

GUIDANCENFT: NFT-BASED REWARD FINE-TUNING OF GUIDED DIFFUSION MODELS

Jiayin Liu¹, Zeyu Michael Li², Bingshuo Qian², Xiang Cheng²

¹Tsinghua University ²Duke University

ABSTRACT

DiffusionNFT (NFT) fine-tunes diffusion models toward task-specific rewards, but optimizing these rewards does not guarantee fidelity to the full prompt. Although classifier-free guidance (CFG) strengthens prompt alignment, NFT is formulated as a CFG-free method, and applying CFG to the fine-tuned velocity field can reduce the optimized reward. This motivates optimizing rewards over the guided output distribution while seeking to retain its prompt alignment. However, applying standard NFT directly to guided samples can cause reward degradation or training collapse. We trace this failure to a mismatch between the reference velocity used in the NFT loss and the score of the guided sample distribution, which biases the improvement direction. We propose GuidanceNFT, which estimates the matching reference velocity through online flow matching on guided samples and uses it to correct both regression terms. With an exact reference, the corrected objective recovers NFT’s intended update. In our experiments, the learned correction sustains reward improvement under both explicit and built-in guidance. On SD3.5-M, GuidanceNFT achieves higher optimized rewards than NFT with and without CFG across GenEval, text-rendering, and multi-reward objectives, together with better overall prompt alignment as measured by off-domain evaluation metrics. Experiments on models with built-in guidance, including Model Guidance and ELF, further demonstrate improved ImageNet generation, a better perplexity–entropy tradeoff on OpenWebText, and higher accuracy on GSM8K.

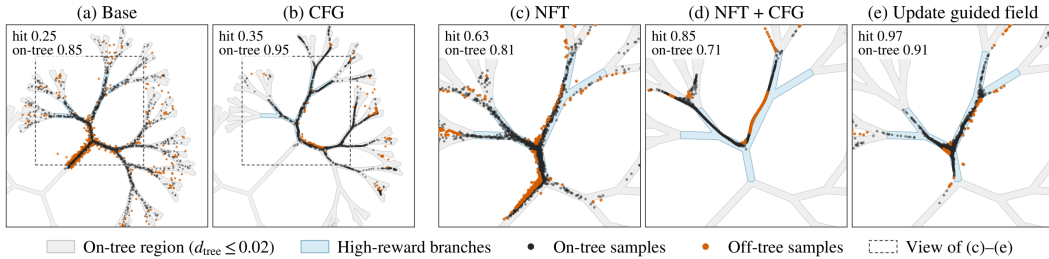


Figure 1: **Updating the guided field improves reward while retaining data structure.** (a–b) Pretrained samples without and with CFG. (c) NFT moves samples toward the rewarded branches, but (d) adding CFG afterward pushes many samples off the tree. (e) Updating the guided field better combines reward targeting with tree fidelity. Panels (c–e) zoom into the dashed region; blue branches mark the reward target and orange points mark off-tree samples.

1 INTRODUCTION

Reward fine-tuning adapts pretrained diffusion models to task-specific objectives. DiffusionNFT (NFT) (Zheng et al., 2026) is a recent and efficient approach that fine-tunes models through forward-process regression by contrasting high- and low-reward generations, requiring neither reward gradients nor trajectory likelihoods. While classifier-free guidance (CFG) strengthens prompt alignment

Emails: jiayin-123@mails.tsinghua.edu.cn;
{zeyu.li030, bingshuo.qian, xiang.cheng}@duke.edu.

by combining conditional and empty-condition predictions (Ho & Salimans, 2022), NFT is formulated as a CFG-free method that, without CFG, exceeds the task rewards of CFG-based fine-tuning such as Flow-GRPO (Liu et al., 2025). However, optimizing a task reward does not guarantee fidelity to the full prompt: an OCR-specialized NFT model can render the requested words while omitting the surrounding scene (Figure 4). Applying CFG to the fine-tuned velocity field does not necessarily restore prompt alignment and, in our SD3.5-M experiments, even reduces the optimized reward for all three training objectives.

This tension motivates applying NFT to improve a guided velocity field with stronger prompt alignment, allowing reward optimization to start from the generation behavior we intend to retain (Figure 1). This approach could also extend NFT to models with *built-in guidance*, whose native predictions are already guided. NFT appears well suited to this setting because it derives an improvement direction from the contrast between positive and negative samples without relying on the internal sampling trajectory. However, we observe reward degradation or training collapse when applying the standard NFT objective directly to guided samples (Figures 3 and 5).

We trace this failure to the reference velocity in the NFT loss. NFT obtains its improvement direction from a regression loss built on velocity targets for re-noised samples and their corresponding rewards. To recover the direction suggested by the positive–negative contrast, it relies on a shared reference velocity that is collinear with the positive and negative conditional mean velocities. The original or guided sampling field, however, need not preserve this property (Section 3). We show that a mismatch in the reference velocity can introduce a bias that rotates the intended update.

We therefore propose *GuidanceNFT*, which estimates the required reference velocity through flow matching on guided samples. This reference turns out to correspond to the conditional score of the rollout distribution. We use it to correct the NFT objective, which then recovers NFT’s intended reward-driven update to the model’s velocity field. In our experiments, this correction prevents the collapse observed with standard NFT and sustains reward improvement.

GuidanceNFT improves both task reward and prompt alignment in our experiments. On SD3.5-M, it achieves higher optimized rewards than NFT with and without CFG across GenEval, text-rendering, and multi-reward objectives, together with better overall prompt alignment as measured by off-domain evaluation metrics. On models with built-in guidance, GuidanceNFT improves ImageNet generation with Model Guidance and achieves a better perplexity–entropy tradeoff on OpenWebText with ELF-B. On the Qwen-based ELF reasoning model, it outperforms NFT’s best retained checkpoint at every reported Pass@ k on GSM8K.

Our main contributions are:

1. **Integrating NFT with guidance.** We apply NFT to guided velocity fields and identify the collinear reference velocity required by its regression objective. Our analysis and experiments link a biased reference to reward degradation and training collapse across multiple settings.
2. **Enabling NFT to tilt guided distributions toward higher rewards.** We introduce GuidanceNFT, which estimates the required reference velocity through flow matching on guided samples and uses it to correct the NFT objective. This correction enables sustained reward improvement in settings where standard NFT degrades or collapses.
3. **Improved task performance and broader alignment.** GuidanceNFT builds on the prompt alignment of guided velocity fields to achieve higher optimized rewards on SD3.5-M, together with better overall prompt alignment as measured by off-domain evaluation metrics. Experiments with Model Guidance and self-conditioning CFG further show improved ImageNet generation, a better perplexity–entropy tradeoff on OpenWebText, and higher accuracy on GSM8K.

2 BACKGROUND

We review the regression objective used by NFT and the explicit and built-in guidance mechanisms that determine the rollout distributions in this work.

2.1 DIFFUSIONNFT: REWARD FEEDBACK THROUGH A SHARED REFERENCE

DiffusionNFT (NFT) (Zheng et al., 2026) fine-tunes a diffusion model using generated samples and their rewards, without requiring reward gradients or trajectory likelihoods. Given an endpoint distribution $q(\cdot | c)$ and processed reward weights $r \in [0, 1]$, it defines positive and negative populations $q^+ \propto rq$ and $q^- \propto (1 - r)q$. The positive population places greater weight on higher-reward samples, motivating an update toward this distribution.

NFT implements this distributional update through the velocity field. Generated endpoints are re-noised to estimate the velocity fields associated with the positive and negative populations. The difference between these fields provides a local improvement direction for the velocity field of the original, untilded distribution.

Rather than fitting the two targets with separate models, NFT trains a single policy v_θ against a frozen copy v^{old} of the policy that generated the samples. For $\beta > 0$, it minimizes

$$\mathcal{L}_{\text{NFT}}(\theta) = \mathbb{E} \left[r \|\beta v_\theta + (1 - \beta)v^{\text{old}} - v\|^2 + (1 - r) \|(1 + \beta)v^{\text{old}} - \beta v_\theta - v\|^2 \right], \quad (1)$$

where x_t is a generated endpoint x re-noised to time t , v is its sample velocity, and both terms are evaluated at the same state (x_t, c, t) . The frozen policy v^{old} serves as the shared reference for the positive and negative terms (Section 3). This objective uses reward feedback to move the policy toward the positive population without requiring the sampling trajectories.

2.2 EXPLICIT AND BUILT-IN GUIDANCE

Classifier-free guidance. Classifier-free guidance (CFG) strengthens conditioning by extrapolating from the empty-condition prediction $v_\emptyset$ toward the conditional prediction v_c (Ho & Salimans, 2022):

$$v_w = v_\emptyset + w(v_c - v_\emptyset) = wv_c + (1 - w)v_\emptyset. \quad (2)$$

Here $w = 1$ gives conditional-only sampling, whereas $w > 1$ amplifies the difference between the two predictions. We refer to this as *explicit* guidance: it is applied at sampling time, requires two model evaluations per step, and can be removed by setting $w = 1$.

Model Guidance. Guidance can also be built into the model’s native prediction. Model Guidance (MG) (Tang et al., 2025) trains the conditional prediction M_c , in velocity coordinates, toward $v + \lambda \text{sg}(M_c - M_\emptyset)$, where sg denotes stop-gradient. For a fixed $0 \leq \lambda < 1$, the ideal fixed point has the CFG form

$$M_c = w_{\text{eff}} C_{\text{data}} + (1 - w_{\text{eff}}) M_\emptyset, \quad w_{\text{eff}} = 1/(1 - \lambda), \quad (3)$$

where C_{data} is the unweighted conditional mean velocity of the pretraining data. The native prediction M_c is therefore already a guided field and requires one model evaluation per inference step. We call this *built-in* guidance: it is fixed during pretraining, so there is no sampling-time guidance scale that can be set to one to recover conditional-only sampling.

Self-conditioning CFG (SCCFG). ELF (Hu et al., 2026) incorporates guidance through its own clean-embedding prediction. The same network produces a self-conditioned field S_θ^w and a field $S_{\theta, \emptyset}^w$ with zero self-conditioning input; both retain any task prompt. The self-conditioning input comes from a detached initial prediction during training and from the preceding sampling step during inference. The training target is

$$v_{\text{target}} = v + (1 - 1/w) \text{sg}(S_\theta^w - S_{\theta, \emptyset}^w). \quad (4)$$

As with MG, the self-conditioned prediction used for sampling is itself a guided field.

Explicit CFG, MG, and SCCFG all produce guided predictions that typically improve sample quality or prompt alignment relative to the unguided conditional prediction. This motivates applying NFT directly to guided predictions. Standard NFT, however, is formulated for conditional-only rollouts ($w = 1$) and does not cover built-in guidance. Section 3 presents GuidanceNFT, which extends NFT to guided samples under both explicit and built-in guidance.

3 GUIDANCENFT

GuidanceNFT adapts NFT to guided samples by correcting its reference velocity. We first express the reward-driven update in terms of the positive and negative velocity fields and identify the reference that aligns both regression terms with this update. We then relate this reference to the score of the re-noised rollout distribution, examine the bias introduced by an inaccurate reference, and estimate the required reference through online flow matching.

3.1 A COLLINEAR REFERENCE FOR POSITIVE AND NEGATIVE FEEDBACK

Following Section 2, we apply NFT to the guided rollout distribution $q_{\text{CFG}}(x \mid c)$, under explicit or built-in guidance, and tilt it toward higher reward. The reward weights $r(x, c)$ tilt $q_{\text{CFG}}(x \mid c)$ into a positive distribution $q_{\text{CFG}}^+(x \mid c)$ and a negative distribution $q_{\text{CFG}}^-(x \mid c)$; re-noising the two populations yields their velocity fields at an intermediate state x_t :

$$v^+(x_t, c, t) = \mathbb{E}_{x \sim q_{\text{CFG}}^+(x \mid x_t, c)}[v], \quad v^-(x_t, c, t) = \mathbb{E}_{x \sim q_{\text{CFG}}^-(x \mid x_t, c)}[v], \quad (5)$$

where $q_{\text{CFG}}^\pm(x \mid x_t, c)$ is the distribution of the endpoints of each population that re-noise to x_t , and v is the sample velocity of the re-noised endpoint. Their contrast $\Delta(x_t, c, t) \propto v^+(x_t, c, t) - v^-(x_t, c, t)$ points from the negative field toward the positive field and is NFT’s improvement direction; its positive scale is fixed below. Taking $\beta = 1$ throughout this section, NFT’s intended update at this state is the minimizer of

$$\ell_{\text{dir}}(v_\theta) = \|(v_\theta - v^{\text{old}}) - \Delta\|^2. \quad (6)$$

Evaluating this objective directly, however, would require fitting v^+ and v^- separately to estimate Δ . NFT instead introduces a shared reference velocity v_{ref} , collinear with v^+ and v^- , and splits the regression into a positive and a negative term:

$$\ell(v_{\text{ref}}; v_\theta) = \alpha \|(v_\theta - v^{\text{old}}) - (v^+ - v_{\text{ref}})\|^2 + (1 - \alpha) \|(v_\theta - v^{\text{old}}) - (v_{\text{ref}} - v^-)\|^2, \quad (7)$$

where $\alpha(x_t, c, t) = \mathbb{E}_{x \sim q_{\text{CFG}}(x \mid x_t, c)}[r(x, c)]$ is the local positive weight. We set $\Delta = 2\alpha(1 - \alpha)(v^+ - v^-)$, the scale at which Equations 6 and 7 can coincide. At x_t , this loss can be rewritten, up to a term independent of v_θ , using only r and the sample velocity v :

$$\mathbb{E}_{x \sim q_{\text{CFG}}(x \mid x_t, c)}[r \|(v_\theta - v^{\text{old}}) - (v - v_{\text{ref}})\|^2 + (1 - r) \|(v_\theta - v^{\text{old}}) - (v_{\text{ref}} - v)\|^2], \quad (8)$$

which can be evaluated from rewarded samples without fitting the two velocity fields separately.

