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

Anonymous authors
Paper under double-blind review

ABSTRACT

Guidance and reward fine-tuning improve diffusion generation in complementary ways. Combining the two, however, can undermine both benefits: applying classifier-free guidance after DiffusionNFT (NFT) can reduce the optimized reward, while incorporating guidance during NFT fine-tuning may cause collapse. We identify a mismatch in NFT's regression center as a source of this instability. The center must match the conditional mean velocity obtained by re-noising generated samples; using a mismatched center introduces a bias that can rotate or reverse the reward-driven update. We propose GuidanceNFT, which learns this center online by flow matching on guided samples and corrects both NFT regression terms while preserving the policy's reward-driven displacement. The method supports explicit and intrinsic guidance without additional inference passes. On SD3.5-M, GuidanceNFT improves the optimized reward over standard NFT across three training objectives and outperforms NFT on all seven evaluation metrics for GenEval and OCR specialization. In particular, OCR reward increases from 0.971 to 0.975 while held-out GenEval success rises from 0.035 to 0.488, indicating improved prompt fidelity beyond the training objective. Across ImageNet generation, OpenWebText, and GSM8K reasoning, GuidanceNFT improves generation quality and task performance under intrinsic guidance. These results show that online center correction enables effective reward fine-tuning of guided diffusion models.

1 INTRODUCTION

Guidance and reward fine-tuning improve diffusion generation in complementary ways. Classifier-free guidance (CFG) strengthens prompt alignment by combining conditional and empty-condition predictions (Ho & Salimans, 2022), while reward fine-tuning adapts generation to a target objective. DiffusionNFT (NFT) (Zheng et al., 2026) performs this adaptation through regression on positive and negative feedback, without reward gradients or trajectory likelihoods. Combining guidance with NFT could therefore improve task-specific performance while retaining broader generation quality (Figure 1).

This broader quality is difficult to capture with a task reward alone. For example, an OCR-specialized NFT model can render the requested words while omitting the surrounding scene (Figure 3). Simply adding CFG after fine-tuning does not resolve this limitation: in our SD3.5-M experiments, it reduces the optimized reward for all three training objectives. A natural alternative is to incorporate guidance during fine-tuning, so that reward feedback reflects the samples used at deployment. Yet standard NFT can then degrade or collapse, even when sampling and training use the same guidance setting. This failure occurs with both explicit CFG and *intrinsic guidance*, where guidance is built into the model's prediction (Figures 2 and 4).

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

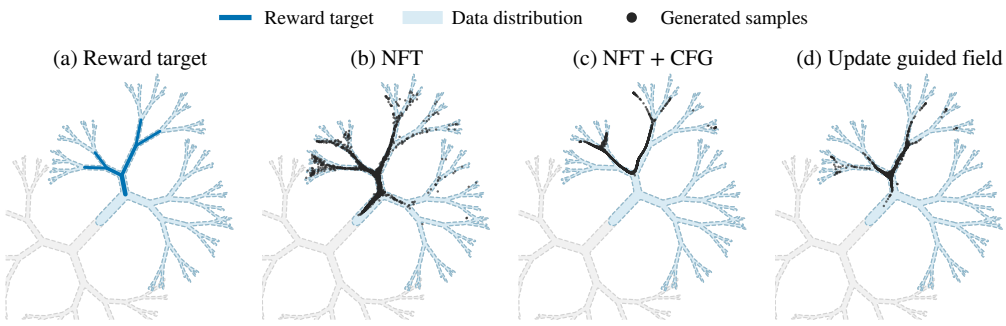


Figure 1: **Intuition for updating a guided field.** (a) Blue branches mark high-reward regions of the tree-shaped data distribution. (b) NFT alone only partly concentrates samples in these regions. (c) Adding guidance after NFT pushes more samples off the tree. (d) Updating the guided field better retains the tree structure while moving samples toward the reward target. This toy illustrates the desired behavior; implementation details are in Appendix B.

The purpose of this paper is to enable effective NFT fine-tuning of guided diffusion models. We identify a mismatch in NFT's regression objective that can undermine reward optimization under guidance. NFT uses a shared reference prediction, or *center*, to learn from higher- and lower-reward samples. For these two regression terms to recover the intended update, the center must match the conditional mean velocity obtained by re-noising the generated samples. The guided prediction used to generate those samples need not satisfy this requirement, even when it combines exact component predictors (Bradley & Nakkiran, 2025; Choi et al., 2026). We derive how this mismatch introduces a bias that can rotate or reverse the reward-driven update. Our experiments link larger measured center gaps to faster collapse and show that correcting the center sustains reward improvement.

Building on this analysis, we propose *GuidanceNFT*, which estimates the regression center separately from the policy being optimized. An auxiliary predictor learns the center by flow matching on guided samples and is refreshed as their distribution changes. We use its difference from NFT's existing center to correct both regression terms, holding the correction fixed during each policy update. NFT can then optimize the guided model using a center fitted to its own samples. The same construction supports explicit and intrinsic guidance, and the auxiliary predictor is discarded at inference.

GuidanceNFT improves both task performance and generation quality. On SD3.5-M, it exceeds NFT with and without CFG in the optimized reward for all three training objectives. For GenEval and OCR specialization, the gains extend to all seven evaluation metrics. In particular, OCR reward increases from 0.971 to 0.975 while held-out GenEval success rises from 0.035 to 0.488, illustrating improved prompt fidelity beyond the training objective. With intrinsic guidance, GuidanceNFT improves ImageNet generation, achieves a better perplexity–entropy tradeoff on OpenWebText, and outperforms NFT's best retained checkpoint at every reported Pass@k on GSM8K.

