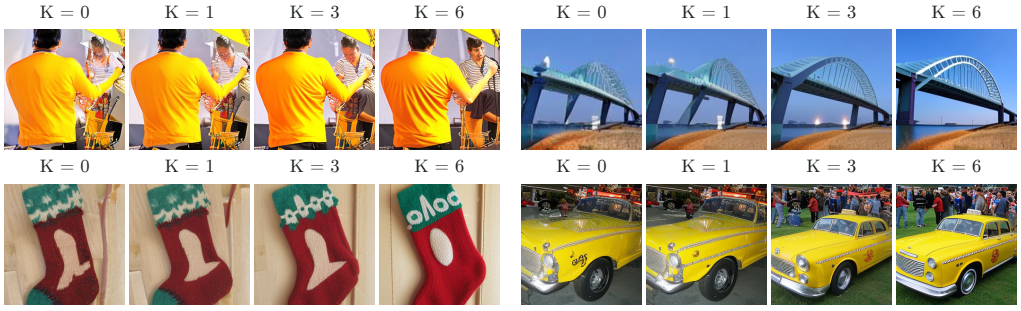
Figure 28: **Quality and diversity are assessed separately.** Actual teddy-bear stimuli and questions for Table 4, using CFG followed by correction ($w = 4$, $K = 8$): single images for quality and four-image sets for diversity. Method labels are added for clarity; the platform randomizes display order.

F More Samples

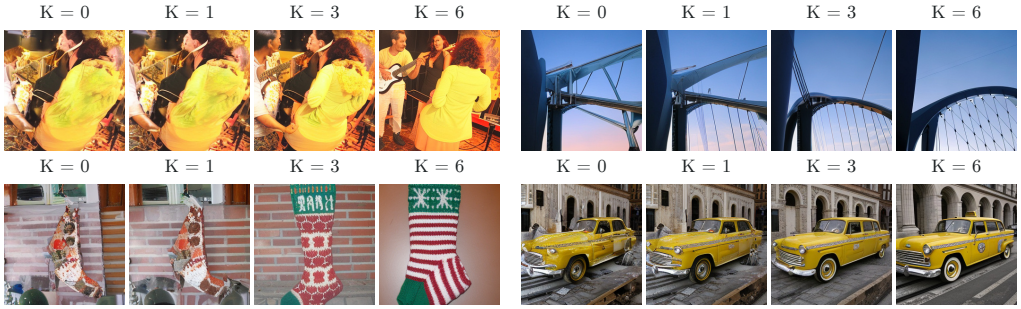
F.1 Effect of Correction Depth

We show the same class and initial noise as correction depth increases, followed by paired galleries over randomly drawn ImageNet classes. All examples use class conditioning without CFG.

SiT-XL/2



SiT-REPA-XL/2



JiT-L/16

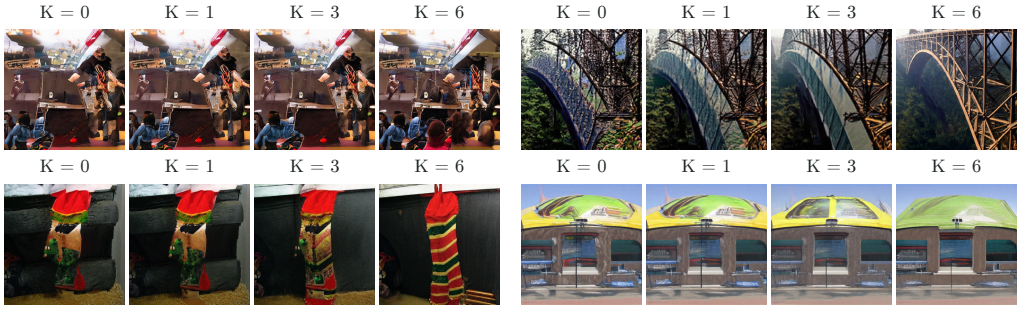


Figure 29: **Correction depth changes both structure and detail.** Each strip follows one sample as K increases, with the initial noise and denoising grid fixed.