To apply this construction to guided rollouts, we must identify a reference that makes both regression terms contribute along Δ . The unweighted conditional mean velocity of the guided rollouts provides such a reference:

$$\bar{v}_{q_{\text{CFG}}}(x_t, c, t) = \mathbb{E}_{x \sim q_{\text{CFG}}(x \mid x_t, c)}[v] = \alpha v^+ + (1 - \alpha) v^-, \quad (9)$$

where the second equality follows from the definitions of $v^\pm$ and α . Setting $v_{\text{ref}} = \bar{v}_{q_{\text{CFG}}}$ makes $\ell(v_{\text{ref}}; v_\theta)$ equal to $\ell_{\text{dir}}(v_\theta)$ up to a constant independent of the policy displacement, so that the sample-based objective in Equation 8 implements the intended update at every x_t . This reference is equivalent to the score of the re-noised rollout distribution: $\bar{v}_{q_{\text{CFG}}} = A_t x_t + B_t \nabla_{x_t} \log q_{\text{CFG}, t}(x_t \mid c)$ for schedule constants A_t, B_t (Appendix A.2).

3.2 REFERENCE MISMATCH UNDER GUIDANCE

When a model’s velocity matches the conditional mean velocity of its re-noised rollout distribution, the required reference is available directly from the model. Standard NFT uses the frozen prediction v^{old} as this reference. For guided rollouts, however, the conditional prediction v_c alone generally does not represent $\bar{v}_{q_{\text{CFG}}}$. Using the explicitly constructed CFG field $v_w = w v_c + (1 - w) v_\emptyset$ (*analytic correction*) does not generally resolve the mismatch: the sampler’s intermediate distributions can differ from the distributions obtained by re-noising its endpoints. Analyses of CFG distinguish its sampling distribution from the density suggested by the instantaneous guided score (Bradley & Nakkiran, 2025). Thus neither v_c nor v_w is guaranteed to supply the required reference; Appendix A.2 details this distinction.

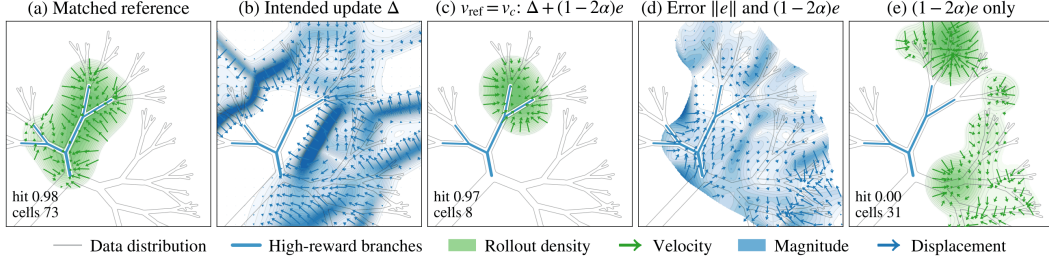


Figure 2: **Reference error distorts the reward-driven update.** (a) A matched reference spreads samples across the rewarded branches through the intended update in (b). (c) An incorrect reference concentrates samples on fewer branches. (d) The reference error adds a displacement that (e), applied alone, moves samples away from the reward target. Hit reports the fraction reaching the reward target and cells the spatial diversity.

An inaccurate reference changes the update learned by NFT. To see this, fix a re-noised state x_t and write $v_{\text{ref}} = \bar{v}_{q_{\text{CFG}}} + e$, where $e = e(x_t, c, t)$ is the reference error at that state. Completing the square in the conditional loss of Equation 7 gives

$$\ell(v_{\text{ref}}; v_{\theta}) = \|(v_{\theta} - v^{\text{old}}) - \Delta - (1 - 2\alpha)e\|^2 + \text{const.} \quad (10)$$

The additional term can alter the magnitude or direction of the intended update, and it cancels only where $\alpha = 1/2$ (Appendix A). Figure 2 isolates this term in the toy tree model with $v_{\text{ref}} = v_c$, where its root-mean-square magnitude in the first round is 2.4 times that of Δ . Applied alone, it moves the sampler off the rewarded branches; added to Δ , it contracts the sampler onto a small part of them (Appendix B.2). In the observed training runs, larger reference gaps accompany faster reward collapse (Figure 3), consistent with the effect of reference error in Equation 10.

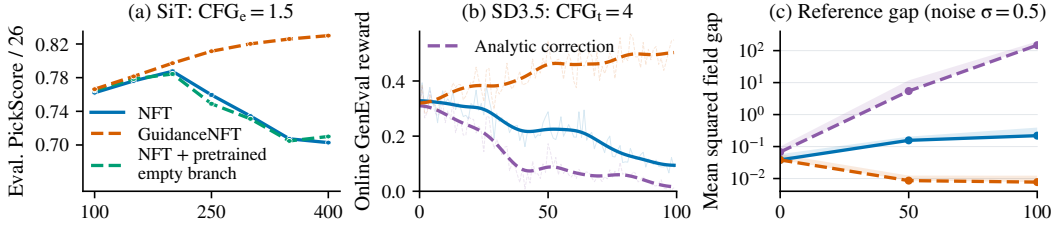


Figure 3: **Reference correction sustains reward improvement under CFG.** (a) At evaluation CFG 1.5, SiT NFT loses reward even after restoring the pretrained empty-condition prediction; GuidanceNFT continues to improve. (b) With CFG-4 rollouts, SD3.5-M NFT and analytic correction collapse, whereas GuidanceNFT improves GenEval reward. (c) Faster collapse accompanies larger reference gaps at noise level $\sigma = 0.5$. Horizontal axes show training updates. Bars in (a) show evaluation-seed std; bands in (c) span the mean to one prompt std above it.

3.3 ONLINE REFERENCE ESTIMATION AND CORRECTION

Since the required reference is not available from the model directly, we estimate it through flow matching on the re-noised guided endpoints. An auxiliary predictor v_{ϕ} is fitted to the unweighted regression targets from the same endpoint population, and its gap to the current reference v^{old} defines the correction $\mathcal{C}_{\phi}$:

$$\mathcal{L}_{\text{FM}}(\phi) = \mathbb{E}_{c, t, x \sim q_{\text{CFG}}(x|c)} \|v_{\phi}(x_t, c, t) - v\|^2, \quad \mathcal{C}_{\phi} = \text{sg}(v_{\phi} - v^{\text{old}}). \quad (11)$$

We refresh this predictor as the rollout distribution changes (Algorithm 1) and apply the same correction $\mathcal{C}_{\phi}$ to both regression terms:

$$\begin{aligned} \mathcal{L}_{\text{GuidanceNFT}}(\theta) = \mathbb{E}_{c, t, x \sim q_{\text{CFG}}(x|c)} [& r(x, c) \|(v_{\theta} - v^{\text{old}}) - (v - v^{\text{old}} - \mathcal{C}_{\phi})\|^2 \\ & + (1 - r(x, c)) \|(v_{\theta} - v^{\text{old}}) - (v^{\text{old}} + \mathcal{C}_{\phi} - v)\|^2]. \end{aligned} \quad (12)$$

This objective uses the reference $v^{\text{old}} + \mathcal{C}_\phi$. With $e = v^{\text{old}} - \bar{v}_{q_{\text{CFG}}}$ the reference error of standard NFT, Equation 10 becomes

$$\ell(v^{\text{old}} + \mathcal{C}_\phi; v_\theta) = \|(v_\theta - v^{\text{old}}) - \Delta - (1 - 2\alpha)(e + \mathcal{C}_\phi)\|^2 + \text{const.} \quad (13)$$

When $\mathcal{C}_\phi = -e$, that is $v_\phi = \bar{v}_{q_{\text{CFG}}}$, the bias cancels and the objective recovers the intended loss in Equation 6. In the SD3.5-M runs shown in Figure 3, online reference correction sustains reward improvement and avoids the collapse observed with the uncorrected and analytically corrected objectives.

4 EXPERIMENTS

Our experiments test whether correcting NFT’s reference velocity enables reward fine-tuning to improve guided models while retaining prompt alignment and generation quality. We first examine training dynamics and reward specialization under explicit CFG, then evaluate models with built-in guidance on ImageNet generation, OpenWebText (Gokaslan et al., 2019) language modeling, and GSM8K reasoning. Training and evaluation protocols, uncertainty estimates, and costs are given in Appendices C and E.

4.1 EXPLICIT CFG: REWARD OPTIMIZATION WITH PROMPT ALIGNMENT

Bringing guidance into reward optimization. On ImageNet (Russakovsky et al., 2015), SiT-XL/2 (Ma et al., 2024) illustrates the difference between optimizing an unguided model and its guided output. Standard NFT’s reward under CFG initially improves and then declines, whereas GuidanceNFT continues to improve (Figure 3(a)). Replacing the empty-condition prediction at inference with its pretrained counterpart, following Phunyaphibarn et al. (2026), yields only a modest recovery, indicating that preserving this branch alone does not sustain the gains. GuidanceNFT instead uses rewards from guided rollouts during training. The selected SiT examples in Figure 6(a) complement the reward curves: GuidanceNFT produces more detailed class instances, including clearer facial features and fur in the coyote and cocker spaniel examples.

Reference correction sustains training on guided rollouts. On SD3.5-M (Stability AI, 2024), standard NFT with CFG-4 rollouts loses GenEval reward and collapses; using the analytic CFG field as the reference accelerates this failure. Online flow-matching estimation of the reference instead yields sustained improvement (Figure 3(b)). Disagreement with separately fitted rollout references follows the same ordering: it grows fastest for the analytic CFG field, more slowly for the uncorrected reference, and remains small for GuidanceNFT (panel (c)). These dynamics are

Table 1: **SD3.5-M reward improvement and quality retention.** W/o and w/ CFG denote evaluation scales 1 and 4. Fine-tuned rows average evaluation seeds 42 and 123; pretrained rows use seed 42. Train reward measures the optimized objective on its test set. Gray marks training criteria; bold marks the best within each objective. Seed stds are given in Appendix D.

Training objective	Method	Train reward	Held-out evaluation						
			GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd
—	Pretrained w/o CFG	—	0.283	0.113	20.69	0.232	0.199	4.990	-0.594
	Pretrained w/ CFG	—	0.636	0.532	22.38	0.278	0.276	5.252	0.820
GenEval	NFT w/o CFG	0.900	0.855	0.434	21.70	0.265	0.234	4.886	0.271
	NFT w/ CFG	0.657	0.565	0.624	21.19	0.271	0.216	4.773	0.148
	GuidanceNFT	0.914	0.877	0.674	21.83	0.277	0.250	4.947	0.682
OCR	NFT w/o CFG	0.971	0.035	0.971	19.61	0.208	0.137	4.208	-1.679
	NFT w/ CFG	0.192	0.265	0.235	20.21	0.242	0.188	4.596	-0.919
	GuidanceNFT	0.975	0.488	0.974	21.83	0.274	0.256	5.120	0.425
Learned3	NFT w/o CFG	1.519	0.671	0.497	23.51	0.284	0.332	5.606	1.397
	NFT w/ CFG	1.495	0.701	0.579	22.99	0.290	0.326	5.397	1.414
	GuidanceNFT	1.528	0.630	0.692	23.24	0.292	0.332	5.653	1.448



Figure 4: **GuidanceNFT retains prompt content during reward specialization.** Selected SD3.5-M examples show object and spatial alignment after Learned3 fine-tuning (first four columns), and readable text within the requested scenes after OCR fine-tuning (last four). Rows compare NFT at CFG 1, NFT at CFG 4, and GuidanceNFT at CFG 4.

consistent with the reference-error bias in Equation 10 and support estimating the reference from the guided rollout distribution.

Higher task reward with better overall performance. Table 1 evaluates SD3.5-M specialized for GenEval (Ghosh et al., 2023), OCR, or Learned3, which combines PickScore (Kirstain et al., 2023), HPSv2.1 (Wu et al., 2023), and CLIPScore (Radford et al., 2021). Adding CFG after standard NFT reduces the optimized reward for all three objectives. GuidanceNFT achieves higher optimized rewards than NFT both with and without CFG, and leads on all seven evaluation metrics for GenEval and OCR specialization. For Learned3, it achieves the highest composite reward together with strong preference and alignment scores. This overall performance supports reward fine-tuning that retains the prompt alignment of guided generation.