Our main contributions are:

1. **A diagnosis of NFT instability under guidance.** We identify a mismatch between NFT's reference prediction and the distribution of samples produced under guidance. We derive how this mismatch biases the reward-driven update and provide empirical evidence linking it to performance degradation and training collapse.
2. **An online correction for guided reward fine-tuning.** We introduce GuidanceNFT, which learns the required regression center from guided samples and corrects the NFT objective as the policy evolves. The method supports explicit and intrinsic guidance without additional inference passes.
3. **Improved task performance and overall generation quality.** GuidanceNFT improves task rewards and overall generation quality on SD3.5-M. It also enables NFT to improve models with intrinsic guidance, including Model Guidance and self-conditioning CFG (SC-CFG). Experiments on ImageNet, OpenWebText, and GSM8K further demonstrate performance gains across models, tasks, and guidance mechanisms.

2 BACKGROUND

2.1 DIFFUSIONNFT

DiffusionNFT (Zheng et al., 2026) learns from the contrast between higher- and lower-reward samples. Given a rollout distribution q and processed reward weights $r \in [0, 1]$, it defines positive and negative populations $q^+ \propto rq$ and $q^- \propto (1 - r)q$. Their conditional mean velocities provide the optimal regression targets for the two populations. Rather than fitting these targets with separate models, NFT parameterizes both predictions using a trainable policy v_θ and a frozen old policy v^{old}:

$$v_\theta^+ = (1 - \beta)v^{\text{old}} + \beta v_\theta, \qquad v_\theta^- = (1 + \beta)v^{\text{old}} - \beta v_\theta. \tag{1}$$

These predictions move in opposite directions around v^{old}, with $\beta > 0$ controlling their separation. Both are evaluated at the same noisy sample (x_t, c, t). Writing v for its forward-process velocity target, NFT minimizes

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

The shared parameterization turns positive and negative feedback into a single policy update, using only generated endpoints and their rewards. Appendix A specifies the forward process and normalized populations.

2.2 CLASSIFIER-FREE AND BUILT-IN GUIDANCE

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

$$v_w = v_\varnothing + w(v_c - v_\varnothing) = wv_c + (1 - w)v_\varnothing. \tag{3}$$

Here $w = 1$ gives conditional-only sampling. The guided field determines the generated endpoints, but its intermediate predictions need not equal the mean velocities obtained by re-noising those endpoints. This distinction holds even for exact component predictors (Bradley & Nakkiran, 2025; Chidambaram et al., 2024) and becomes central when using guided samples in NFT.

Model Guidance. Model Guidance (MG) (Tang et al., 2025) incorporates guidance into the model's output, requiring one prediction per inference step. In velocity coordinates, it trains M_c toward $v + \lambda\,\text{sg}(M_c - M_\varnothing)$. For a fixed $0 \le \lambda < 1$, its ideal fixed point has the CFG form

$$M_c = w_{\text{eff}} C_{\text{data}} + (1 - w_{\text{eff}})M_\varnothing, \qquad w_{\text{eff}} = \frac{1}{1 - \lambda}, \tag{4}$$

where C_{data} is the unweighted conditional mean velocity of the pretraining data. The native prediction therefore represents a guided field even without an external CFG operation.

Self-conditioning CFG (SCCFG). ELF (Hu et al., 2026) learns guidance from its own clean-embedding prediction. The same network produces a self-conditioned field S_θ^w and a field $S_{\theta,\varnothing}^w$ 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 at inference. The training target incorporates guidance through

$$v_{\text{target}} = v + \left(1 - \frac{1}{w}\right)\text{sg}(S_\theta^w - S_{\theta,\varnothing}^w). \tag{5}$$

Its corresponding CFG form is $S_\theta^w = wB_\theta^w + (1 - w)S_{\theta,\varnothing}^w$, where B_θ^w is the algebraically reconstructed base prediction (Appendix A.6).

Both MG and SCCFG thus expose NFT to a guided native prediction. Setting external CFG to 1 preserves this built-in guidance, while changing the internal scale changes the rollout distribution itself. To update the original guided distribution with NFT, we therefore need the center induced by its own re-noised endpoints.

3 GUIDANCENFT

Applying NFT to guided rollouts requires its regression objective to recover the reward-driven change in velocity. We first derive this change and show how an inaccurate center distorts it. This identifies both the quantity to estimate and the correction needed for guided fine-tuning.

3.1 Center Exactness Determines the NFT Update

Fix the guided rollout law $q_{\mathrm{CFG}}(\cdot \mid c)$ and processed reward weights $r \in [0, 1]$ for one policy update. Here q_{CFG} also covers built-in guidance. At a re-noised state $z = (x_t, c, t)$, let $\alpha = \mathbb{E}_{q_{\mathrm{CFG}}}[r \mid z] \in (0, 1)$. The optimal regression fields for the positive and negative populations are

$$v^+ = \frac{\mathbb{E}_{q_{\mathrm{CFG}}}[rv \mid z]}{\alpha}, \qquad v^- = \frac{\mathbb{E}_{q_{\mathrm{CFG}}}[(1-r)v \mid z]}{1-\alpha}, \qquad \Delta = v^+ - v^-. \tag{6}$$

Thus Δ separates the velocity associated with higher-reward samples from that associated with lower-reward samples. Under Gaussian noising, v^+ and v^- are affine transforms of the corresponding noised-population scores (Appendix A). NFT aims to learn a displacement along this reward-improving direction without fitting the two fields separately.