F.2 SiT: Random-Class Generation

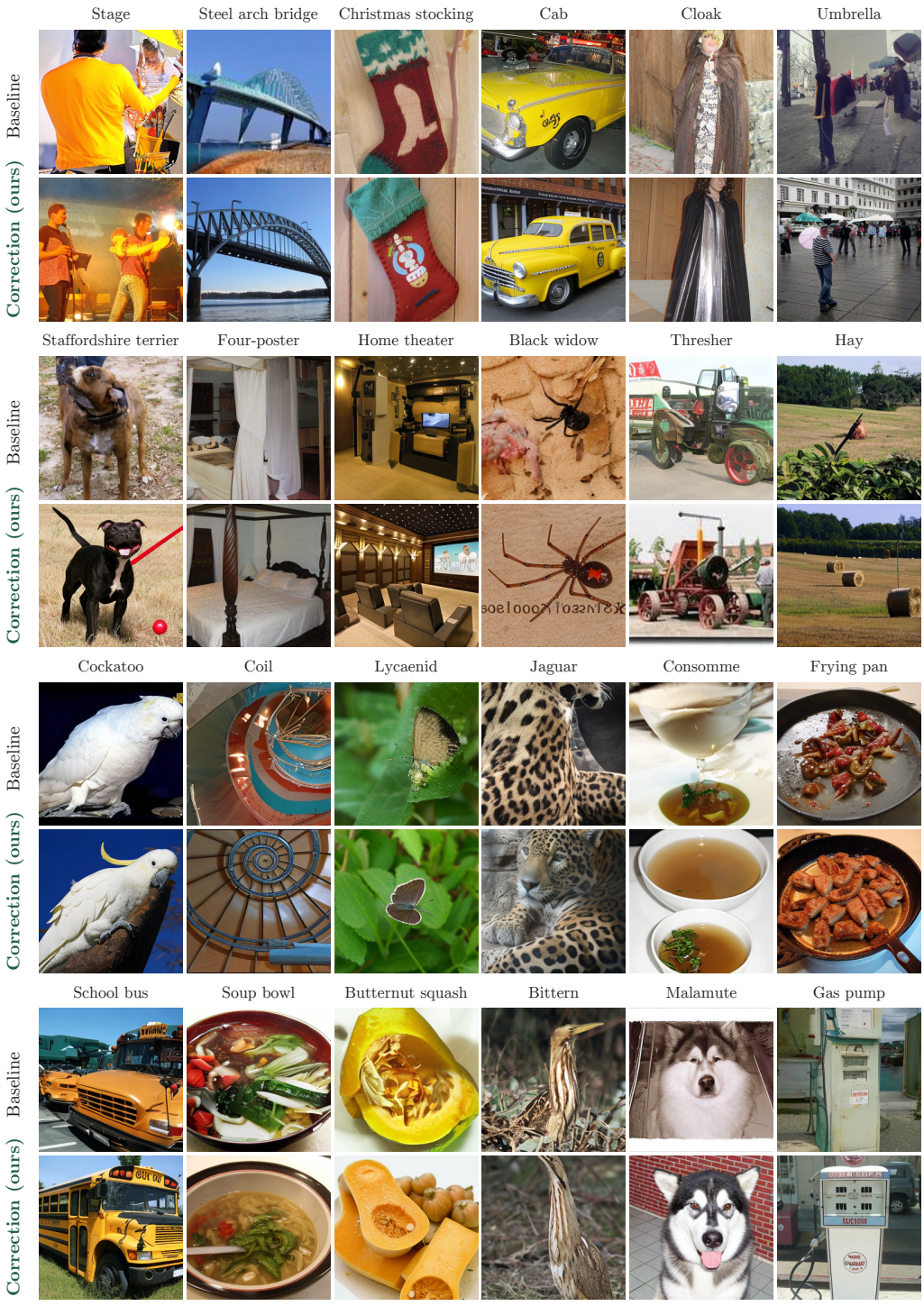


Figure 30: **Correction refines unguided SiT samples across random classes.** Within each pair of rows, baseline ($K = 0$) and correction ($K = 6$) share the same class and initial noise at approximately matched NFEs.

F.3 SiT$_{\text{REPA}}$: Random-Class Generation

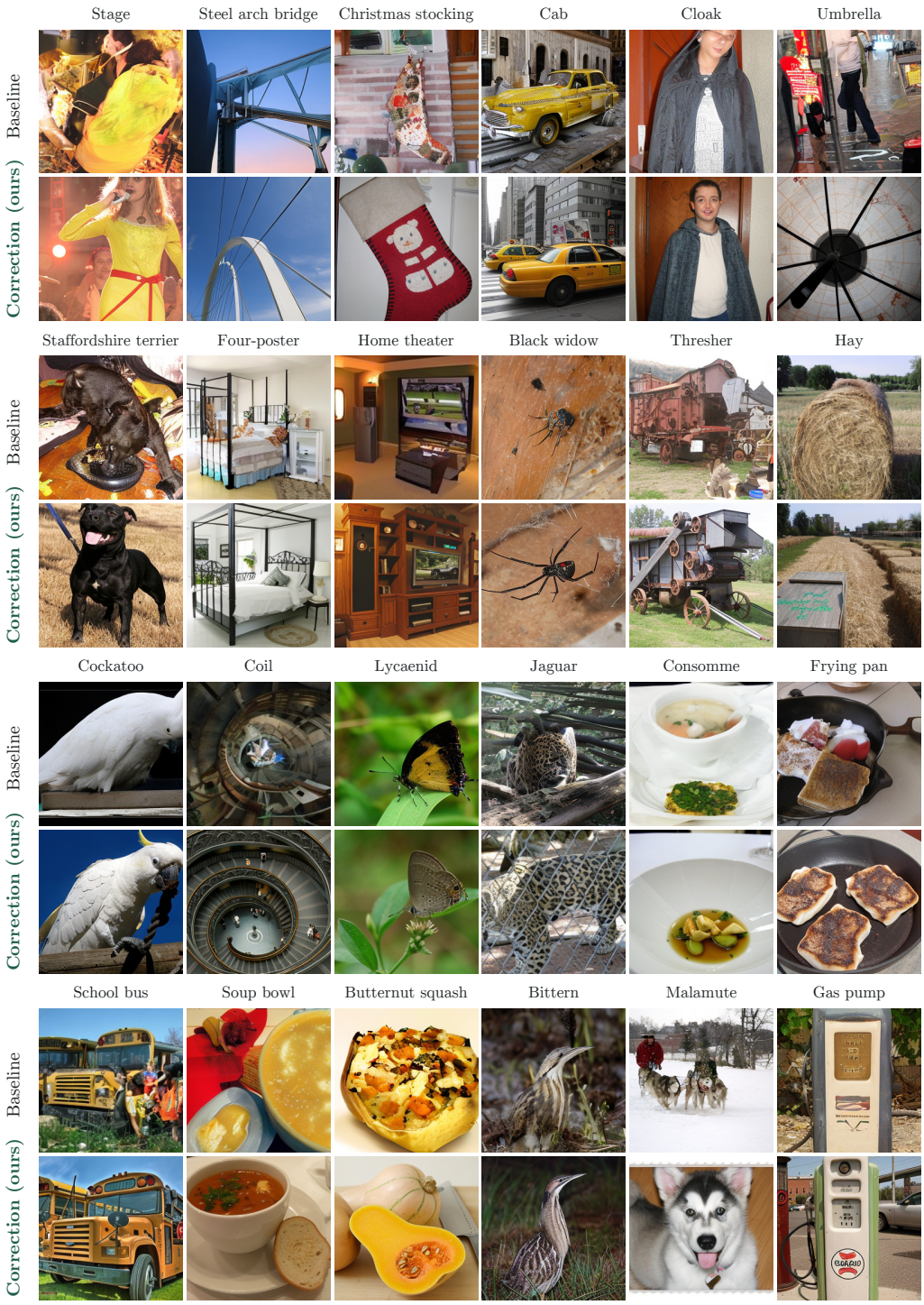


Figure 31: **Correction refines unguided SiT$_{\text{REPA}}$ samples across random classes.** Within each pair of rows, baseline ($K = 0$) and correction ($K = 6$) share the same class and initial noise at approximately matched NFEs.