OCR illustrates how task reward and broader prompt alignment can improve together. NFT without CFG attains an OCR reward of 0.971 but a held-out GenEval score of only 0.035. GuidanceNFT raises the OCR reward to 0.975 and reaches 0.488 on GenEval. The selected examples in Figure 4 (last four columns) show readable text within the requested scenes, including a watch on a wrist and a sign on a mountain summit. These results indicate that accurate text rendering can coexist with substantially better compositional alignment than standard NFT achieves.

For Learned3, GuidanceNFT attains a composite reward of 1.528, compared with 1.519 for NFT without CFG and 1.495 for NFT with CFG. It also improves OCR, CLIPScore, Aesthetic (Schuhmann et al., 2022), and ImageReward (Xu et al., 2023) over both NFT variants. The selected examples in Figure 4 (first four columns) illustrate retention of requested content, including a portrait within a triangular frame and an umbrella resting on a spoon. Although the gains are not uniform across metrics, these results show that Learned3 optimization can improve reward while retaining image quality and prompt alignment.

4.2 BUILT-IN GUIDANCE: REWARD OPTIMIZATION OF NATIVE GUIDED FIELDS

Model Guidance (MG) (Tang et al., 2025) and ELF self-conditioning guidance (Hu et al., 2026) incorporate guidance into their native predictions. We test whether estimating the reference from each model’s own rollouts also improves reward fine-tuning in these settings.

Sustained improvement in image generation. On MG, GuidanceNFT continues to improve ImageNet reward as standard NFT peaks and then declines (Figure 5(a)). Its selected checkpoint reaches a PickScore of 21.580, compared with 20.010 for NFT’s best retained checkpoint (Appendix F). The selected MG examples in Figure 6(b) show clearer object details, including the snowmobile body

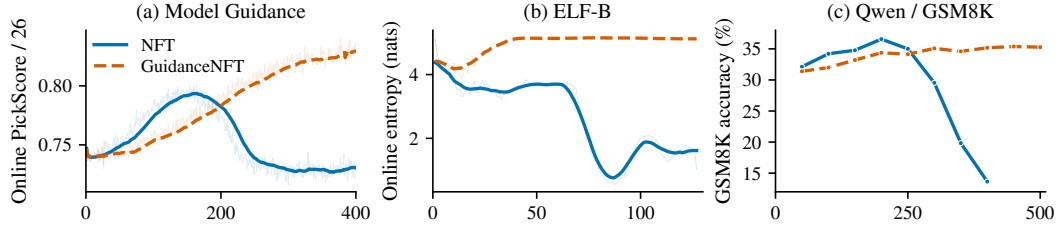


Figure 5: **GuidanceNFT avoids degradation with built-in guidance.** (a) ImageNet reward continues to improve on Model Guidance. (b) On ELF-B, it maintains text entropy on OpenWebText at SCCFG 3. (c) On the Qwen-based ELF model, it retains GSM8K accuracy at SCCFG 2. Standard NFT degrades in all three settings. Horizontal axes show training updates; faint traces and bold curves show raw and smoothed values.

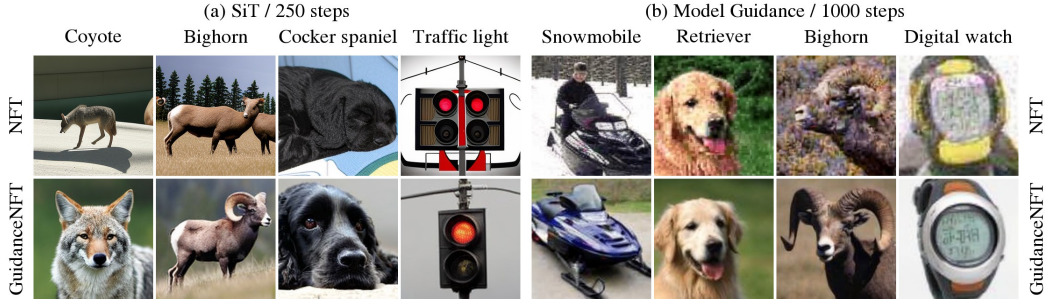


Figure 6: **GuidanceNFT produces more detailed class-conditional images.** Selected ImageNet examples compare NFT (top) and GuidanceNFT (bottom). (a) SiT uses CFG 1 for NFT and CFG 1.5 for GuidanceNFT. (b) Model Guidance uses built-in guidance with external CFG 1.

and the bighorn’s curved horns. Together with the SiT results in Section 4.1, these findings support reference correction for ImageNet reward fine-tuning under both explicit and built-in guidance.

Stable reward optimization for language and reasoning. On ELF-B, standard NFT eventually produces repetitive text on OpenWebText, whereas GuidanceNFT improves reward while retaining varied output (Figures 5(b) and 7, left). A similar pattern appears on the Qwen-based ELF reasoning model: NFT’s accuracy on GSM8K (Cobbe et al., 2021) collapses after an initial improvement, while GuidanceNFT maintains its gains (Figure 5(c)). GuidanceNFT also outperforms NFT’s best retained checkpoint before collapse, raising mean Pass@1 from 41.68% to 43.03% and Pass@8 from 67.02% to 68.61% (Table 2).

Comparison with CEDR across both language tasks. CEDR (Cheng, 2026) adapts NFT to self-conditioned models through velocity reconstruction and achieves strong reasoning results. We compare GuidanceNFT with the released CEDR checkpoint on GSM8K and with our reimplementa-tion of its reconstruction on OpenWebText, denoted CEDR-style reconstruction. On OpenWebText, GuidanceNFT achieves a more favorable perplexity–entropy tradeoff than the pretrained model and CEDR-style reconstruction, retaining higher token entropy than reconstruction at comparable per-plexity (Figure 7, right). On GSM8K, GuidanceNFT exceeds the CEDR checkpoint in mean Pass@ k at every reported k , with Pass@1 increasing from 42.16% to 43.03% and Pass@8 from 66.26% to 68.61% (Table 2). These results support reference correction for both language quality and reason-ing accuracy under the evaluated training configurations, detailed in Appendices C and G.

5 RELATED WORK

Reward fine-tuning through regression. DDPO (Black et al., 2024) and Flow-GRPO (Liu et al., 2025) optimize rewards through trajectory policy gradients. DiffusionNFT (Zheng et al., 2026) instead contrasts rewarded endpoints through forward-process regression. Reward Score Matching (Lee et al., 2026), AdvantageFlow (Kveton et al., 2026), and Reinforce Adjoint Match-

Table 2: **GSM8K with SCCFG-2 evaluation.** Pass@ k (%): at least one correct answer in k attempts. Selected checkpoints are within 500 updates, including NFT’s best retained point. Values are mean $\pm$ problem-bootstrap std (Appendix G).

Method	Pass@1	Pass@2	Pass@4	Pass@8
Pretrained	39.17 $\pm$ 1.04	49.82 $\pm$ 1.15	59.20 $\pm$ 1.21	67.32 $\pm$ 1.29
NFT	41.68 $\pm$ 1.08	51.53 $\pm$ 1.17	59.78 $\pm$ 1.22	67.02 $\pm$ 1.30
CEDR	42.16 $\pm$ 1.11	51.36 $\pm$ 1.19	59.40 $\pm$ 1.24	66.26 $\pm$ 1.32
GuidanceNFT	43.03 $\pm$ 1.10	52.66 $\pm$ 1.16	61.24 $\pm$ 1.20	68.61 $\pm$ 1.26

ing (Bergmeister et al., 2026) also connect reward optimization to regression. Reward-based velocity matching (RVM) (Choi et al., 2026) recovers NFT through reward weights and anchor choices. We identify the reference that aligns NFT’s positive and negative feedback on guided samples, and show how reference error biases the update.

Guidance and the rollout distribution. CFG (Ho & Salimans, 2022) strengthens conditioning through extrapolated predictions; Model Guidance (Tang et al., 2025) and ELF (Hu et al., 2026) incorporate guidance into the native field. Guidance analyses distinguish the sampler’s distribution from the density suggested by its instantaneous score (Bradley & Nakkiran, 2025; Chidambaram et al., 2024; Jiang & Ma, 2026). NFT requires the conditional mean velocity of re-noised endpoints, which need not equal the sampling field. The SCCFG-compatible NFT adaptation of Cheng (2026), denoted CEDR in our experiments, algebraically reconstructs a base prediction. GuidanceNFT instead fits the reference to the current rollout population, covering both explicit and built-in guidance.

Estimating a reference from generated samples. Generated-distribution estimators appear in variational score distillation (Wang et al., 2023) and distribution matching distillation (DMD) (Yin et al., 2024b); DMD2 (Yin et al., 2024a) links inaccurate estimation to training instability. GuidanceNFT uses flow matching (Lipman et al., 2023) on unweighted guided endpoints to estimate the shared reference for NFT’s two regressions. Where guided distillation (Meng et al., 2023) and rectified flow (Liu et al., 2023) train sampling fields, our auxiliary fit calibrates the reward update. Online fitting adds training cost but no inference passes.

Preserving quality beyond the optimized reward. Pretrained empty-condition predictions can retain generation quality after fine-tuning (Phunyaphibarn et al., 2026). Centered Reward Distillation (Zhu et al., 2026) controls drift through within-prompt reward centering and a guided pretrained reference; ReNFT (Bao et al., 2026) repairs diversity collapse after reward training. We correct NFT’s velocity reference during training, a different role from reward centering or pretrained regularization. Our evaluations examine prompt alignment and text diversity alongside task reward.

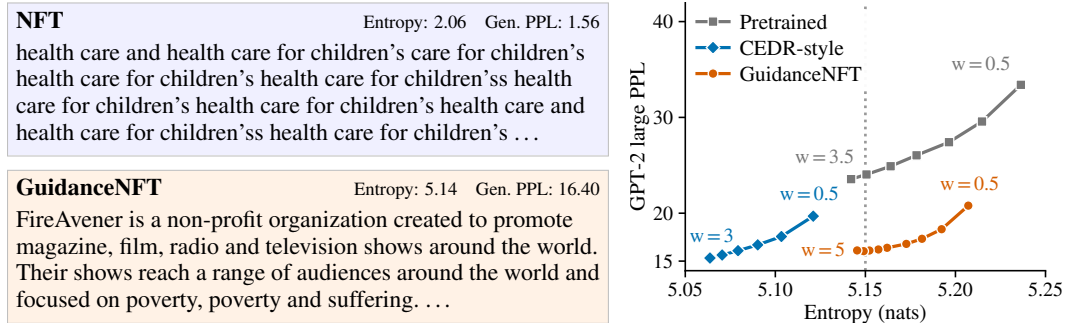


Figure 7: **GuidanceNFT improves the language quality-diversity tradeoff.** Left: ELF-B outputs remain varied with GuidanceNFT but become repetitive with NFT; both train and sample at SCCFG 3. Right: GuidanceNFT improves the perplexity-entropy tradeoff over the pretrained model and CEDR-style reconstruction. GuidanceNFT and reconstruction train at SCCFG 3 and 1, respectively. Markers vary evaluation SCCFG, with endpoint w values labeled. The dotted line marks entropy 5.15 nats for comparing perplexity at matched diversity.

6 CONCLUSION

NFT requires a reference velocity that matches its rollout distribution. Under guidance, a reference gap adds a component to the reward-driven update, which explains why matching sampling and training guidance alone is insufficient. GuidanceNFT estimates the reference online with flow matching and applies a shared correction to NFT’s positive and negative regressions. Across explicit and built-in guidance, it prevents the degradation observed with NFT and improves reward optimization, image quality, and language generation without additional inference passes.

REPRODUCIBILITY STATEMENT

The appendices specify the population assumptions, training and evaluation protocols, checkpoint-selection rules, and uncertainty calculations. Analysis scripts trace the reported summaries to recorded evaluations and distinguish training runs from evaluation seeds.

AI USE STATEMENT

Generative AI tools assisted with literature review, manuscript structure and editing, mathematical checks, experiment design, analysis code, statistical summaries, and figure preparation. The reported model results are drawn from recorded evaluations; analytical diagnostics are identified separately.