To determine when NFT recovers this direction, write the policy displacement as $d = v_\theta - v^{\mathrm{old}}$ and center its implicit predictions at a frozen field C. Standard NFT uses $C = v^{\mathrm{old}}$. Conditioning its squared loss on z gives, up to a constant in d,

$$\ell_z(C; d) = \alpha \|C + \beta d - v^+\|^2 + (1-\alpha)\|C - \beta d - v^-\|^2. \tag{7}$$

Differentiating with respect to d yields $\beta d^* = \alpha(v^+ - C) - (1-\alpha)(v^- - C)$. The positive term pulls toward v^+; the negative term pushes away from v^-. To recover NFT's prescribed improvement from both branches, their weighted residuals must satisfy $\alpha(v^+ - C) = (1-\alpha)(C - v^-) = \alpha(1-\alpha)\Delta$. Solving either equality gives the unique matched center:

$$\bar{v}_{q_{\mathrm{CFG}}} = \mathbb{E}_{q_{\mathrm{CFG}}}[v \mid z] = \alpha v^+ + (1-\alpha)v^-. \tag{8}$$

At this center the two residuals are $v^+ - \bar{v}_{q_{\mathrm{CFG}}} = (1-\alpha)\Delta$ and $v^- - \bar{v}_{q_{\mathrm{CFG}}} = -\alpha\Delta$, so both terms contribute in the direction of Δ.

For an arbitrary center, substituting $C = \bar{v}_{q_{\mathrm{CFG}}} + e$ instead gives the exact decomposition

$$\boxed{d^* = \underbrace{[2\alpha(1-\alpha)/\beta]\Delta}_{\text{reward-driven displacement}} + \underbrace{[(1-2\alpha)/\beta]e}_{\text{center-induced bias}}.} \tag{9}$$

Thus the unique matched center recovers NFT's intended displacement by making the two weighted branches contribute equally to it. An error parallel to Δ can change its magnitude or reverse its sign; an orthogonal component can rotate the update away from the reward direction. Matching the center removes this bias across local reward balances, preventing the objective from learning a spurious correction alongside the reward signal. The identity holds for the unrestricted optimum of the fixed-rollout surrogate; Appendix A.4 also derives the effect of center error on the parameter gradient before convergence.

3.2 Why Guidance Can Introduce a Center Gap

A guided sampler generates endpoints by integrating a velocity field. NFT then draws fresh noisy states from those endpoints. The sampler's intermediate distributions need not coincide with the forward-noised endpoint distributions, so its velocity need not supply NFT's center. For the linear noising path $x_t = tx + (1-t)\epsilon$, this distinction has the explicit form

$$\bar{v}_{q_{\mathrm{CFG}}}(x_t, c, t) = \frac{x_t}{t} + \frac{1-t}{t}\nabla_{x_t} \log q_{\mathrm{CFG},t}(x_t \mid c), \qquad 0 < t < 1, \tag{10}$$

where $q_{\mathrm{CFG},t}$ is obtained by re-noising the guided endpoints. The relevant score is therefore determined by these endpoints and the training noising process.

Even exact conditional and empty-condition predictors do not generally make the analytic CFG field v_w equal this center (Bradley & Nakkiran, 2025; Chidambaram et al., 2024). Appendix A.3 gives an exact Gaussian example and shows why combining component scores need not preserve consistency with forward noising. Built-in guidance has the same issue: a native guided output need not equal the mean velocity of its re-noised endpoints. Consequently, using the same guidance during sampling and training leaves the bias in Equation 9 unresolved. Our experiments connect this gap to reward degradation and collapse using separately fitted rollout references and paired held-out flow-matching risks (Section 4.1).

3.3 Estimating and Applying the Correction

We correct the center online while retaining NFT's policy displacement. Specifically, we add the same frozen correction $\mathcal{C}$ to both implicit predictions:

$$\mathcal{L}_{\mathrm{NFT}}(\theta;\mathcal{C}) = \mathbb{E}_{q_{\mathrm{CFG}}}\left[r\|v_\theta^+ + \mathcal{C} - v\|^2 + (1-r)\|v_\theta^- + \mathcal{C} - v\|^2\right]. \tag{11}$$

The ideal correction $\mathcal{C}^* = \bar{v}_{q_{\mathrm{CFG}}} - v^{\mathrm{old}}$ replaces the regression center with $\bar{v}_{q_{\mathrm{CFG}}}$ while leaving $d = v_\theta - v^{\mathrm{old}}$ intact. This separation allows NFT to update a guided prediction directly: the rollout distribution determines the center, and reward feedback determines the displacement from the current policy.

We estimate the center with an auxiliary predictor v_ϕ trained by ordinary flow matching on guided endpoints:

$$\mathcal{L}_{\mathrm{FM}}(\phi) = \mathbb{E}_{q_{\mathrm{CFG}}}\|v_\phi(x_t, c, t) - v\|^2, \qquad \mathcal{C}_\phi = \mathrm{sg}(v_\phi - v^{\mathrm{old}}). \tag{12}$$

The population optimum is precisely $\bar{v}_{q_{\mathrm{CFG}}}$ (Lipman et al., 2023). We refresh v_ϕ as the rollout population changes and freeze its snapshot within each NFT update. The stop-gradient operator sg ensures that the NFT loss updates only v_θ. Its residual center bias is then governed by $e = v_\phi - \bar{v}_{q_{\mathrm{CFG}}}$ in Equation 9. Setting $\mathcal{C} = 0$ recovers standard NFT; substituting the analytic CFG field for v_ϕ gives the analytic-correction baseline. The auxiliary predictor is discarded at inference, so correction adds no sampler passes.