F.4 JiT: Random-Class Generation

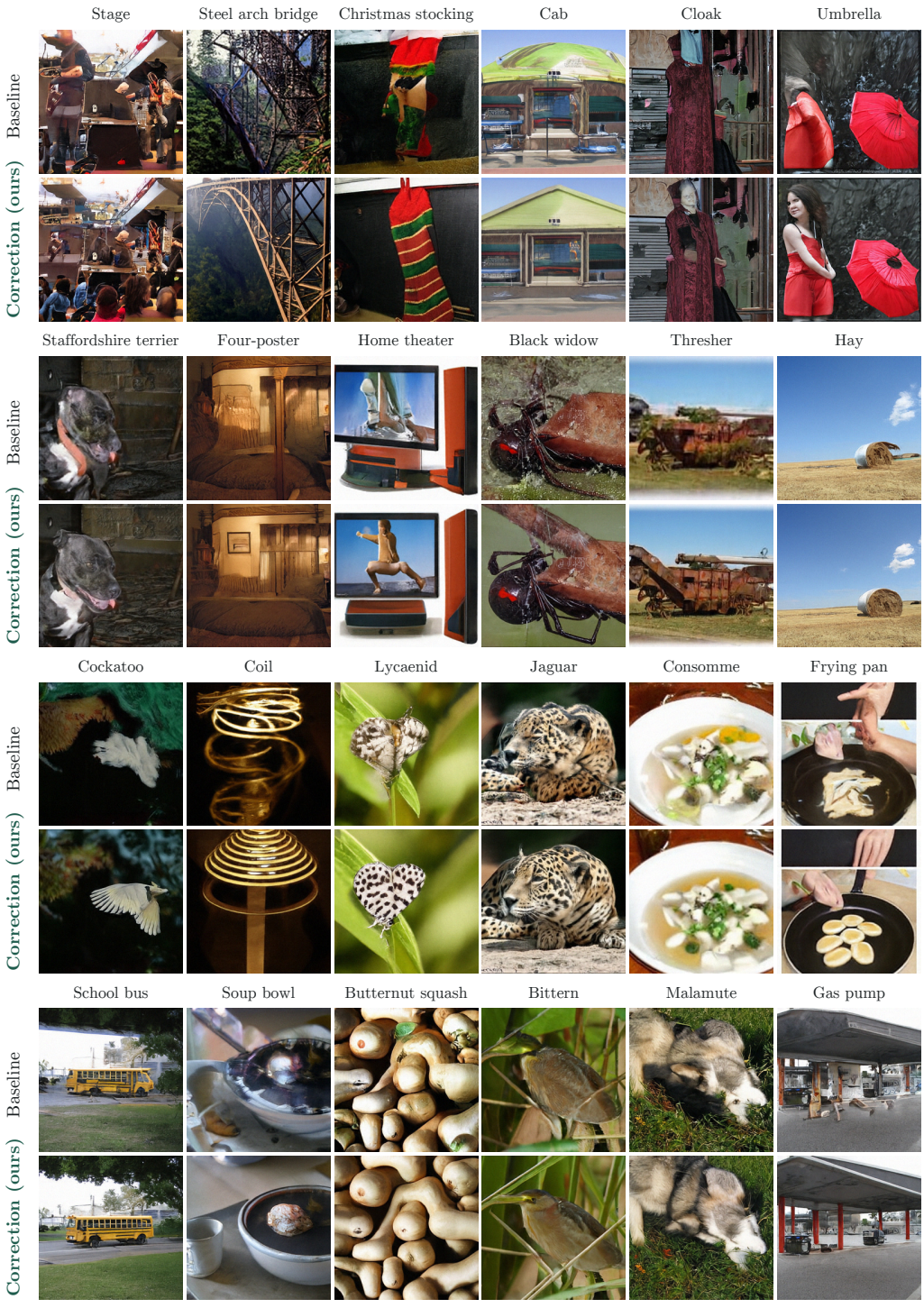


Figure 32: **Early correction changes structure as well as appearance in JiT.** Within each pair of rows, baseline ($K = 0$) and correction ($K = 6$) share the same class and initial noise at matched NFEs.