REFERENCES

- Yuchen Bao, Chao Wen, Haowei Wang, Ruoxin Chen, Donghao Luo, Jiahui Zhan, Wenjian Huang, Shen Chen, Yiting Wang, Taiping Yao, Chengjie Wang, Shouhong Ding, and Jianguo Zhang. ReNFT: Repairing mode collapse in reward post-training via internal probability-mass recalibration. *arXiv preprint arXiv:2609.00061*, 2026. URL <https://arxiv.org/abs/2609.00061>.
- Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling RL post-training of diffusion and flow-matching models. *arXiv preprint arXiv:2605.10759*, 2026. URL <https://arxiv.org/abs/2605.10759>.
- Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, pp. 4965–4987, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/14f75513f0f1ca01de1e826b52e6b840-Abstract-Conference.html.
- Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *Transactions on Machine Learning Research*, 2025. URL <https://openreview.net/forum?id=zrWNtzSZsf>.
- Xiang Cheng. Reasoning with continuous latent diffusion. *arXiv preprint arXiv:2609.35694*, 2026. URL <https://arxiv.org/abs/2609.35694>.
- Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? a fine-grained analysis in a simple setting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 84968–85005, 2024. doi: 10.52202/079017-2698. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9a3942c235daa9c5f62a8598ae81a946-Abstract-Conference.html.
- Jaemoo Choi, Wei Guo, Yuchen Zhu, Arash Vahdat, Molei Tao, Julius Berner, and Yongxin Chen. Scaling reinforcement learning for diffusion models via velocity matching. *arXiv preprint arXiv:2608.23664*, 2026. URL <https://arxiv.org/abs/2608.23664>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John

- Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152, 2023. doi: 10.52202/075280-2270. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67ble7b1-Abstract-Datasets_and_Benchmarks.html.
- Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. OpenWebText corpus. <http://Skylion007.github.io/OpenWebTextCorpus>, 2019.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021. doi: 10.18653/v1/2021.emnlp-main.595. URL <https://aclanthology.org/2021.emnlp-main.595/>.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL <https://arxiv.org/abs/2207.12598>.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026. URL <https://arxiv.org/abs/2605.10938>.
- Enze Jiang and Zheng Ma. Analytic distribution of classifier-free guidance for schedule design. *arXiv preprint arXiv:2607.19725*, 2026. URL <https://arxiv.org/abs/2607.19725>.
- Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52996–53021, 2024. doi: 10.52202/079017-1679. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5ee7ed60a7e8169012224dec5fe0d27f-Abstract-Conference.html.
- Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663, 2023. doi: 10.52202/075280-1594. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/73aacd8b3b05b4b503d58310b523553c-Abstract-Conference.html.
- Branislav Kveton, Anup Rao, Subhojyoti Mukherjee, Krishna Kumar Singh, and Viet Dac Lai. AdvantageFlow: Regularized advantage-weighted RL in flow models. *arXiv preprint arXiv:2605.26013*, 2026. URL <https://arxiv.org/abs/2605.26013>.
- Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models. *arXiv preprint arXiv:2604.17415*, 2026. URL <https://arxiv.org/abs/2604.17415>. Also presented at the ICML 2026 SPIGM Workshop.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL <https://arxiv.org/abs/2210.02747>.
- Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40783–40818, 2025. doi: 10.52202/085713-1362. URL

- https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a10c46572628d58cb44fb705f25cbbf-Abstract-Conference.html.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL <https://arxiv.org/abs/2209.03003>.
- 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*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/9828_ECCV_2024_paper.php.
- Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Meng_On_Distillation_of_Guided_Diffusion_Models_CVPR_2023_paper.html.
- Prin Phunyaphibarn, Phillip Y. Lee, Jaihoon Kim, and Minhyuk Sung. Unconditional priors matter! improving conditional generation of fine-tuned diffusion models. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3982–3992, 2026. URL https://openaccess.thecvf.com/content/WACV2026/html/Phunyaphibarn_Unconditional_Priors_Matter_Improving_Conditional_Generation_of_Fine-Tuned_Diffusion_Models_WACV_2026_paper.html.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL <https://link.springer.com/article/10.1007/s11263-015-0816-y>.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- Stability AI. Stable Diffusion 3.5 Medium model card. Hugging Face model card, 2024. URL <https://huggingface.co/stabilityai/stable-diffusion-3.5-medium>. Accessed 2026-09-25.
- Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025. URL <https://arxiv.org/abs/2502.12154>.
- Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In

- Advances in Neural Information Processing Systems*, volume 36, pp. 8406–8441, 2023. doi: 10.52202/075280-0368. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1a87980b9853e84dfb295855b425c262-Abstract.html.
- Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. URL <https://arxiv.org/abs/2306.09341>.
- Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023. doi: 10.52202/075280-0700. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487, 2024a. doi: 10.52202/079017-1505. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.
- Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.
- Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, pp. 134129–134150, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d8e68ddfe22520b45b8fb8d5cbde5a21-Abstract-Conference.html.
- Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Diffusion reinforcement learning via centered reward distillation. *arXiv preprint arXiv:2603.14128*, 2026. URL <https://arxiv.org/abs/2603.14128>.

A DERIVATIONS FOR GUIDANCENFT

We derive the reference required by NFT, quantify the bias caused by a reference gap, and connect reference estimation to the resulting policy update. The rollout distribution and reward weights are fixed within each update; all second moments are finite. Population optima are taken over unrestricted square-integrable fields, with $\beta > 0$. For Gaussian noising, assume differentiable schedules and sufficient regularity to differentiate under the integral:

$$X_t = a_t X + \sigma_t \epsilon, \quad U = \dot{a}_t X + \dot{\sigma}_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (14)$$

Here $c \sim p(c)$, $X \sim q(\cdot | c)$, and $t \sim p(t)$, with t and ϵ independent of X . Time runs from noise to data, and $a_t, \sigma_t > 0$ at interior times. We write $z = (X_t, c, t)$, $q = q_{\text{CFG}}$, $C_q = \bar{v}_{q_{\text{CFG}}}$, and $d = v_\theta - \text{sg}(v^{\text{old}})$. The target U is the sample velocity v in the main text.

A.1 REWARD-WEIGHTED TARGETS AND THE REFERENCE ERROR

Let $Z(c) = \mathbb{E}_q[r | c] \in (0, 1)$ and define

$$q^+(X | c) = \frac{r(X, c)q(X | c)}{Z(c)}, \quad q^-(X | c) = \frac{[1 - r(X, c)]q(X | c)}{1 - Z(c)}. \quad (15)$$

At a noisy state, their regression targets are

$$\alpha(z) = \mathbb{E}[r | z], \quad v^+(z) = \frac{\mathbb{E}[rU | z]}{\alpha(z)}, \quad v^-(z) = \frac{\mathbb{E}[(1 - r)U | z]}{1 - \alpha(z)}. \quad (16)$$

Positive Gaussian kernels ensure $0 < \alpha < 1$. Weighted least squares identifies these fields as the two optimal predictions because

$$\mathbb{E}[r\|h - U\|^2 | z] = \alpha\|h - v^+\|^2 + \mathbb{E}[r\|U - v^+\|^2 | z], \quad (17)$$

and the analogous identity holds for $1 - r$. The cross term vanishes by $\mathbb{E}[r(U - v^+) | z] = 0$. Adding the two weighted conditional means gives

$$C_q = \mathbb{E}[U | z] = \alpha v^+ + (1 - \alpha)v^-, \quad \Delta = 2\alpha(1 - \alpha)(v^+ - v^-). \quad (18)$$

For an arbitrary frozen reference C (v_{ref} in the main text), NFT minimizes

$$\mathcal{L}(C; d) = \mathbb{E}[r\|C + \beta d - U\|^2 + (1 - r)\|C - \beta d - U\|^2]. \quad (19)$$

The shared correction in Equation 12 sets $C = v^{\text{old}} + C_\phi$. Applying the weighted-regression identity reduces its conditional objective, up to terms independent of d , to

$$\ell_z(C; d) = \alpha\|C + \beta d - v^+\|^2 + (1 - \alpha)\|C - \beta d - v^-\|^2. \quad (20)$$

Its gradient and unique minimizer satisfy

$$\begin{aligned} \nabla_d \ell_z &= 2\beta[\beta d + (2\alpha - 1)C - \alpha v^+ + (1 - \alpha)v^-], \\ \beta d_C^* &= \alpha(v^+ - C) + (1 - \alpha)(C - v^-) \\ &= \Delta + (1 - 2\alpha)(C - C_q). \end{aligned} \quad (21)$$

This proves Equation 10, which takes $\beta = 1$. Reference error contributes a separate displacement, rather than simply rescaling the intended one. At states with $\alpha \neq 1/2$, exact recovery of the intended displacement requires $C = C_q$; at $\alpha = 1/2$, the reference-error term cancels.

For $C = C_q$, completing the square gives

$$\mathcal{L}(C_q; d) = \beta^2 \mathbb{E} \left\| d - \frac{\Delta}{\beta} \right\|^2 + \bar{K}, \quad (22)$$

where $\bar{K}$ is independent of d . Consequently,

$$v_{\theta^*} = v^{\text{old}} + \frac{\Delta}{\beta}. \quad (23)$$

The correction changes the regression reference while retaining the old policy as the baseline of the update.

A.2 WHY A GUIDED SCORE NEED NOT MATCH ITS ROLLOUT DISTRIBUTION

For $K_t(x | X) = \mathcal{N}(x; a_t X, \sigma_t^2 I)$, the identities $\nabla_x \log q_t(x | c) = -\mathbb{E}[\epsilon | z]/\sigma_t$ and $C_q = \dot{a}_t \mathbb{E}[X | z] + \dot{\sigma}_t \mathbb{E}[\epsilon | z]$ express the reference through the score of the re-noised endpoint distribution:

$$C_q(x, t, c) = A_t x + B_t \nabla_x \log q_t(x | c), \quad A_t = \frac{\dot{a}_t}{a_t}, \quad B_t = \frac{\dot{a}_t \sigma_t^2}{a_t} - \sigma_t \dot{\sigma}_t. \quad (24)$$

A valid reference therefore corresponds to a score family generated by forward noising of a single endpoint distribution.

For exact component scores $s_g = \nabla \log p_{g,t}$ and $s_b = \nabla \log p_{b,t}$, CFG gives

$$w s_g + (1 - w) s_b = \nabla \log \left(p_{g,t}^w p_{b,t}^{1-w} \right). \quad (25)$$

A score at each time need not define a family consistent with a single endpoint distribution under forward noising (Bradley & Nakkiran, 2025; Chidambaram et al., 2024; Jiang & Ma, 2026). To see the distinction directly, consider the special case of additive Gaussian noising, parameterized by increasing noise variance τ . Both component densities satisfy the heat equation $\partial_\tau p = \frac{1}{2} \nabla^2 p$. For $h_\tau = p_{g,\tau}^w p_{b,\tau}^{1-w}$, direct differentiation gives

$$\frac{\partial_\tau h_\tau - \frac{1}{2} \nabla^2 h_\tau}{h_\tau} = \frac{w(1-w)}{2} \|s_g - s_b\|^2. \quad (26)$$

When the right-hand side varies with x , a time-dependent normalizer cannot remove it: even a normalizable guided density need not solve the same forward equation. Special component families can satisfy it.

An exact Gaussian mismatch. Take $a_t = t$, $\sigma_t = 1-t$, component endpoint distributions $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 4)$, and CFG scale $w = 2$. Write $D_g = t^2 + (1-t)^2$ and $D_b = 4t^2 + (1-t)^2$. The component velocities are $v_g = \frac{1}{2} \partial_t \log D_g x$ and $v_b = \frac{1}{2} \partial_t \log D_b x$. Starting from $\mathcal{N}(0, 1)$, the guided ODE with $g = 2v_g - v_b$ therefore has variance D_g^2/D_b , and its endpoint distribution is $q = \mathcal{N}(0, 1/4)$. Re-noising that endpoint distribution instead gives variance $t^2/4 + (1-t)^2$. At $t = 1/4$,

$$\begin{aligned} \text{Var}(X_t^{\text{sampler}}) &= 25/52, & \text{Var}(X_t^{\text{renoise}}) &= 37/64, \\ g(x, t) &= -\frac{124}{65}x, & C_q(x, t) &= -\frac{44}{37}x, \\ e(x, t) &= g(x, t) - C_q(x, t) = -\frac{1728}{2405}x. \end{aligned} \quad (27)$$

No neural approximation or numerical solver is involved. The example instantiates the Gaussian guidance analyses cited above; its implication for NFT is the additional term in Equation 10.

A.3 REFERENCE ACCURACY

The flow-matching objective in Equation 11 has the standard conditional-mean decomposition (Lipman et al., 2023):

$$\mathcal{L}_{\text{FM}}(\phi) = \mathbb{E} \|v_\phi - C_q\|^2 + \mathbb{E} \|U - C_q\|^2. \quad (28)$$

For two frozen predictors evaluated under the same distribution, writing $R_q(h) = \mathbb{E} \|h(z) - U\|^2$ gives

$$R_q(h_1) - R_q(h_2) = \mathbb{E} \|h_1 - C_q\|^2 - \mathbb{E} \|h_2 - C_q\|^2. \quad (29)$$

An empirical comparison uses the same held-out endpoints, times, noise, masks, and conditioning inputs for both predictors. A positive risk difference supports smaller reference MSE for h_2 without estimating the irreducible variance.

For $e = v_\phi - C_q$, Equation 21 gives $d_{v_\phi}^* - d_{C_q}^* = (1 - 2\alpha)e/\beta$, and hence

$$\mathbb{E} \|d_{v_\phi}^* - d_{C_q}^*\|^2 = \frac{1}{\beta^2} \mathbb{E} [(1 - 2\alpha)^2 \|e\|^2] \leq \frac{R_q(v_\phi) - R_q(C_q)}{\beta^2}. \quad (30)$$

The error of the regression target learned by NFT is therefore controlled by the FM excess risk of the reference.

A.4 TRAINING PROCEDURE

Algorithm 1 describes one outer iteration of GuidanceNFT. The rollout distribution, reward weights, and old-policy predictions are held fixed within the policy update. The flow-matching objective for the reference is unweighted by reward and uses the same endpoint population as the NFT objective.