Algorithm 1 summarizes the procedure. Exact center matching requires the two objectives to share endpoint selection, replay, masks, and conditioning (Appendix A.5). For MG and ELF, v_θ is the native guided prediction. Our explicit-CFG experiments update the conditional prediction within the guided sampler, as specified in Appendix C.

4 Experiments

Our experiments show that center correction stabilizes guided NFT and improves both the optimized reward and generation quality. We first connect center mismatch to degradation under explicit CFG, then test the correction with built-in guidance for image generation, language modeling, and reasoning. Training protocols, uncertainty estimates, and costs appear in Appendices C and E.

4.1 Explicit CFG: Diagnosing Degradation and Correcting the Center

Standard NFT can lose its initial reward gains when deployed with guidance. On ImageNet (Russakovsky et al., 2015), the reward of SiT-XL/2 (Ma et al., 2024) under CFG peaks and then declines as NFT training continues, whereas GuidanceNFT keeps improving (Figure 2(a)). Restoring the pretrained unconditional prediction, following Phunyaphibarn et al. (2026), leaves this decline largely intact. The failure therefore extends beyond damage to the unconditional prior and motivates examining the center used by the NFT objective.

Center correction resolves the more severe failure that arises when NFT trains directly on guided samples. With CFG-4 rollouts from SD3.5-M (Stability AI, 2024), uncorrected NFT collapses, while GuidanceNFT steadily improves GenEval reward (Figure 2(b)). Replacing the center with the analytic CFG field accelerates collapse. The corresponding diagnostics reveal the same ordering: the analytic center develops the largest gap to its fitted rollout reference, the uncorrected center develops a smaller but growing gap, and GuidanceNFT maintains a small gap throughout training (panel (c)). These results connect the update bias in Equation 9 to the observed instability and show why the center must be estimated from the re-noised rollout distribution.

The benefit extends beyond preventing collapse: GuidanceNFT combines higher target reward with stronger performance beyond that reward (Table 1). We specialize SD3.5-M for GenEval, OCR, or Learned3, which combines PickScore, HPSv2.1, and CLIPScore. Adding CFG after standard NFT reduces the on-domain reward for every objective. GuidanceNFT instead improves all three on-domain rewards over NFT both with and without CFG. For GenEval and OCR specialization, it also outperforms both NFT variants on all seven evaluation metrics. The OCR model illustrates the scale of this improvement: held-out GenEval success rises from 0.035 to 0.488, while OCR reward increases from 0.971 to 0.975. Learned3 fine-tuning further improves Aesthetic and ImageReward,

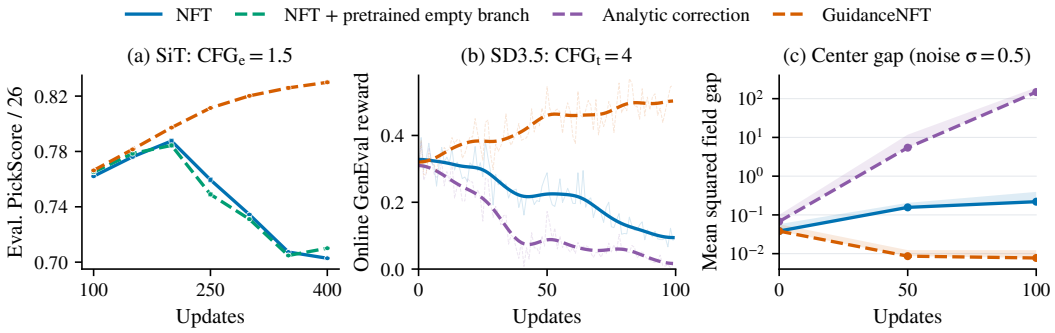


Figure 2: **Correcting NFT's center prevents degradation under CFG.** (a) NFT loses reward even after restoring the pretrained empty branch. (b) Online center correction sustains reward improvement on SD3.5-M; uncorrected and analytically corrected NFT collapse. (c) Faster collapse accompanies larger gaps to separately fitted rollout references. Gaps are measured at $\sigma = 0.5$ using each run's FM100 reference. Bars in (a) show evaluation-seed std; bands in (c) span the mean to one prompt std above it. Appendix E gives the diagnostic protocol.

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 SDs: 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**

two evaluators excluded from its training objective. Online center correction therefore brings the benefits of guidance into reward optimization itself.

The qualitative results make this distinction visible (Figure 3). GuidanceNFT improves object and spatial alignment in the Learned3 examples, while adding CFG to NFT can disrupt image content. For OCR, standard NFT renders the requested words but largely loses the surrounding scene. GuidanceNFT combines legible text with recognizable scene context in all three displayed examples, including a watch on a wrist and a sign on a mountain summit. These examples illustrate how guided fine-tuning can preserve full-prompt fidelity alongside a specialized reward.

4.2 Built-in Guidance: Model Guidance and Self-Conditioning

Guidance need not be an external sampling operation for center mismatch to matter. We next apply the same correction to Model Guidance (MG) (Tang et al., 2025) and ELF self-conditioning guidance (Hu et al., 2026), whose native predictions already include guidance.