F.5 UNCONDITIONAL GENERATION

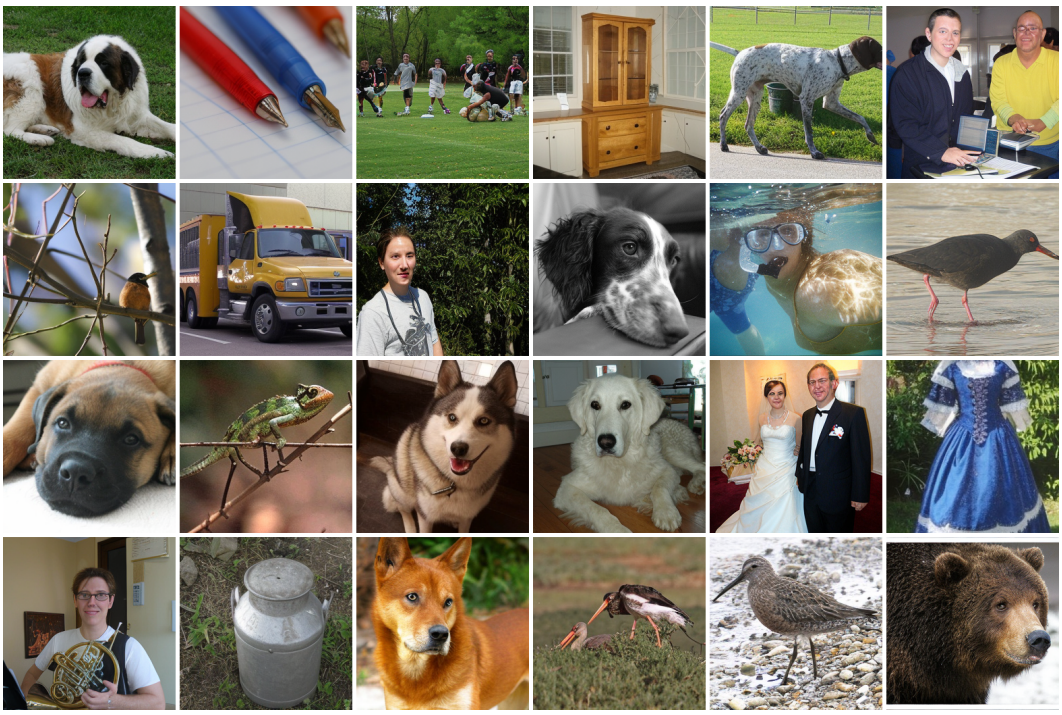


Figure 33: **High-quality unconditional generation without guidance.** Selected ImageNet samples from SiT_{REPA}-XL/2 with iterative correction.

F.6 Unconditional Generation from a Text-to-Image Model

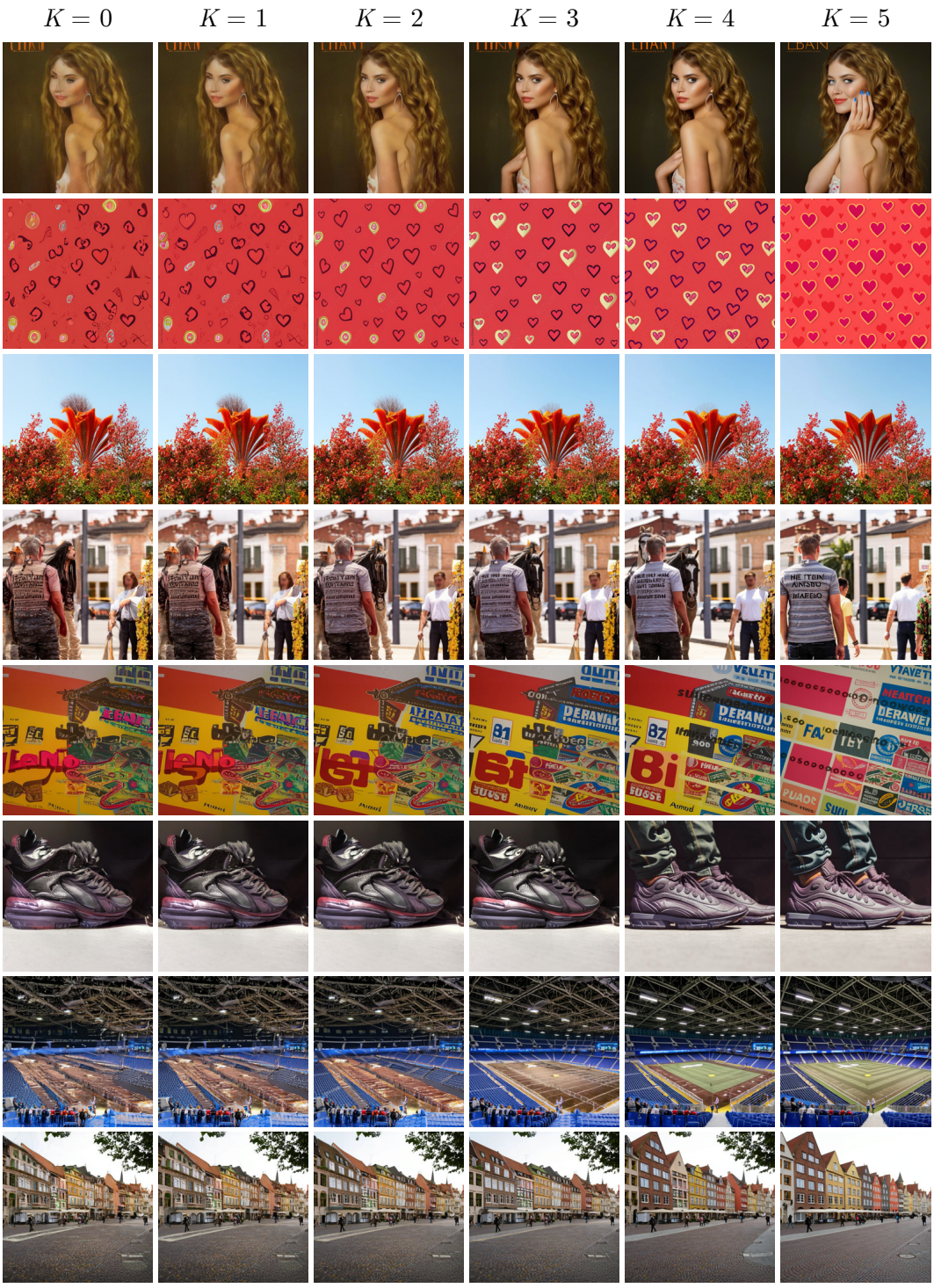


Figure 34: **Unconditional generation supports iterative correction without text or CFG.** Each row shares its initial noise across $K = 0, \ldots, 5$. The checkpoint and denoising steps are fixed; larger K adds correction compute. All seeds are shown without selection.