Algorithm 1 GuidanceNFT (one outer iteration)

1. Freeze the rollout policy and baseline prediction v^{old} .
 2. Generate endpoints X with the specified guidance and sampler; evaluate and freeze reward weights r .
 3. Draw fresh (t, ϵ) and construct (X_t, U) by Equation 14.
 4. Refresh v_ϕ with unweighted FM on the endpoint population used by the NFT objective.
 5. Freeze v_ϕ and form the policy displacement $d = v_\theta - v^{\text{old}}$.
 6. Form $\mathcal{C}_\phi = \text{sg}(v_\phi - v^{\text{old}})$ and regress $v^{\text{old}} + \mathcal{C}_\phi \pm \beta d$ in the two terms of Equation 12.
 7. Update the policy using the NFT regressions and the regularization penalty (Appendix C); refresh the rollout policy according to its schedule.
-

B TOY EXPERIMENTS

B.1 GUIDED-FIELD NFT

The tree geometry follows the Autoguidance toy construction (Karras et al., 2024). Figure 1 compares the pretrained model with and without CFG, NFT, guidance applied after NFT, and NFT trained on the guided field; Table 3 reports their metrics. All five sampled configurations use the same 2,048 evaluation noises. Guidance applied after NFT yields reward 0.985 and on-tree rate 71.5%. Guided-field training improves these to 0.996 and 90.7%, respectively; its on-tree rate remains below the 94.6% of pretrained CFG.

The conditional model is the class-A width-64 network; its guide is the pretrained class-AB width-32 network. Panel (b) applies CFG with this pretrained pair; all guided panels use $w = 3$. Panel (d) changes only the conditional model to its ten-update NFT checkpoint. Panel (e) jointly trains the two predictions through their combined guided field. Training uses 512 rollouts per update, eight rollout steps, learning rate 0.002, and ten outer updates; evaluation uses 32 steps. Hit denotes $d_{\text{target}} \leq 0.04$, and on-tree denotes $d_{\text{tree}} \leq 0.02$. Reward is $\exp[-(d_{\text{target}}/0.55)^2]$, so target proximity and geometric fidelity are separate measurements.

Table 3: **Endpoint metrics for the toy experiment in Appendix B.1.** Figure 1 displays the same five configurations.

Method	Reward $\uparrow$	Hit $\uparrow$	On-tree $\uparrow$
Base	0.755	0.254	0.848
CFG	0.721	0.347	0.946
NFT	0.956	0.630	0.805
NFT + CFG	0.985	0.851	0.715
Guided-field NFT	0.996	0.971	0.907

B.2 ISOLATED EFFECT OF REFERENCE ERROR

Equation 10 predicts that an inaccurate reference adds the displacement $(1 - 2\alpha)e$ to the intended update Δ . We examine this term in the CFG sampler of the tree model (Appendix B.1; class-A width-64 conditional model, class-AB width-32 guide, $w = 4$) by applying each displacement directly, without training a network. In each of ten rounds, 65,536 endpoints are sampled from the current field with 32 steps and fixed initial noises. The re-noised endpoint population is treated as a Gaussian mixture, from which $\bar{v}_{q_{\text{CFG}}}$, α , and Δ are evaluated exactly on a 448×448 grid at every sampling noise level $\sigma \geq 0.02$, and the selected displacement is added pointwise to the sampler field. We compare the matched reference $v_{\text{ref}} = \bar{v}_{q_{\text{CFG}}}$, for which the update reduces to Δ , with the two references discussed in Section 3: the current guided field v_w and the frozen conditional

prediction v_c . For each of the latter we apply either the error-induced term $(1 - 2\alpha)e$ alone or the full displacement $\Delta + (1 - 2\alpha)e$, where $e = v_{\text{ref}} - \bar{v}_{q_{\text{CFG}}}$.

Figure 8 shows the fields at $\sigma = 0.093$. Both references differ from $\bar{v}_{q_{\text{CFG}}}$ over most of the rollout support: in the first round the density-weighted relative error $\|e\|_{\text{rms}}/\|\bar{v}_{q_{\text{CFG}}}\|_{\text{rms}}$ is 1.03 for v_w and 0.75 for v_c , and the induced term is larger than the intended update (root-mean-square magnitude 0.47 and 0.32 against 0.13 for Δ). Applied alone, the induced term redirects the sampler although it carries no reward-driven displacement: with $v_{\text{ref}} = v_w$ the on-tree rate falls from 96.5% to 74.5%, and with $v_{\text{ref}} = v_c$ the hit rate falls from 28.4% to 0.0%, whereas the matched reference concentrates the sampler on the rewarded branches. When the induced term is added to Δ , the samples still reach the rewarded branches (hit rate 96.4% and 97.4%, against 97.9% for the matched reference), but they concentrate on a smaller part of them: the effective number of occupied cells is 40 for v_w and 8 for v_c , compared with 73 for the matched reference and 182 before any update. Because the reward saturates on the target branches, the reference error appears as a loss of coverage rather than of reward. Hit and on-tree rates follow Appendix B.1; the effective number of occupied cells is the exponential of the entropy of the endpoint histogram with cell width 0.02.

C TRAINING AND EVALUATION PROTOCOLS

We distinguish the rollout distribution, the reference in the NFT loss, and the prediction that receives its gradient. Write $g_{\theta,w} = wv_{\theta,c} + (1-w)v_{\theta,\emptyset}$. “Conditional update” identifies the prediction in the loss; shared parameters can also change the empty-condition prediction. Only an explicitly frozen pretrained branch remains fixed at inference. Table 4 lists the three for every method.

Table 4: **Rollout distribution, loss reference, and loss prediction.** q_1 denotes unguided rollouts and q_w guided rollouts (q_{CFG} in Section 3), including built-in guidance with external CFG 1. SD3.5 uses $w = 4$, SiT uses $w = 1.5$, and the Qwen-based ELF reasoning model uses native SCCFG 2. Conditional output denotes $v_{\theta,c}$; the native guided field is the model’s built-in guided prediction. Loss references use detached old outputs or a fitted reference. For CEDR, the reconstructed reference and loss prediction apply its detached SCCFG correction. NFT + pretrained empty branch (Phunyaphibarn et al., 2026) replaces only the empty-condition prediction $v_{\emptyset}$ (the empty branch) at inference.

Method	Rollouts	Loss reference	Loss prediction
<i>SD3.5 and SiT: explicit CFG</i>			
NFT	q_1	Conditional output	Conditional output
NFT + pretrained empty branch	q_1	Conditional output	Conditional output
NFT (uncorrected)	q_w	Conditional output	Conditional output
NFT (analytic correction)	q_w	Analytic CFG field	Conditional output
GuidanceNFT	q_w	Fitted reference	Conditional output
<i>Model Guidance: built-in guidance</i>			
NFT	q_w	Native guided field	Native guided field
GuidanceNFT	q_w	Fitted reference	Native guided field
<i>Qwen-based ELF reasoning model (GSM8K): native SCCFG 2</i>			
NFT	q_w	Native guided field	Native guided field
GuidanceNFT	q_w	Fitted reference	Native guided field
			Native guided field with
CEDR	q_w	Reconstructed base pre-diction	detached correction

The pretrained-empty-branch control is applied at inference only: it uses $wv_{\theta,c} + (1-w)v_{\text{pre},\emptyset}$ on the NFT checkpoint and requires no additional training run. On GSM8K, our NFT and GuidanceNFT runs use native velocities without CEDR’s velocity reconstruction. Appendix G separately specifies CEDR’s detached SCCFG correction.

SD3.5 settings. The reference comparison uses five rollout steps, 512×512 resolution, 48 prompt groups with 24 images each, one policy update per iteration, $\beta = 1$, and regularization coefficient

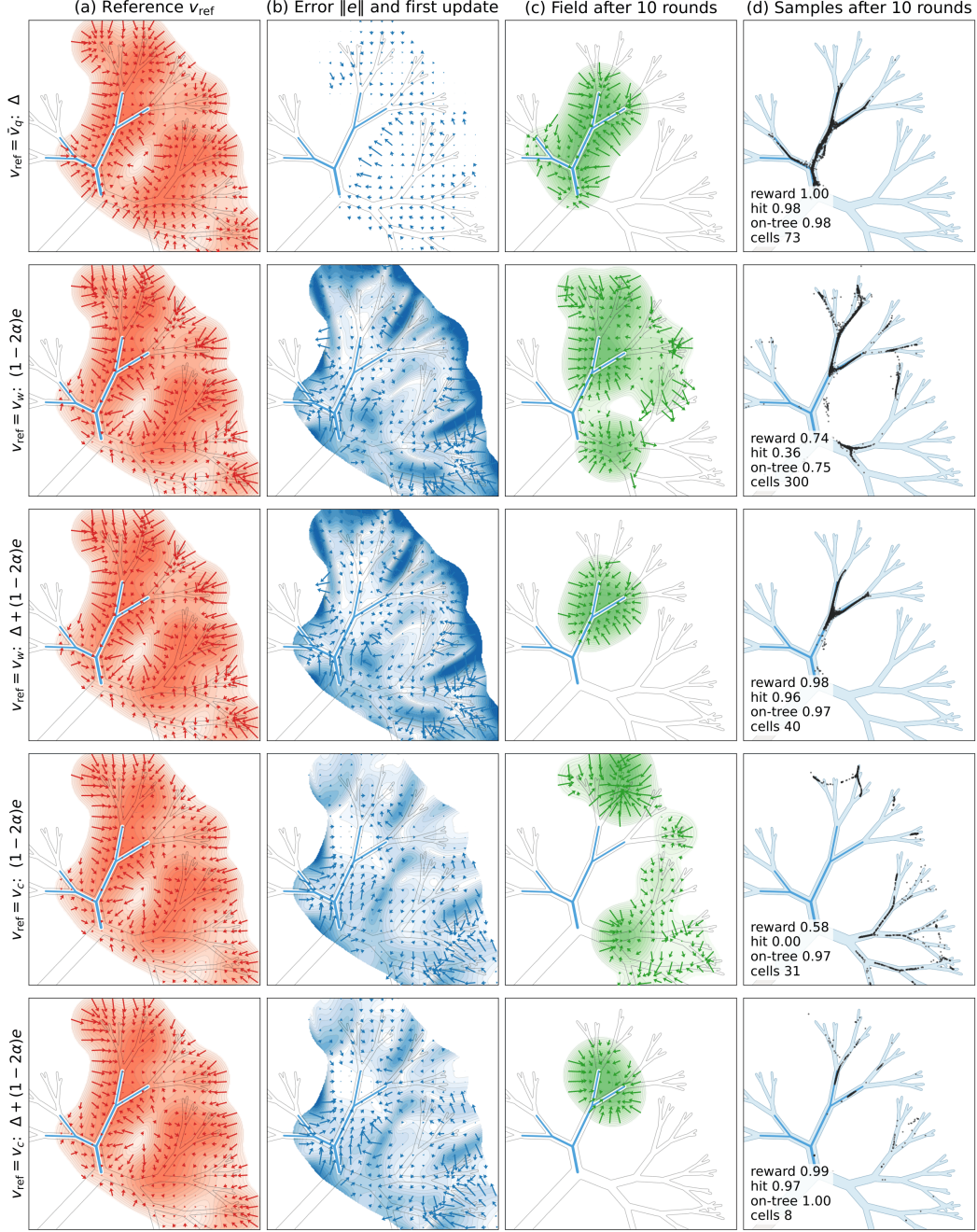


Figure 8: **Reference error redirects samples or reduces reward-target coverage.** Rows compare the intended update with reference-error displacements, applied alone or together with it. Columns show (a) the reference field, (b) reference-error magnitude and update direction, (c) the resulting field and rollout density, and (d) final samples. Error alone redirects samples away from rewarded branches; combined with the intended update, it narrows their spatial coverage. Blue branches mark the reward target.

10^{-4} . All three runs update the conditional prediction. Their corrections are $\mathcal{C} = 0$, $\text{sg}(v_4^{\text{old}} - v_c^{\text{old}})$, and $\text{sg}(v_\phi - v_c^{\text{old}})$, respectively. The learned reference is persistent, with one fitting epoch per refresh, learning rate 3×10^{-4} , gradient clipping at 1, and EMA decay 0.9. The loss uses clean-sample prediction space and detached branch-MAE normalization.

The update-200 GenEval and Learned3 runs use ten-step training rollouts. Reference initialization uses 100 fitting epochs with the policy frozen: 24 rollout batches for fitting and four held out, four time draws, gradient accumulation 24, and uniform continuous-time sampling. NFT and GuidanceNFT omit conditioning dropout. Learned3 uses $\beta = 0.1$ and GenEval $\beta = 1$; both use regularization coefficient 10^{-4} . Learned3 optimizes PickScore/26 + CLIPScore + HPSv2.1; reported PickScore is unscaled.

Scheduled PickScore assessments use 2,048 Pick-a-Pic test prompts, 40 steps, and a fixed evaluation seed. Logged scores are multiplied by 26 to recover unscaled PickScore. These assessments use held-out prompts, separate from rollout rewards and DrawBench (Saharia et al., 2022). The training splits contain 50,000 GenEval metadata rows and 25,432 Learned3 prompts, separate from DrawBench. OCR uses 1,018 evaluation examples per seed and is held out for the GenEval and Learned3 runs. The OCR-trained runs follow the NFT OCR configuration: 19,653 training prompts, ten-step training rollouts, $\beta = 0.1$, two policy updates per iteration, regularization coefficient 10^{-4} , and its old-policy decay schedule.