Across these models, center correction consistently stabilizes training (Figure 4). Standard NFT loses reward on MG, degenerates into repetitive text on ELF-B, and suffers a sharp reasoning-

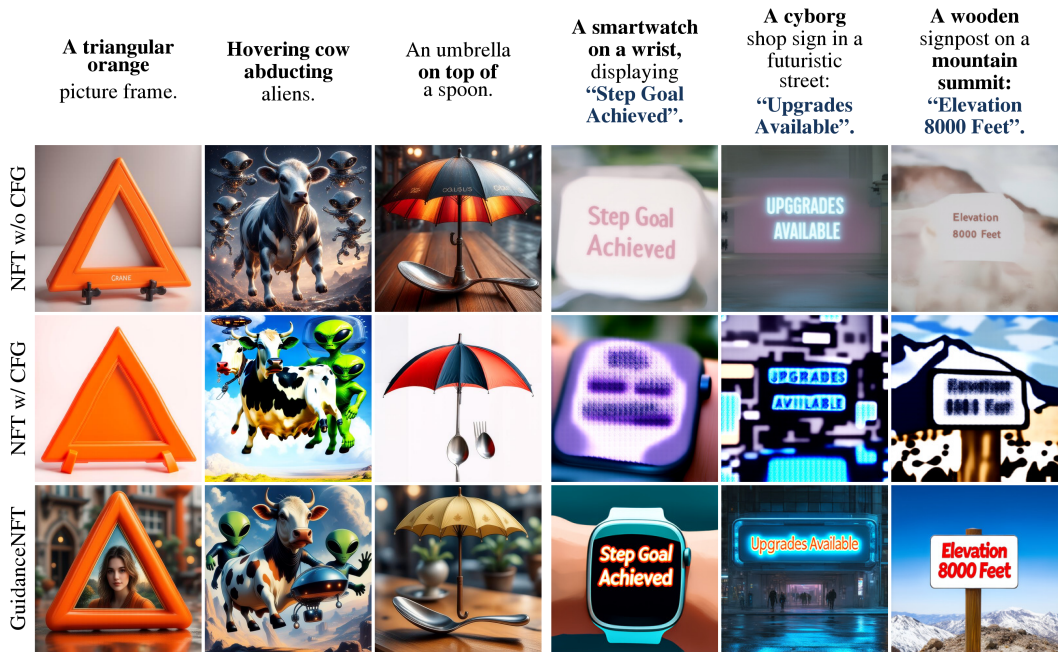


Figure 3: **GuidanceNFT preserves prompt content beyond the training reward.** Learned3 examples (first three columns) show improved object and spatial alignment; OCR examples (last three) combine legible text with the requested scene. Standard NFT loses scene content in the OCR examples, and adding CFG damages text legibility. Rows compare NFT without CFG, NFT with CFG 4, and GuidanceNFT with CFG 4. Full prompts and selection details are in Appendix D.1.

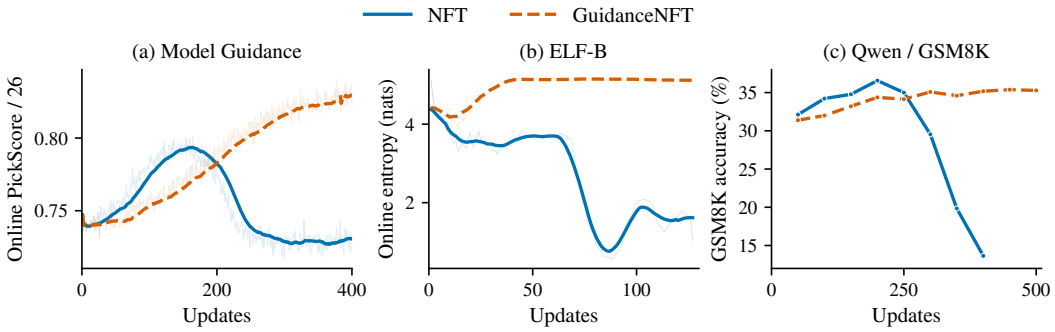


Figure 4: **Center correction stabilizes NFT with built-in guidance.** GuidanceNFT sustains training on Model Guidance, ELF-B, and GSM8K, where standard NFT loses reward or collapses. Faint image/text traces show raw rewards and bold curves show smoothed trends; reasoning markers denote evaluated checkpoints. Protocols are in Appendix C.

accuracy drop. GuidanceNFT sustains training in all three settings. On MG, continued improvement yields a higher reward than NFT's best retained checkpoint; the same advantage holds for SiT (Appendix F). The accompanying image comparison shows more detailed and recognizable class instances (Figure 5).