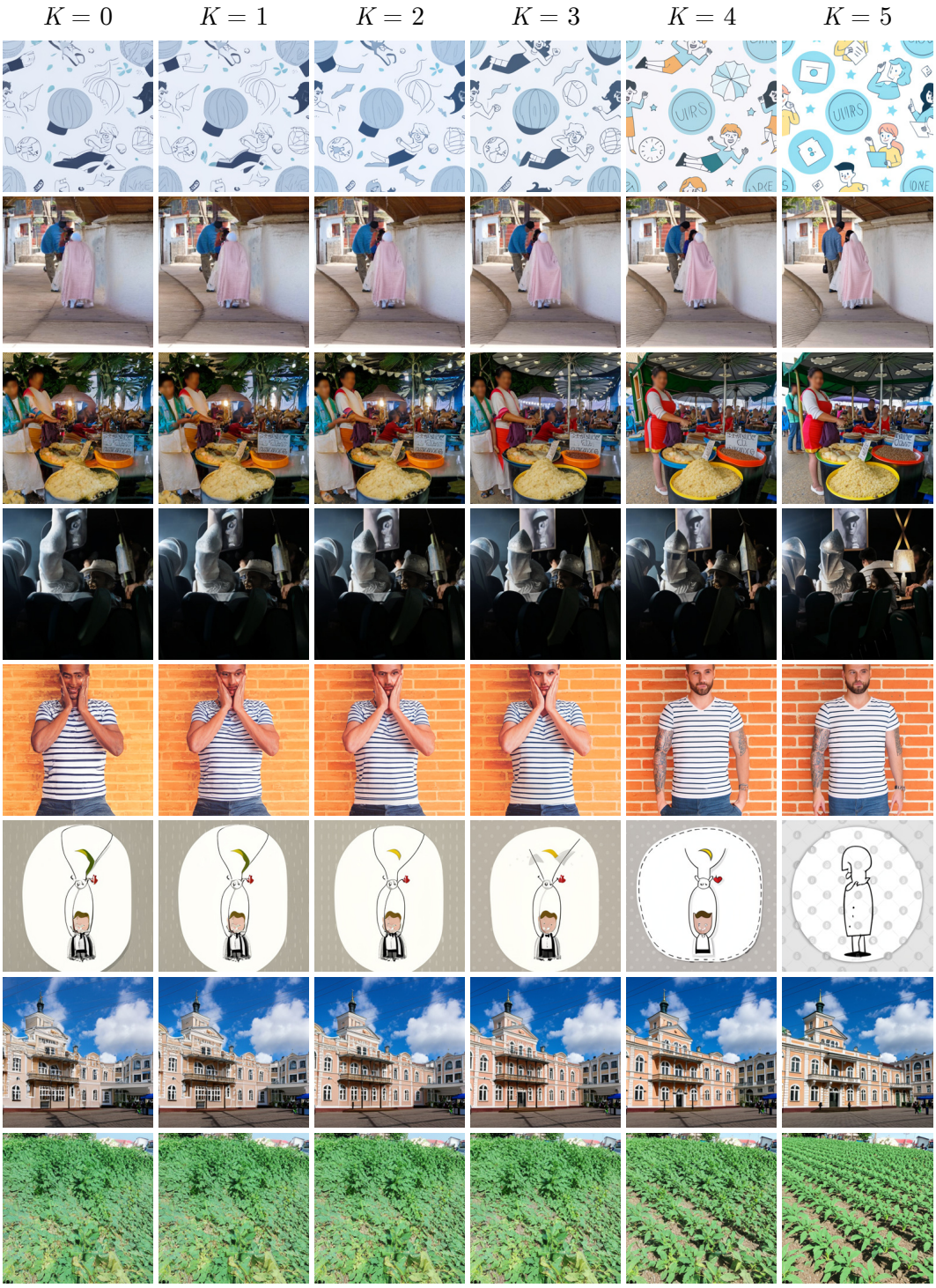


Figure 35: **Unconditional generation supports iterative correction without text or CFG.** Each row shares its initial noise across $K = 0, \ldots, 5$. The checkpoint and denoising steps are fixed; larger K adds correction compute. All seeds are shown without selection.