Reward and loss conventions. PickScore/Pick-a-Pic (Kirstain et al., 2023), HPSv2 (Wu et al., 2023), and ImageReward (Xu et al., 2023) provide learned preference scores. HPS uses the v2.1 checkpoint. The CLIPScore column follows NFT’s scorer: the paired image–text logit divided by 100, with CLIP ViT-L/14 (Radford et al., 2021) as its source-code default. It is not the original rescaled CLIPScore definition of Hessel et al. (2021). SD3.5 uses AdamW with learning rate 3×10^{-4} , coefficients (0.9, 0.999), weight decay 10^{-4} , $\epsilon = 10^{-8}$, and gradient clipping at 1. Training uses FP16, two H200 GPUs, and LoRA (Hu et al., 2022); the retained Learned3 checkpoint specifies rank 32, scaling parameter 64, and dropout 0. The conditional microbatch is 24 per GPU with accumulation 24. The sampler is deterministic DPM2. Following the NFT implementation, training uses a fraction 0.99 of the rollout time grid.

Let s_i be the summed raw task score, $\bar{s}_{c_i}$ its prompt-group mean in the current collected batch, and σ_B the standard deviation across that batch. The SD3.5 reward mapping is

$$A_i = \frac{s_i - \bar{s}_{c_i}}{\sigma_B + 10^{-4}}, \quad R_i = \frac{1}{2} + \frac{\text{clip}(A_i, -5, 5)}{10}. \quad (31)$$

Prompt statistics are cleared after each outer update. The NFT branch losses receive weights R_i and $1 - R_i$, with an overall multiplier $5/\beta$. In clean-sample prediction space, a velocity residual is multiplied by the noise time. Its squared residual is divided by its detached per-example mean absolute residual, lower-bounded by 10^{-5} . The regularization term penalizes the deviation of the policy prediction from that of a frozen copy of the model. These adaptive denominators and regularizers follow the NFT implementation.

Held-out GenEval aggregates retain all valid rewards, including the minimum value -10 . The online reward curve in Figure 3(b) excludes samples with reward -10 , whereas the advantage calculation during training uses all rewards.

ELF-B settings. OpenWebText training uses 384 endpoints per outer update, eight time draws per endpoint, 100 updates, learning rate 3×10^{-5} , $\beta = 1$, regularization coefficient 0.1, decoder cross-entropy coefficient 0.25, and an adaptive entropy-floor term targeting 5.1 nats. The quality evaluator is GPT-2 large (Radford et al., 2019). Native SCCFG changes the model’s self-conditioning field; external CFG is 1 throughout the reported sweep. The comparison matches policy-update counts, and learned-reference training adds estimator updates. Every sweep point uses 1,024 samples, a shared evaluation seed, 32 SDE steps, and $\gamma = 1.5$. The separate repetition diagnostic uses 1,000 samples.

Figure 7 evaluates the pretrained model, CEDR-style reconstruction trained with SCCFG-1 rollouts, and GuidanceNFT trained with SCCFG-3 rollouts. Reconstruction samples guidance inputs for its regression loss independently of rollout guidance. Each fine-tuned checkpoint is at update 100, and each curve varies evaluation SCCFG. The repetition diagnostic separately compares native-field NFT and GuidanceNFT trained at SCCFG 3.

SiT and Model Guidance settings. Both use class-name PickScore, 1,152 endpoints per update, nine time draws per endpoint, ten-step SDE rollouts, 400 policy updates, learning rate 3×10^{-6} ,

$\beta = 1$, and regularization coefficient 10^{-4} . SiT NFT trains at CFG 1; GuidanceNFT trains at CFG 1.5 after 100 reference initialization updates. Model Guidance uses built-in guidance with external CFG 1 for both methods. The fitted-reference EMA decay is 0.9. Evaluation uses three seeds, each with two samples for every ImageNet class. A few MG updates have no logged reward; the training curves leave them as gaps.

Curve smoothing. Figure 5(a) uses a 30-update trailing mean and panel (b) a 15-update centered mean over the 127 updates logged by both runs (checkpoints are evaluated at update 100); Figure 3(b) uses a Gaussian filter with a five-update standard deviation.

Selection and uncertainty. Table 5 records checkpoint selection. SD3.5 uses fixed checkpoints. SiT NFT selects the best PickScore among updates 100–400 at intervals of 50, choosing update 200 at both CFGs; MG NFT selects update 200 over 400 from a fixed-seed assessment. Both GuidanceNFT image comparisons use update 400. On GSM8K, NFT selects the best Pass@1 over updates 50–400 at intervals of 50, while GuidanceNFT uses update 500.

Table 5: **Checkpoints used for the main comparisons.** NFT is selected by PickScore for SiT/MG and by Pass@1 under the 64-step asynchronous protocol for GSM8K. GuidanceNFT uses the reported fixed checkpoints. Candidate grids differ across methods; reported uncertainty is conditional on the selected checkpoints.

Setting	Retained NFT candidates	NFT selected	GuidanceNFT reported
SiT / ImageNet	100–400, every 50 updates	200	400
Model Guidance / ImageNet	200, 400	200	400
Qwen / GSM8K	50–400, every 50 updates	200	500

Evaluation-seed std uses two seeds for fine-tuned SD3.5 rows and three for SiT/MG; Learned3 sums reward components within each seed before computing std. Paired differences average all images or answers within each prompt or question, then report std across these units. GenEval task-macro comparisons instead use the six task means. The GSM8K main table reports std across question-bootstrap means. Fixed-checkpoint paired comparisons appear in Table 6.

Table 6: **SD3.5 at update 200.** Stds describe paired differences across 553 prompt means for GenEval dense reward, six task means for GenEval task-macro success, and 200 prompt means for Learned3. Each GenEval prompt retains four images; each Learned3 prompt retains ten images across two evaluation seeds. Learned3 is PickScore/26 + CLIPScore + HPSv2.1. These stds describe spread across units; they measure neither training variation nor the uncertainty of the mean.

Metric	Comparator	GuidanceNFT – comparator (mean $\pm$ std)
GenEval dense	NFT, CFG 4	+0.2462 $\pm$ 0.3182
GenEval dense	NFT, CFG 1	+0.0143 $\pm$ 0.1737
GenEval dense	NFT + pretrained empty branch, CFG 4	+0.1043 $\pm$ 0.2719
GenEval task macro	NFT, CFG 4	+0.3053 $\pm$ 0.1353
GenEval task macro	NFT, CFG 1	+0.0269 $\pm$ 0.0256
GenEval task macro	NFT + pretrained empty branch, CFG 4	+0.1337 $\pm$ 0.0406
Learned3 sum	NFT, CFG 4	+0.0176 $\pm$ 0.0357
Learned3 sum	NFT, CFG 1	−0.0019 $\pm$ 0.0398
Learned3 sum	NFT + pretrained empty branch, CFG 4	−0.0049 $\pm$ 0.0408

D ADDITIONAL SD3.5 RESULTS

Metrics and uncertainty. Table 1 evaluates GenEval-, OCR-, and Learned3-trained checkpoints at update 200 using ten-step training rollouts and 40-step inference. NFT trains at CFG 1 and is evaluated at CFG 1/4; GuidanceNFT trains and is evaluated at CFG 4. Train reward is the optimized objective on its on-domain test set. The seven evaluator columns use separately generated samples; gray cells mark evaluators whose reward is also the training objective. All columns are held out with respect to prompts: evaluation prompts (the GenEval and OCR test splits and DrawBench) are

disjoint from the training prompts (the GenEval, OCR, and Pick-a-Pic training splits). All scores are higher-is-better, and bold includes ties at the displayed precision.

Every fine-tuned row averages two evaluation seeds equally; pretrained rows use one seed. Table 7 reports the corresponding means and sample stds. Each seed uses 2,212 GenEval images, 1,018 OCR images, 2,048 Pick-a-Pic images for Learned3 Train reward, and 1,000 DrawBench images for the five general-quality metrics. Learned3 Train reward sums raw PickScore/26, HPSv2.1, and CLIPScore within each seed.

Table 7: **SD3.5 rewards and evaluation-seed variation.** Each cell shows the mean on the first line and $\pm$ sample std on the second, using two evaluation seeds per fixed update-200 checkpoint. Train reward uses on-domain test evaluations; the remaining metrics follow Table 1. All evaluations use 40 sampling steps at 512×512 ; PickScore is unscaled. Bold marks the best mean within each objective. These stds measure sampling variation, not training variability.

Objective Method		Train CFG reward	GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	
GenEval	NFT	1	0.900 ± 0.0024	0.855 ± 0.0077	0.434 ± 0.0059	21.70 ± 0.0021	0.265 ± 0.0002	0.234 ± 0.0001	4.886 ± 0.0132	0.271 ± 0.0178
		4	0.657 ± 0.0139	0.565 ± 0.0003	0.624 ± 0.0144	21.19 ± 0.0061	0.271 ± 0.0014	0.216 ± 0.0006	4.773 ± 0.0116	0.148 ± 0.0224
GenEval	GuidanceNFT	4	0.914 ± 0.0015	0.877 ± 0.0013	0.674 ± 0.0016	21.83 ± 0.0099	0.277 ± 0.0004	0.250 ± < 0.0001	4.947 ± 0.0128	0.682 ± 0.0153
OCR	NFT	1	0.971 ± 0.0014	0.035 ± 0.0042	0.971 ± 0.0008	19.61 ± 0.0118	0.208 ± 0.0009	0.137 ± 0.0006	4.208 ± 0.0036	-1.679 ± 0.0183
		4	0.192 ± 0.0067	0.265 ± 0.0064	0.235 ± 0.0070	20.21 ± 0.0188	0.242 ± 0.0001	0.188 ± 0.0008	4.596 ± 0.0051	-0.919 ± 0.0642
OCR	GuidanceNFT	4	0.975 ± 0.0010	0.488 ± 0.0099	0.974 ± 0.0009	21.83 ± 0.0293	0.274 ± 0.0003	0.256 ± 0.0022	5.120 ± 0.0072	0.425 ± 0.0426
Learned3	NFT	1	1.519 ± 0.0003	0.671 ± 0.0061	0.497 ± 0.0075	23.51 ± 0.0150	0.284 ± 0.0001	0.332 ± 0.0008	5.606 ± 0.0135	1.397 ± 0.0092
		4	1.495 ± 0.0002	0.701 ± 0.0026	0.579 ± 0.0001	22.99 ± 0.0109	0.290 ± 0.0001	0.326 ± 0.0002	5.397 ± 0.0034	1.414 ± 0.0005
Learned3	GuidanceNFT	4	1.528 ± 0.0003	0.630 ± 0.0003	0.692 ± 0.0011	23.24 ± 0.0079	0.292 ± 0.0002	0.332 ± 0.0001	5.653 ± 0.0003	1.448 ± 0.0020

GenEval Train reward uses the native dense reward, which grants partial credit and retains valid negative values. The GenEval evaluator column uses image-micro averages of saved success flags. The benchmark’s alternative task-macro aggregation weights its six tasks equally: on one evaluation seed it gives 0.8547/0.5762 for NFT at CFG 1/4 and 0.8815 for GuidanceNFT at CFG 4. Success follows the official all-condition rule (Ghosh et al., 2023).

Where the GenEval gains arise. The task breakdown in Table 8 shows broad improvement at CFG 4. Among 553 paired prompts, GuidanceNFT increases the number of successful images for 285, decreases it for 25, and leaves 243 unchanged. Across image pairs, 754 failures become successes and 64 successes become failures. Each prompt contributes its four-image mean to the within-task std of paired differences.

Table 8: **GenEval success by task.** Results use update 200, evaluation CFG 4, one evaluation seed, and four images per prompt. Differences and sample stds are in percentage points across paired prompt means within each task. These stds describe prompt variation at fixed checkpoints.

Task	Prompts	NFT (%)	GuidanceNFT (%)	Difference $\pm$ std
color_attr	100	22.25	74.25	+52.00 $\pm$ 39.99
colors	94	61.70	82.98	+21.28 $\pm$ 39.99
counting	80	58.75	92.19	+33.44 $\pm$ 39.17
position	100	56.50	87.00	+30.50 $\pm$ 46.00
single_object	80	87.19	99.06	+11.88 $\pm$ 26.38
two_object	99	59.34	93.43	+34.09 $\pm$ 40.00

D.1 QUALITATIVE COMPARISONS

Figure 4 combines four Learned3 DrawBench samples at update 300 with four OCR online-evaluation previews at update 200. Table 1 reports all objectives at a common update 200, whereas the Learned3 examples use the final checkpoint of that run. Learned3 candidates were ranked by GuidanceNFT’s combined reward, $\text{PickScore}/26 + \text{HPSv2.1} + \text{CLIPScore}$, among paired samples for which both NFT settings scored below their respective medians and GuidanceNFT exceeded both. Repeated prompts were removed from the displayed shortlist; the first three distinct prompts and one further shortlisted prompt were selected for the figure. The OCR cases were selected from the saved preview grid. Both groups use 40 inference steps, with noise paired across methods within each column.

Figures 9 and 10 extend these comparisons to six prompts per objective. The Learned3 examples compare object, spatial, and text alignment; the OCR examples compare target-text rendering and scene content.

The full prompts for Figure 4, from left to right, are reproduced verbatim below. Figure headings are short summaries rather than the full generation instructions. Learned3 images are 512×512 outputs; OCR images are saved 256×256 previews, enlarged by nearest-neighbor pixel replication.