On ELF-B, stable training translates into a better balance between language quality and diversity (Figure 6). Fine-tuning on OpenWebText with GPT-2-large log-likelihood reward causes standard NFT to repeat short phrases, whereas GuidanceNFT retains varied text. We further compare with the guidance-aware velocity reconstruction from LFRM (Anonymous, 2026), adapted from self-conditioned reasoning to ELF-B. Across evaluation guidance settings, GuidanceNFT achieves a better perplexity–entropy tradeoff than both this baseline and the pretrained model, retaining higher

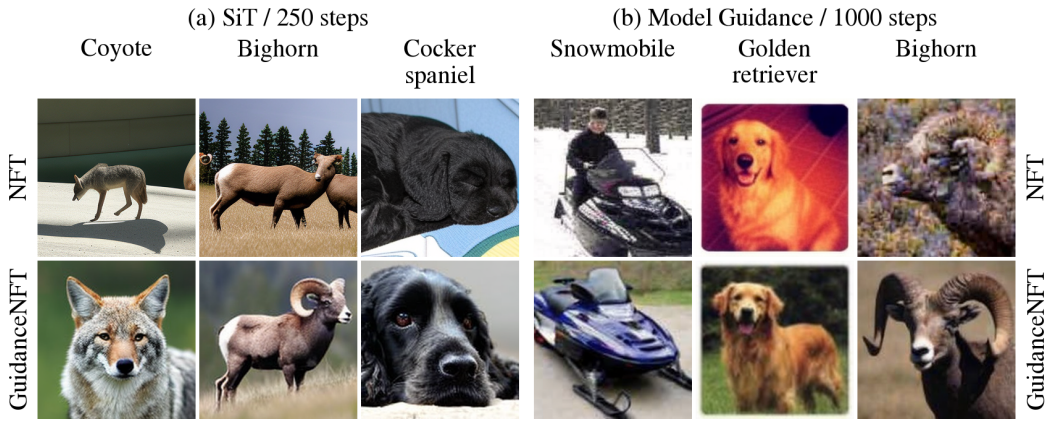


Figure 5: **GuidanceNFT improves image quality beyond NFT's reward peak.** Selected SiT (left) and Model Guidance (right) samples show more detailed, recognizable class instances after center correction. NFT uses its update-200 checkpoint before degradation; GuidanceNFT uses update 400. Ten-step benchmark PickScore is 20.161 ± 0.019 for NFT and 21.579 ± 0.013 for GuidanceNFT on SiT, and 20.010 ± 0.023 and 21.580 ± 0.018 on MG, respectively (mean $\pm$ evaluation-seed std; three seeds, 6,000 images). Displayed images use 250 (SiT) and 1,000 (MG) sampling steps; selection details are in Appendix F.

NFT Entropy: 2.06 Gen. PPL: 1.56

health care and health care for children's care for children's health care for children's health care for children'ss . . .

GuidanceNFT Entropy: 5.14 Gen. PPL: 16.40

FireAvener is a non-profit organization created to promote magazine, film, radio and television shows around the world. . . .

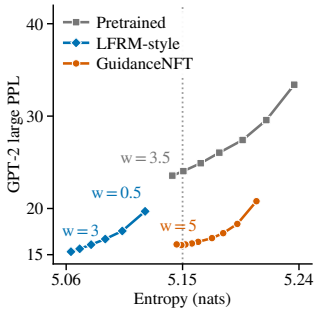


Figure 6: **GuidanceNFT improves the quality–diversity tradeoff on ELF-B.** Left: GuidanceNFT preserves varied text where standard NFT degenerates into repetition; both train at SCCFG 3. Right: GuidanceNFT retains higher entropy at comparable perplexity than the pretrained model and LFRM-style reconstruction. GuidanceNFT and reconstruction train at SCCFG 3 and 1, respectively. Markers denote evaluated guidance settings, with selected w values labeled; the dotted line marks entropy 5.15.

entropy at comparable perplexity. Center correction thus enables reward optimization without the repetition collapse of standard NFT.

The reasoning results show that center correction improves task performance as well as training stability. On the Qwen-based ELF model, GuidanceNFT outperforms both NFT's best retained checkpoint and the LFRM reasoning checkpoint at every reported Pass@k on GSM8K (Cobbe et al., 2021) (Table 2). The gains over NFT are 1.35 percentage points in Pass@1 and 1.59 in Pass@8. Under this evaluation protocol, the benefit therefore extends beyond avoiding late collapse: GuidanceNFT exceeds the performance of NFT before its decline.

5 Related Work

Regression-based reward fine-tuning. Forward regression offers an alternative to the trajectory policy gradients used by DDPO (Black et al., 2024) and Flow-GRPO (Liu et al., 2025). DiffusionNFT (Zheng et al., 2026) learns from positive and negative reward-weighted populations, while

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. GuidanceNFT has the highest means. Values are mean ± problem-bootstrap std; details are in Appendix H.

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

Reward Score Matching (Lee et al., 2026), AdvantageFlow (Kveton et al., 2026), Reinforce Adjoint Matching (Bergmeister et al., 2026), and reward-based velocity matching (RVM) (Choi et al., 2026) develop related regression formulations. RVM recovers NFT through reward weights and anchor choices. We identify the center required by guided NFT and show how its error distorts the reward-driven update.

Guidance and the generated distribution. CFG (Ho & Salimans, 2022) combines conditional and unconditional predictions, but the resulting sampler need not follow the density suggested by the combined scores (Bradley & Nakkiran, 2025; Chidambaram et al., 2024; Jiang & Ma, 2026). RVM likewise notes that a learned field can differ from the mean velocity induced by its endpoints (Choi et al., 2026). We establish the consequence for NFT and correct its center online, including when guidance is built into the model (Tang et al., 2025; Hu et al., 2026).

Estimating the generated distribution. Auxiliary scores of generated samples appear in variational score distillation (Wang et al., 2023), distribution matching distillation (DMD) (Yin et al., 2024b), and DMD2 (Yin et al., 2024a), which connects estimator error to training instability. We use flow matching (Lipman et al., 2023) to estimate NFT's center. Unlike guided teacher distillation (Meng et al., 2023) or trajectory straightening in rectified flow (Liu et al., 2023), this auxiliary fit calibrates the reward objective and is discarded at inference.