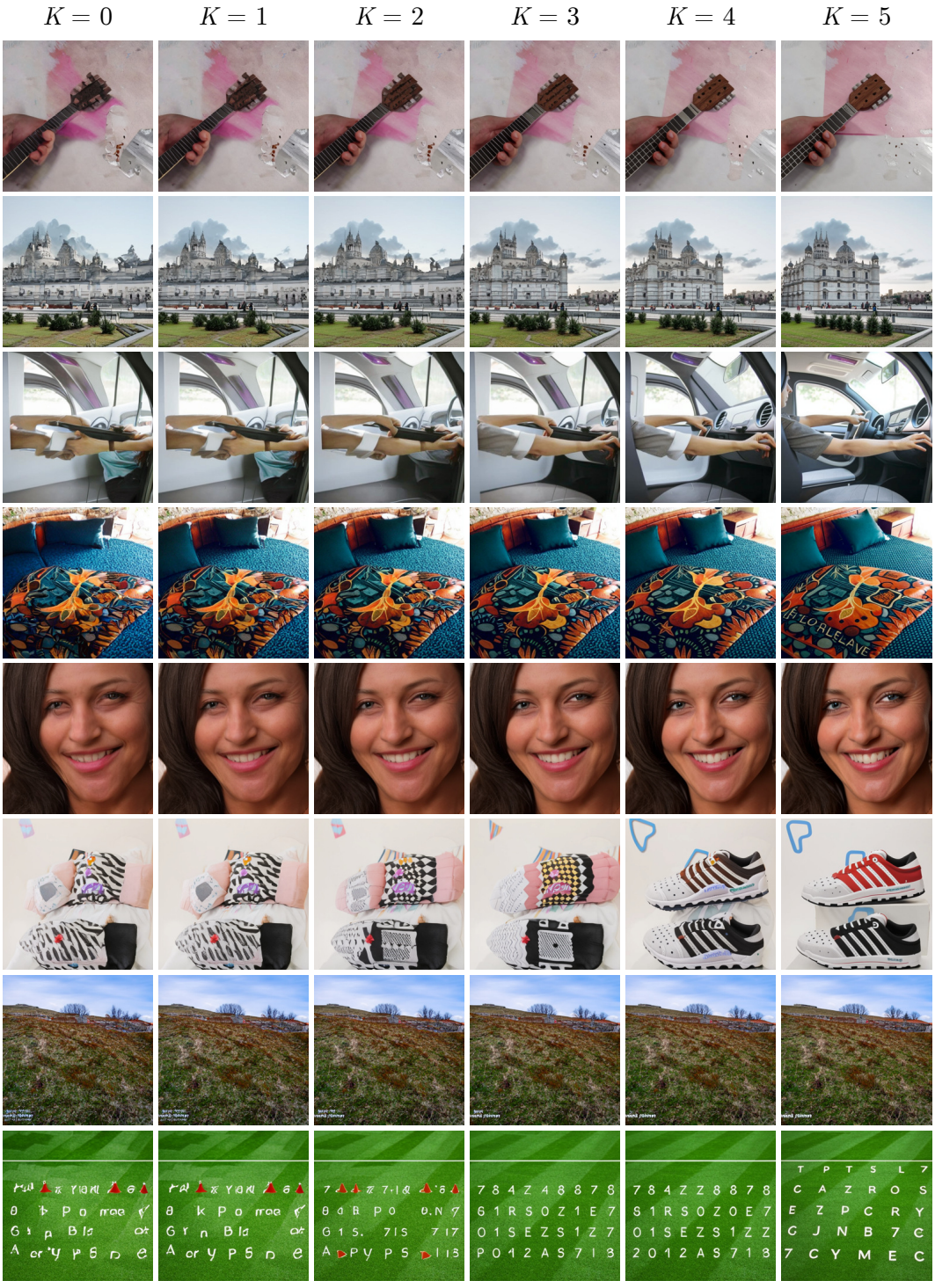


Figure 36: **Unconditional generation supports iterative correction without text or CFG.** Each row shares its initial noise across $K = 0, \ldots, 5$. The checkpoint and denoising steps are fixed; larger K adds correction compute. All seeds are shown without selection.

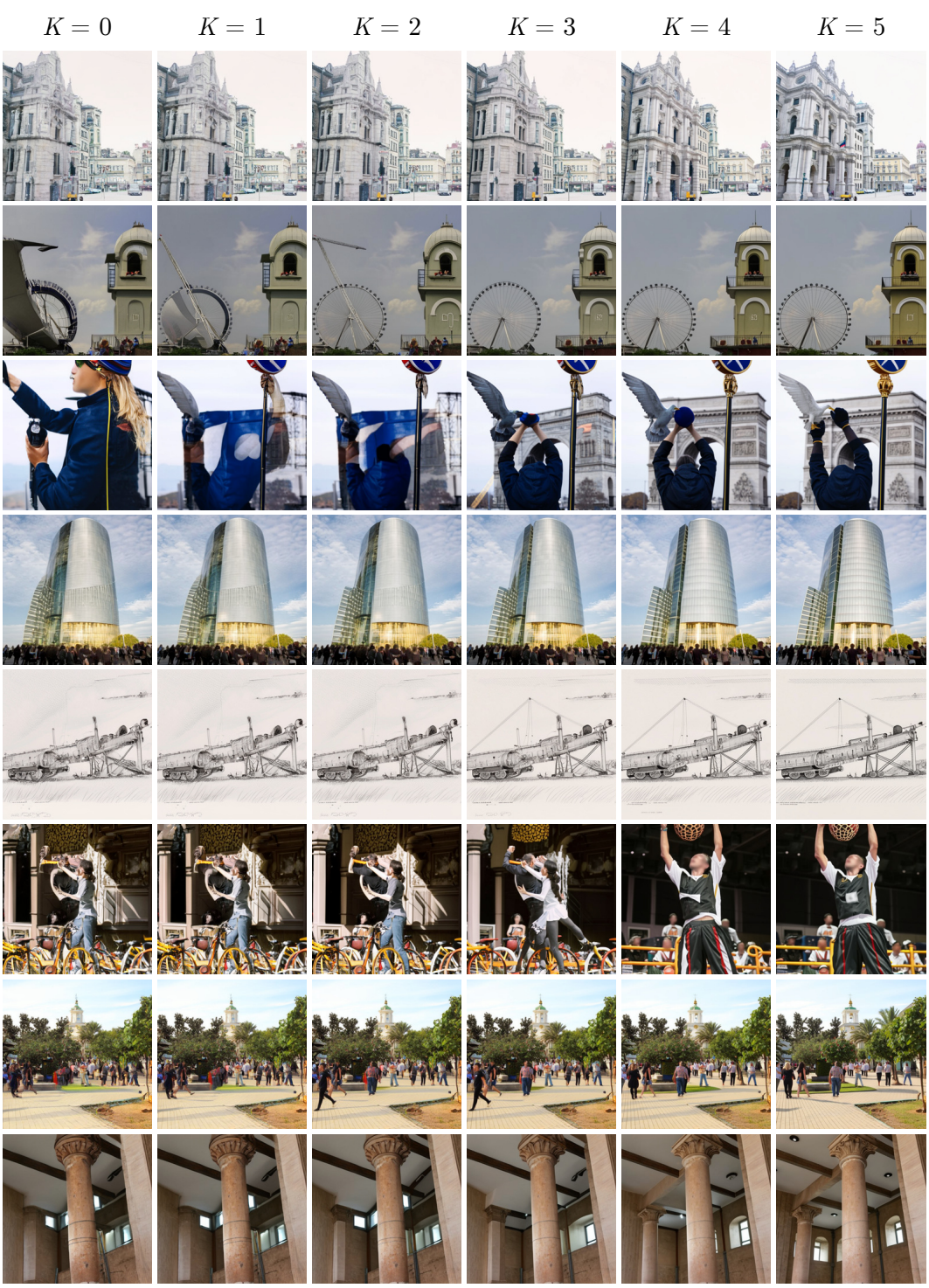


Figure 37: **Unconditional generation supports iterative correction without text or CFG.** Each row shares its initial noise across $K = 0, \ldots, 5$. The checkpoint and denoising steps are fixed; larger K adds correction compute. All seeds are shown without selection.