1. A triangular orange picture frame. An orange picture frame in the shape of a triangle.
2. Hovering cow abducting aliens.
3. An umbrella on top of a spoon.
4. A bird scaring a scarecrow.
5. A close-up of a sleek smartwatch on a wrist, the screen displaying "Step Goal Achieved" with a celebratory animation, set against a blurred cityscape at dusk, capturing the moment of accomplishment.
6. A futuristic urban street at night, a cyborg shop sign glowing "Upgrades Available" in neon blue, reflecting off wet pavements, with cybernetic pedestrians and sleek, futuristic buildings in the background.
7. A high-altitude mountain summit with a wooden signpost clearly marked "Elevation 8000 Feet", surrounded by rocky terrain and a backdrop of distant, snow-capped peaks under a clear blue sky.
8. A realistic photo of a tech campus courtyard at night, featuring a glowing "AI Training Zone" hologram floating in the center, surrounded by futuristic buildings and greenery, with soft ambient lighting enhancing the futuristic atmosphere.

Figure 11 displays selected GenEval-specialist examples at update 200. Matching the seed controls the noise draw within a prompt. NFT CFG-4 samples show less texture and shading in these examples; the GuidanceNFT samples remain stylized.

E EMPIRICAL REFERENCE DIAGNOSTICS AND TRAINING COSTS

Diagnostic design. We test whether a separate predictor better estimates the mean velocity of guided endpoints. For each frozen checkpoint, an auxiliary FM model is fitted for exactly 100 AdamW updates on 384 endpoints, with 96 held out: eight fitting and two held-out endpoints for each of 48 classes (MG) or category-balanced prompts (SD3.5). Classes and prompts are shared between fitting and evaluation. The auxiliary EMA, denoted C_{FM100} , is a finite-fit reference. Each comparison uses identical held-out endpoints, noise levels, and Gaussian draws for both fields. We report two complementary statistics:

$$D_q(F) = R_q(F) - R_q(C_{\text{FM100}}), \quad G_q(F) = \mathbb{E}_q \|F - C_{\text{FM100}}\|^2/d, \quad (32)$$

where d is the field dimension. The paired risk difference D_q cancels irreducible conditional variance and compares reference MSEs (Equation 29); G_q measures field disagreement. Standard deviations use the 48 class or prompt means, averaging paired endpoints within each unit.

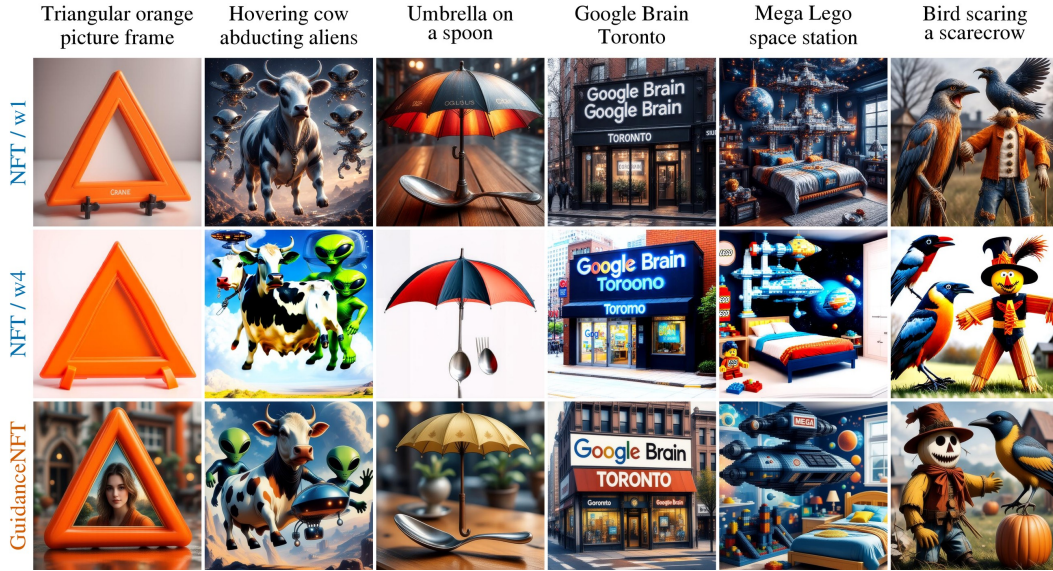


Figure 9: **Learned3 fine-tuning retains more requested content with GuidanceNFT.** Six selected SD3.5-M examples compare object, spatial, and text alignment. Columns show different prompts; rows show NFT at CFG 1, NFT at CFG 4, and GuidanceNFT at CFG 4, from top to bottom.

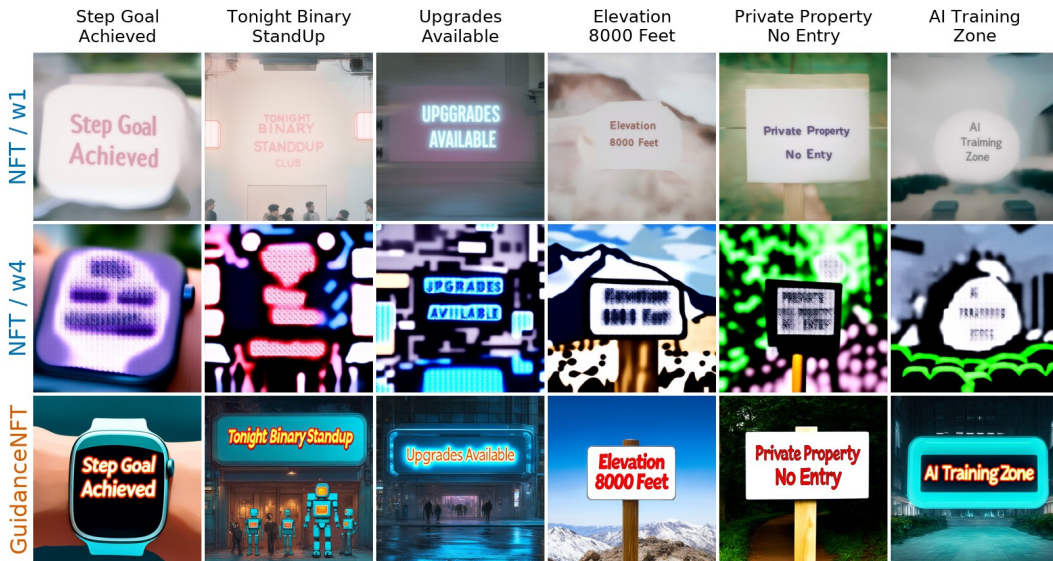


Figure 10: **GuidanceNFT combines readable text with the requested scene.** Six selected SD3.5-M OCR examples compare NFT at CFG 1 (top), NFT at CFG 4 (middle), and GuidanceNFT at CFG 4 (bottom). NFT often omits scene content, while adding CFG disrupts text legibility. Column headings give the target text.

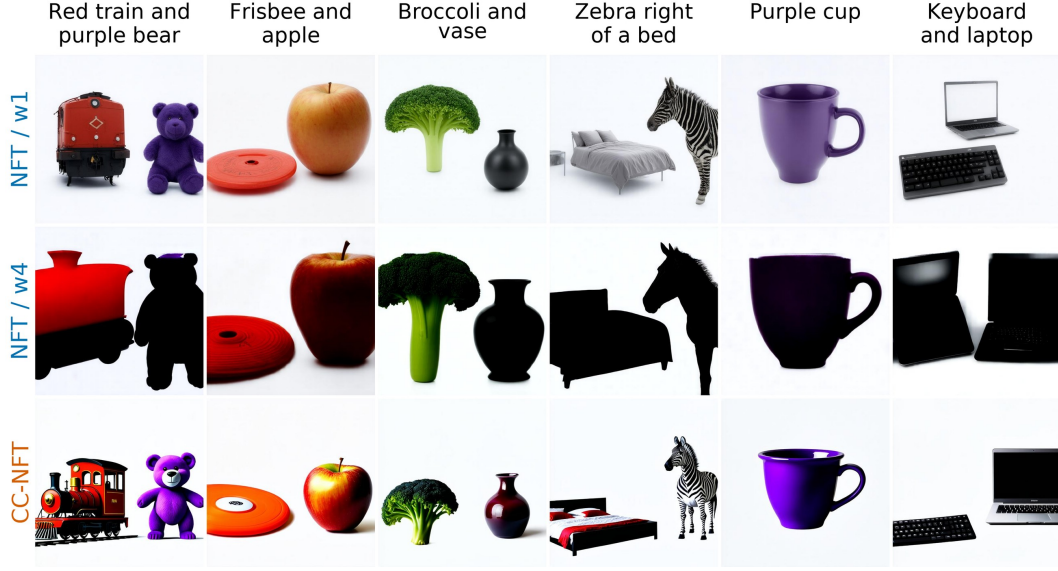


Figure 11: **GuidanceNFT preserves color and shading during GenEval fine-tuning.** Selected SD3.5-M examples compare NFT at CFG 1 (top), NFT at CFG 4 (middle), and GuidanceNFT at CFG 4 (bottom). Adding CFG to NFT darkens images and obscures detail; GuidanceNFT retains more visible color and shading. Column headings summarize the prompts.

Model Guidance. NFT at update 200 and GuidanceNFT at update 400 each generate their own population with ten-step Euler-Maruyama rollouts and external CFG 1. Each auxiliary predictor starts from the corresponding native policy, with learning rate 3×10^{-6} , effective batch 96, and EMA decay 0.9. Evaluation averages 11 fixed noise levels from 0.1 to 0.9 equally.

The native-minus-FM100 risk is 0.6969 ± 0.7250 on NFT endpoints and 0.0712 ± 0.0201 on GuidanceNFT endpoints (class std). On the latter population, the saved GuidanceNFT reference has risk difference -0.00715 ± 0.00396 , outperforming FM100. Directly comparing the native field with the saved reference gives 0.07835 ± 0.02287 , supporting separate reference estimation on the same rollout population (Figure 12).

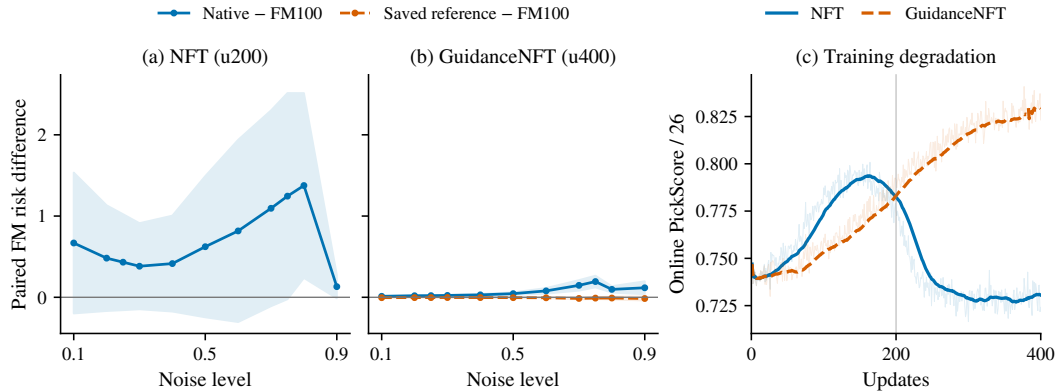


Figure 12: **A fitted reference better represents Model Guidance rollouts.** (a–b) On NFT and GuidanceNFT samples, fitted references have lower flow-matching error than the native field; zero denotes a separately fitted reference. (c) GuidanceNFT continues to improve reward as NFT declines; the vertical line marks update 200. Bands show mean $\pm$ one std across classes.

SD3.5. We evaluate the three five-step CFG-4 runs in Figure 3(b) at updates 0, 50, and 100. Each saved old-policy adapter generates its own deterministic DPM2 endpoints. The tested references

are the old conditional field, the analytic CFG-4 field, and the saved reference EMA, respectively. Prompts and initial-noise seeds are paired across runs; each run retains its own endpoints and independently fitted FM100 reference. The auxiliary models start from their respective old conditional fields and use learning rate 3×10^{-4} , effective batch 8, zero weight decay, gradient clipping at 1, and dynamic EMA capped at 0.9. Fitting draws fresh Gaussian noise and uniform continuous times while the policy and saved training reference remain frozen.

Evaluation uses $\sigma \in \{0.1, 0.2, \dots, 0.9\}$ with equal weights. At initialization the learned reference equals the uncorrected conditional field; differences between their initial diagnostic gaps arise from separate auxiliary fits. At $\sigma = 0.5$, between updates 50 and 100, field disagreement grows from 0.156 to 0.219 without correction and from 5.43 to 148.06 with analytic correction, while GuidanceNFT remains near 0.008 (Figure 3(c)). The corresponding noise-averaged gaps change from 0.1999 to 0.2922, from 5.6244 to 127.3269, and from 0.01135 to 0.01183. Learned correction thus maintains agreement with its rollout reference as the two alternatives diverge (Figure 13).