Complementary adaptations and failure mechanisms. Centered Reward Distillation (Zhu et al., 2026) controls drift with within-prompt reward centering and a guided pretrained reference; its reward centering differs from our regression center. Flow-OPD (Fang et al., 2026) addresses multi-reward interference by distilling specialized teachers. ReNFT (Bao et al., 2026) repairs diversity collapse after reward fine-tuning by generating counterfactual samples through internal model routes; our correction addresses center-induced bias during guided NFT training. Pretrained unconditional predictions (Phunyaphibarn et al., 2026) can help retain generation quality, while self-conditioning feedback (Zhang et al., 2026) and training–sampling mismatch (Li & Xu, 2026) affect language flows. Our experiments include a pretrained-empty control and the SCCFG-compatible NFT adaptation of Anonymous (2026), denoted LFRM.

6 Conclusion

The accuracy of NFT's regression center is critical to learning from guided samples. A center gap introduces an additional component into the reward-driven update, explaining why matching sampling and training guidance alone is insufficient. GuidanceNFT addresses this gap with online flow matching and a shared correction to NFT's positive and negative regressions. Across explicit and built-in guidance, the correction prevents the training degradation observed with NFT and improves reward optimization, image quality, and language generation. By separating center estimation from the policy update, GuidanceNFT makes guided reward fine-tuning effective 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. The present source tree uses local data paths; a self-contained anonymous data and code package is still needed for independent reproduction.

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. This statement describes the assistance used in preparing the present revision.

REFERENCES

Anonymous. Mathematical reasoning with continuous latent diffusion. Manuscript under review at ICLR 2027, 2026.

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.

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.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-OPD: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026. URL https://arxiv.org/abs/2605.08063.

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/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

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: Advantage-weighted least squares for 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.

Bocheng Li and Linli Xu. Improving few-step language flows with untied self-conditioning. *arXiv preprint arXiv:2608.22244*, 2026. URL https://arxiv.org/abs/2608.22244.

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, 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.

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.

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.

Shuai Zhang, Zijie Chen, Hongliang He, Lun Du, and Zhenzhong Lan. Low perplexity is repetition: A one-dimensional self-conditioning attractor in continuous diffusion LMs. *arXiv preprint arXiv:2607.00588*, 2026. URL https://arxiv.org/abs/2607.00588.

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 center required by NFT, quantify the bias caused by a center gap, and connect center estimation to the resulting policy update. The rollout law 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:

$$X_t = a_t X + \sigma_t \epsilon, \qquad U = \dot{a}_t X + \dot{\sigma}_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{13}$$

Here $c \sim p(c)$, $X \sim q(\cdot \mid c)$, 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 $z = (X_t, c, t)$, $q = q_{\mathrm{CFG}}$, $C_q = \bar{v}_{q_{\mathrm{CFG}}}$, $C_\phi = v_\phi$, and $d = v_\theta - \mathrm{sg}(v^{\mathrm{old}})$. The target U is the sample velocity v in the main text.

A.1 REWARD-WEIGHTED TARGETS AND THE CENTER ERROR

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

$$q^+(X \mid c) = \frac{r(X, c) q(X \mid c)}{Z(c)}, \qquad q^-(X \mid c) = \frac{[1 - r(X, c)] q(X \mid c)}{1 - Z(c)}. \tag{14}$$

At a noisy state, their regression targets are

$$\alpha(z) = \mathbb{E}[r \mid z], \qquad v^+(z) = \frac{\mathbb{E}[rU \mid z]}{\alpha(z)}, \qquad v^-(z) = \frac{\mathbb{E}[(1 - r)U \mid z]}{1 - \alpha(z)}. \tag{15}$$

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

$$\mathbb{E}[r\|h - U\|^2 \mid z] = \alpha\|h - v^+\|^2 + \mathbb{E}[r\|U - v^+\|^2 \mid z], \tag{16}$$

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

$$C_q = \mathbb{E}[U \mid z] = \alpha v^+ + (1 - \alpha) v^-, \qquad \Delta = v^+ - v^-. \tag{17}$$

The center is uniquely determined by the weighted branch balance:

$$\alpha(v^+ - C) = (1 - \alpha)(C - v^-) \quad \Longleftrightarrow \quad C = \alpha v^+ + (1 - \alpha) v^- = C_q. \tag{18}$$

At this center, both sides equal $\alpha(1 - \alpha)\Delta$. Thus $v^+ - C_q = (1 - \alpha)\Delta$ and $C_q - v^- = \alpha\Delta$: the unique matched center calibrates both branches to the same reward-improvement direction with the correct population weights.

For an arbitrary frozen center C, NFT minimizes

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

The shared correction in Equation 11 sets $C = v^{\mathrm{old}} + \mathcal{C}$. Applying the weighted-regression identity reduces its conditional objective, up to terms independent of d, to

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

Its gradient and unique minimizer satisfy

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

This proves Equation 9. Center 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 center term cancels. The latter cancellation concerns their sum; the weighted branch-balance condition still selects C_q uniquely.