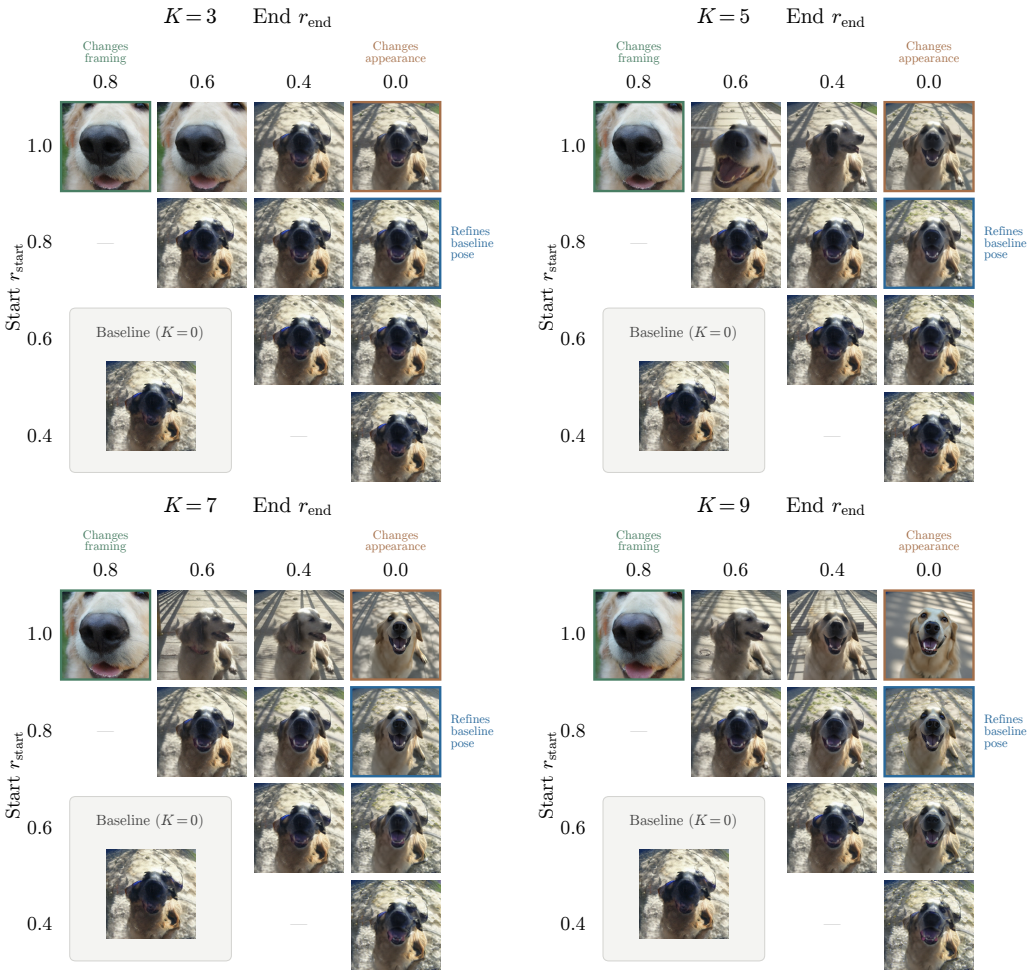


Figure 38: **Skipping early correction refines the baseline pose.** Gray boxes show the matched $K = 0$ baseline; green, blue and orange mark early-only, skip-early and full-range correction, respectively.

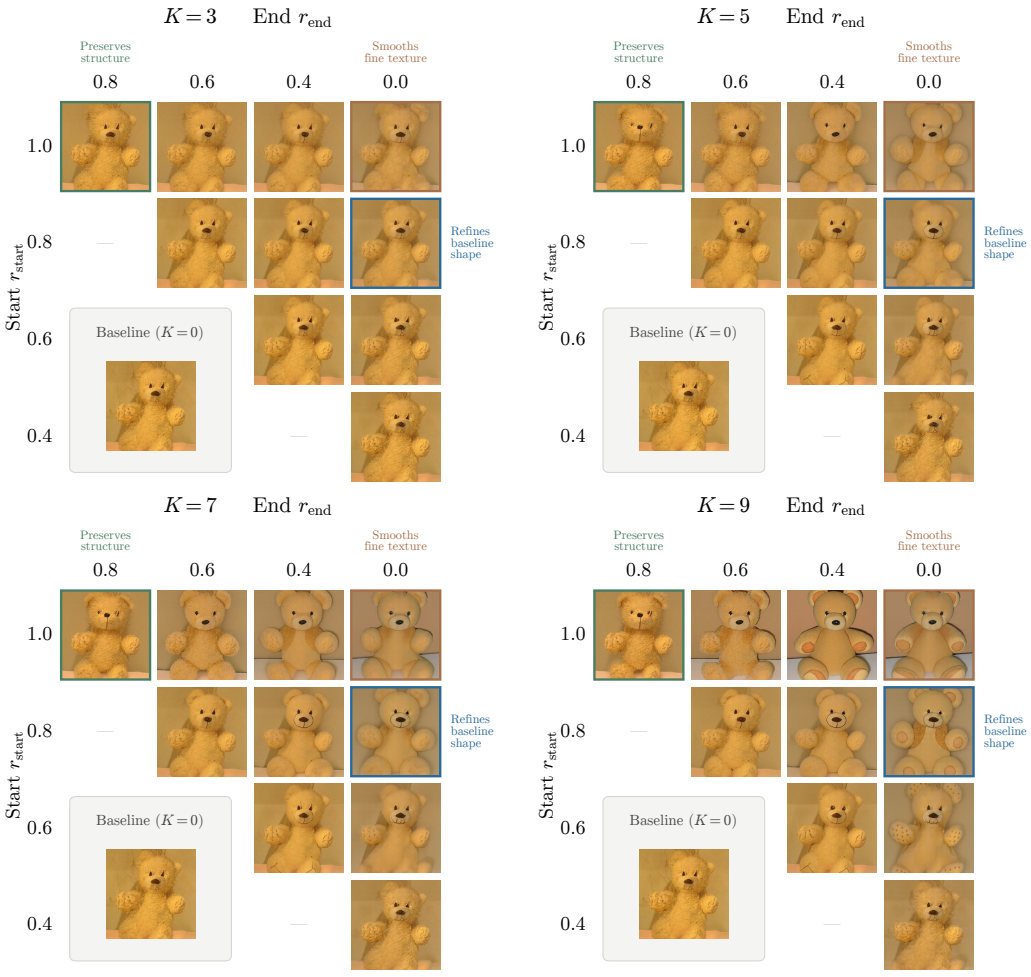


Figure 39: **Full-range correction changes fine texture and shape.** Gray boxes show the matched $K = 0$ baseline; green, blue and orange mark early-only, skip-early and full-range correction, respectively.