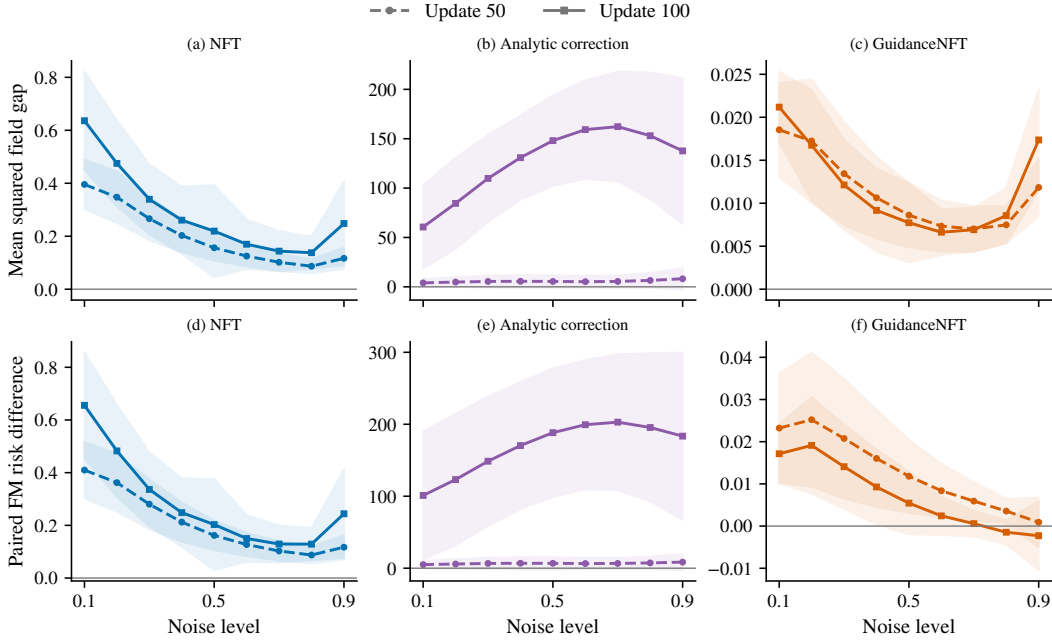


Figure 13: GuidanceNFT limits reference mismatch as training progresses. Columns compare NFT, analytic correction, and GuidanceNFT on SD3.5-M. Top: disagreement with a fitted rollout reference across noise levels. Bottom: flow-matching error relative to that reference; negative values indicate lower error. From update 50 (dashed) to 100 (solid), mismatch grows for NFT and analytic correction but remains small for GuidanceNFT. Bands show mean $\pm$ one std across prompts.

The same runs show a corresponding separation in reward trajectories. First-ten to last-ten mean rewards over updates 0–99 fall from 0.3085 to 0.0186 with analytic correction and from 0.3251 to 0.0953 without correction, but rise from 0.3198 to 0.4977 with learned correction. A ten-update trailing mean falls below 0.1 at updates 41 and 97 for the two failing runs, respectively. These trace summaries support the association between reference mismatch and degradation.

Training cost. Table 9 measures intervals between consecutive completed policy updates, including rollout and reward scoring. Median GenEval iteration time is 387.9 seconds for NFT and 672.4 seconds for GuidanceNFT, a ratio of 1.73; the Learned3 ratio is 1.77. Known evaluation, checkpoint, and restart boundaries are excluded.

Reference initialization and refresh add training computation. The estimator is discarded at deployment, adding no inference passes.

Table 9: **Recorded SD3.5 training costs.** Host seconds on two H200 GPUs, FP16, microbatch 24 per GPU, and 1,152 rollout endpoints per update. n counts eligible complete iterations; IQR is the 25th–75th percentile. Ratios divide iteration medians by NFT within each task. The total includes rollout, reward scoring, reference fitting, and policy updates; component medians need not sum to the total median. Known evaluation/checkpoint and restart boundaries are excluded. Unmarked long pauses remain (2/3 GenEval and 1/0 Learned3 intervals exceed five times the respective median for NFT/GuidanceNFT). Reference initialization is excluded. These are recorded per-iteration latencies, not total training costs; deployment requires no auxiliary-reference pass.

Task	Method	Rollout/ n reward		Reference	Policy	Iteration median [IQR]	Ratio
GenEval	NFT	193	113.6	0.0	274.3	387.9 [387.2, 388.7]	1.00×
GenEval	GuidanceNFT	184	182.2	150.6	339.7	672.4 [671.1, 674.3]	1.73×
Learned3	NFT	94	92.5	0.0	274.5	367.3 [366.7, 367.8]	1.00×
Learned3	GuidanceNFT	96	160.5	150.5	339.2	650.2 [649.9, 650.6]	1.77×

F SiT AND MODEL GUIDANCE EVALUATIONS

Table 10 reports fixed-checkpoint PickScore for SiT and Model Guidance, complementing the training trajectories in the main text.

Table 10: **Unscaled PickScore ($\uparrow$), mean $\pm$ sample std.** Each checkpoint uses three evaluation seeds and 6,000 images. NFT uses its best observed checkpoint (200); GuidanceNFT uses 400. MG has external CFG 1. Bold marks the best mean. Stds are conditional on the selected checkpoints and do not measure training variation.

Method	SiT, CFG 1	SiT, CFG 1.5	Model Guidance
Pretrained	18.849 $\pm$ 0.012	19.287 $\pm$ 0.015	19.239 $\pm$ 0.003
NFT	20.161 $\pm$ 0.019	20.476 $\pm$ 0.025	20.010 $\pm$ 0.023
GuidanceNFT	21.465 $\pm$ 0.006	21.579 $\pm$ 0.013	21.580 $\pm$ 0.018

Each SiT checkpoint–CFG coordinate contains 6,000 images over three evaluation seeds. Table 11 reports unscaled PickScore over the retained checkpoint sequence, complementing the normalized trajectories in Figure 3(a).

Pretrained-empty SiT control. Replacing NFT’s empty-condition prediction with the pretrained one gives a modest recovery but does not prevent its late decline. At update 400, PickScore/26 is 0.7028 for NFT, 0.7101 with substitution, and 0.8299 for GuidanceNFT. NFT peaks at 0.7875 at update 200; GuidanceNFT improves through update 400. This inference-only control uses CFG 1.5, SiT’s default guidance on the first three latent channels, ten SDE steps, batch size eight, and the same three evaluation seeds at updates 100–400 every 50 updates.

Selected PickScore examples. Figure 6 compares NFT at update 200 with GuidanceNFT at update 400. The left panel uses SiT with 250 sampling steps; the right uses Model Guidance with 1,000 steps. SiT uses CFG 1 for NFT and CFG 1.5 for GuidanceNFT; Model Guidance uses external CFG 1. Noise is paired across methods within each column. Side labels report unscaled PickScore, mean $\pm$ sample std over three evaluation seeds (6,000 images), under the separate ten-step benchmark.

G GSM8K REASONING PROTOCOLS

The starting checkpoints are from the Qwen-based ELF reasoning model, an ELF-B variant used in CEDR (Cheng, 2026), built on ELF (Hu et al., 2026). It uses Qwen-derived (Yang et al., 2025) feature groups and a learned prompt encoder; inference does not execute the Qwen teacher Transformer.

Table 11: **SiT-XL/2 unscaled PickScore ($\uparrow$) at every retained checkpoint** (columns 100–400). Each cell shows the mean with $\pm$ sample std beneath it across three evaluation-seed means, conditional on a fixed checkpoint. Each seed covers all 1,000 ImageNet classes with two images per class (2,000 images per seed; 6,000 images per entry), using ten-step SDE sampling. NFT trains with CFG-1 rollouts. GuidanceNFT first fits its reference for 100 updates and then trains with CFG-1.5 rollouts. Bold marks the best checkpoint within each method and evaluation-CFG row.

Method	CFG	100	150	200	250	300	350	400
NFT		19.392	19.791	20.161	19.419	19.013	18.477	18.317
	1	± 0.025	± 0.007	± 0.019	± 0.023	± 0.009	± 0.003	± 0.003
	1.5	19.814	20.179	20.476	19.747	19.094	18.390	18.273
		± 0.006	± 0.010	± 0.025	± 0.026	± 0.006	± 0.003	± 0.004
GuidanceNFT		19.442	19.880	20.375	20.853	21.157	21.345	21.465
	1	± 0.016	± 0.015	± 0.016	± 0.003	± 0.009	± 0.008	± 0.006
	1.5	19.924	20.319	20.728	21.097	21.325	21.474	21.579
		± 0.005	± 0.003	± 0.008	± 0.003	± 0.007	± 0.016	± 0.013

CEDR’s NFT implementation. CEDR’s NFT adaptation uses the following detached correction for current, old, and regularization-target roles p :

$$\tilde{v}_p = v_p^{\text{main}} - \text{sg}[b_{\text{sc}}(1 - g^{-1})(v_p^{\text{main}} - v_p^0)], \quad b_{\text{sc}} \in \{0, 1\}. \quad (33)$$

Here v_p^0 uses zero answer self-conditioning, while the main pass uses the detached bootstrap prediction when $b_{\text{sc}} = 1$; both retain the question prompt. The loss samples $b_{\text{sc}} \sim \text{Bernoulli}(0.5)$ and $\log(1 + g) \sim \text{Uniform}(\log 1.5, \log 6)$, independently of the SCCFG-2 rollout scale. The correction retains the current main-pass Jacobian. The corrected fields enter NFT with $\beta = 1$, NFT coefficient 5, and a squared-velocity regularization penalty with coefficient 0.1, averaged over latent coordinates and valid response tokens. Section 5.1 and Appendix I of Cheng (2026) give the full objective and gradient routing. Our native-field NFT and GuidanceNFT runs do not use this reconstruction. When $b_{\text{sc}} = 1$, the corrected value is $v_p^{\text{main}}/g + (1 - g^{-1})v_p^0$, the base prediction implied by ELF’s SCCFG relation (Hu et al., 2026).

Evaluation and training protocols. The main comparison evaluates all 1,319 GSM8K questions with 64 steps, SCCFG 2, feature-group clocks (2.5, 2, 1.5) under the asynchronous time sampling of Cheng (2026), and eight evaluation seeds. The pretrained row uses ELF/prompt EMA 0.9999/0.9999; all trained rows use 0.99/0.99. Table 2 reports Pass@1/2/4/8 with problem-bootstrap std: 4,000 resamples of the 1,319 questions with replacement, preserving all eight answers within each sampled question. Each replicate averages unbiased question-level Pass@ k estimates; std uses the sample denominator. All rows are recomputed from the saved answer cohorts.

Our NFT and GuidanceNFT runs use 32-step SCCFG-2 training rollouts and native main-pass velocities, without velocity reconstruction. The CEDR baseline retains the question prompt when zeroing answer self-conditioning. Its training also includes one reference solution per 15 generated answers, removes all-correct groups, and draws eight logit-normal times per endpoint. Our runs use generated endpoints only, retain groups containing both correct and incorrect answers, and use 31 draws from rollout time steps. Pretrained denotes the starting checkpoint before NFT fine-tuning.

Under the main protocol, NFT’s Pass@1 is 40.03, 40.59, 41.21, 41.68, 39.63, 27.42, 17.48, and 12.23% at updates 50–400, respectively. The main table reports its best checkpoint (update 200), so GuidanceNFT’s gains are measured against NFT before its late degradation. Paired differences, computed per question and resampled with the same question bootstrap, are $+1.35 \pm 0.38$, $+1.13 \pm 0.48$, $+1.46 \pm 0.64$, and $+1.59 \pm 0.95$ points over NFT and $+0.86 \pm 0.49$, $+1.30 \pm 0.57$, $+1.84 \pm 0.72$, and $+2.35 \pm 1.00$ points over CEDR at $k = 1, 2, 4, 8$ (mean $\pm$ bootstrap std).

Synchronous SCCFG-2 checkpoint sweep. The sweep in Figure 5(c) and Table 12 evaluates the same runs with 32 steps, synchronous clocks (1, 1, 1), and three seeds. NFT reaches 36.57% at update 200 before falling to 13.65% at update 400. GuidanceNFT reaches 35.38% at update 450 and 35.28% at update 500. NFT stops after its training rollouts produce only incorrect answers,

so it has no checkpoints at updates 450 and 500. Checkpoint selection follows the conventions in Appendix C.

Table 12: **Synchronous SCCFG-2 evaluation.** The sweep uses 1,319 GSM8K questions, 32 steps, synchronous clocks (1, 1, 1), external CFG 1, ELF/prompt EMA 0.99/0.99, and three evaluation seeds. Each cell shows mean accuracy (%) with $\pm$ sample std beneath it, across evaluation seeds for one training run. Bold marks each method's largest observed mean. NFT training stopped after its rollouts produced only incorrect answers, so update 400 is its last checkpoint (dashes). This protocol differs from the 64-step asynchronous main-table evaluation.

Update	50	100	150	200	250	300	350	400	450	500
NFT	32.12 ± 0.34	34.22 ± 0.46	34.77 ± 0.38	36.57 ± 0.31	35.00 ± 0.88	29.54 ± 1.18	19.84 ± 1.35	13.65 ± 0.35	—	—
GuidanceNFT	31.39 ± 0.27	31.99 ± 0.20	33.21 ± 0.35	34.37 ± 0.12	34.14 ± 0.12	35.08 ± 0.62	34.60 ± 0.49	35.18 ± 0.30	35.38 ± 0.24	35.28 ± 0.29