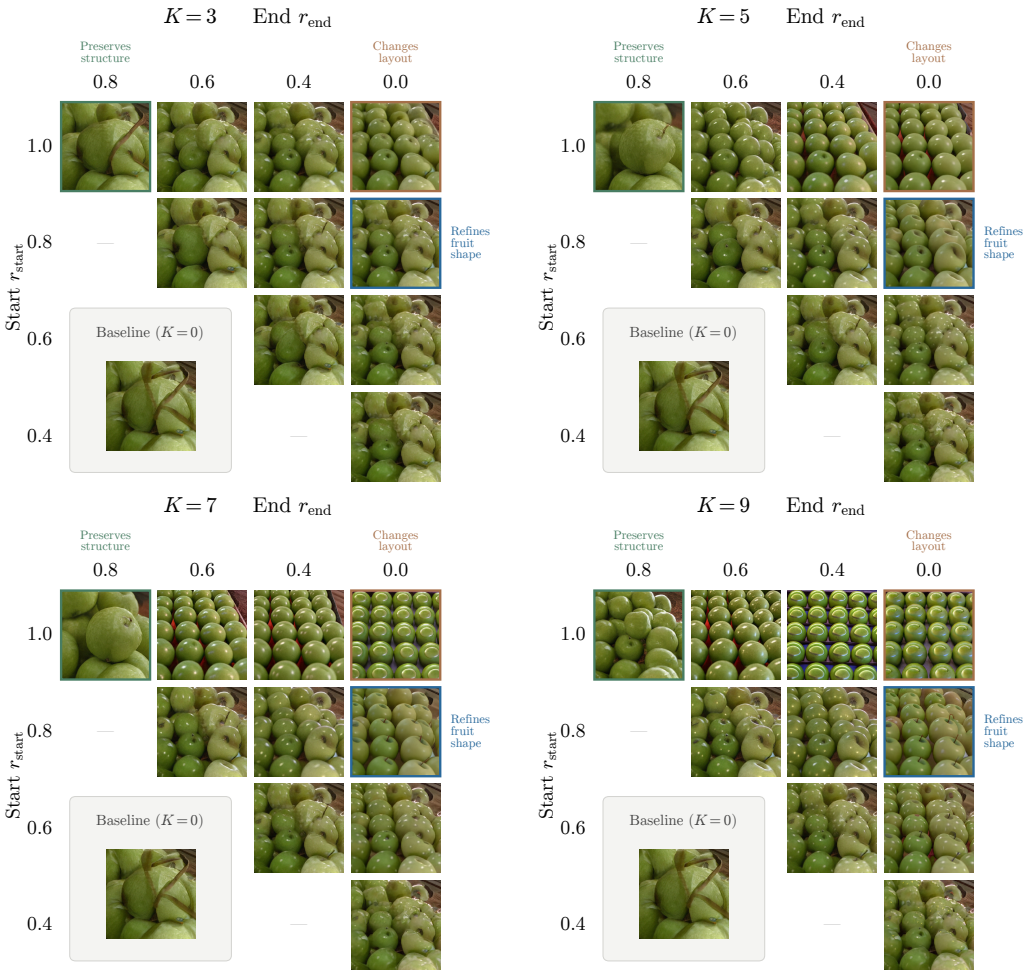


Figure 40: **Full-range correction changes object layout and appearance.** Gray boxes show the matched $K = 0$ baseline; green, blue and orange mark early-only, skip-early and full-range correction, respectively.

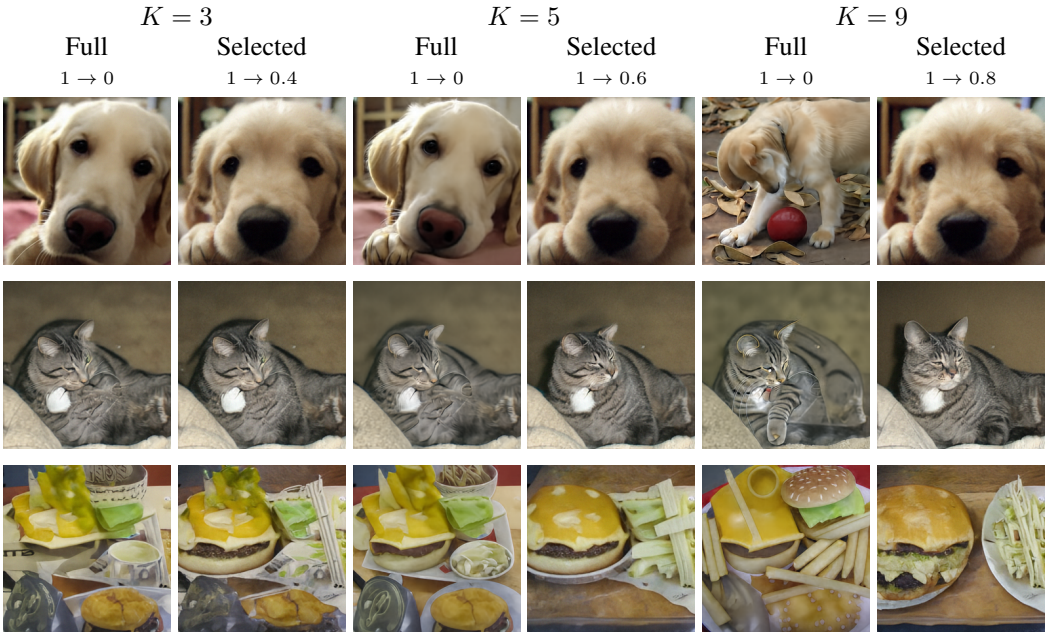


Figure 41: **Restricting correction to early steps changes the generated structure.** Matched-noise comparisons of full-trajectory correction and the interval selected by FID8K for each K.