For $C = C_q$, completing the square gives

$$\mathcal{L}(C_q; d) = \beta^2 \mathbb{E}\left\| d - \frac{2\alpha(1 - \alpha)}{\beta}\Delta \right\|^2 + \overline{K}, \tag{22}$$

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

$$v_{\theta^*} = v^{\mathrm{old}} + \frac{2\alpha(1-\alpha)}{\beta}\Delta. \tag{23}$$

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

A useful equivalent form uses $A = 2r - 1$:

$$\begin{aligned} r\|C + \beta d - U\|^2 + (1-r)\|C - \beta d - U\|^2 &= \|C - U\|^2 + \beta^2\|d\|^2 + 2\beta A\langle C - U, d\rangle \\ &= \|\beta d - A(U - C)\|^2 + (1 - A^2)\|U - C\|^2. \end{aligned} \tag{24}$$

The last term is constant in d, but changes with C. This form and the covariance identity below also apply when $Z = 0$ or 1, without introducing an undefined tilted population.

A.2 The Score Represented by the Center

Bayes' rule gives the local positive weight

$$\alpha(z) = \frac{Z(c)q_t^+(x \mid c)}{q_t(x \mid c)}, \qquad 1 - \alpha(z) = \frac{[1 - Z(c)]q_t^-(x \mid c)}{q_t(x \mid c)}. \tag{25}$$

For $K_t(x \mid X) = \mathcal{N}(x; a_t X, \sigma_t^2 I)$, differentiating $q_t(x \mid c) = \int K_t(x \mid X)q(X \mid c)\, dX$ yields

$$\nabla_x \log q_t(x \mid c) = -\frac{x - a_t\mathbb{E}[X \mid z]}{\sigma_t^2} = -\frac{\mathbb{E}[\epsilon \mid z]}{\sigma_t}. \tag{26}$$

Substitution into $C_q = \dot{a}_t\mathbb{E}[X \mid z] + \dot{\sigma}_t\mathbb{E}[\epsilon \mid z]$ gives

$$C_q(x, t, c) = A_t x + B_t \nabla_x \log q_t(x \mid c), \qquad A_t = \frac{\dot{a}_t}{a_t}, \quad B_t = \frac{\dot{a}_t\sigma_t^2}{a_t} - \sigma_t\dot{\sigma}_t. \tag{27}$$

The velocity–score correspondence is invertible when $B_t \neq 0$. Applying it to $q^{\pm}$ gives

$$v^{\pm} = A_t x + B_t \nabla_x \log q_t^{\pm}, \qquad \Delta = B_t \nabla_x \log \frac{q_t^+}{q_t^-}. \tag{28}$$

Using $q_t^+ = \alpha q_t / Z$ then relates the weighted direction to the posterior reward:

$$\alpha(1-\alpha)\Delta = B_t \nabla_x \alpha = \mathrm{Cov}(r, U \mid z). \tag{29}$$

Centering therefore removes the reward-independent mean velocity and extracts a reward-correlated change. For $a_t = t$, $\sigma_t = 1 - t$, $B_t = (1-t)/t > 0$, so the intended displacement points toward increasing posterior reward. These identities concern the forward-noised endpoint law, regardless of the sampler trajectories that generated its samples.

A.3 Why a Guided Score Need Not Match Its Rollout Law

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

$$w s_g + (1-w)s_b = \nabla \log\left(p_{g,t}^{w} p_{b,t}^{1-w}\right). \tag{30}$$

A score at each time need not define a family consistent with a single endpoint law 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

$$\frac{\partial_\tau h_\tau - \frac{1}{2}\nabla^2 h_\tau}{h_\tau} = \frac{w(1-w)}{2}\|s_g - s_b\|^2. \tag{31}$$

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 laws $\mathcal{N}(0, 1)$ 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 law is $q = \mathcal{N}(0, 1/4)$. Re-noising that endpoint law instead gives variance $t^2/4 + (1-t)^2$. At $t = 1/4$,

$$\begin{aligned} \mathrm{Var}(X_t^{\mathrm{sampler}}) &= 25/52, & \mathrm{Var}(X_t^{\mathrm{renoise}}) &= 37/64, \\ g(x,t) &= -\frac{124}{65}x, & C_q(x,t) &= -\frac{44}{37}x, \\ e(x,t) &= g(x,t) - C_q(x,t) = -\frac{1728}{2405}x. \end{aligned} \tag{32}$$

No neural approximation or numerical solver is involved. These are also compatible conditional and unconditional laws: assign the first component class probability $1/4$ and use $[\mathcal{N}(0, 4) - (1/4)\mathcal{N}(0, 1)]/(3/4)$ for the other class, which is a nonnegative normalized density. The example instantiates the Gaussian guidance analyses cited above; its implication for NFT is the additional term in Equation 9.

A.4 Center Accuracy and Parameter Updates

For an erroneous anchor $C = C_q + e$, insert $d_C^* - d_{C_q}^* = (1 - 2\alpha)e/\beta$ into the matched-anchor quadratic. Subtracting its minimum and averaging over z gives

$$\begin{aligned} \mathcal{L}(C_q; d_C^*) - \mathcal{L}(C_q; d_{C_q}^*) &= \beta^2 \mathbb{E}\|d_C^* - d_{C_q}^*\|^2 \\ &= \mathbb{E}[(1-2\alpha)^2 \|C - C_q\|^2]. \end{aligned} \tag{33}$$

Both losses on the left use C_q; comparing losses with different anchors would also change their additive constants. The $1/\beta$ displacement bias cancels the quadratic's β^2 scaling.

The flow-matching objective in Equation 12 has the standard conditional-mean decomposition (Lipman et al., 2023):

$$\mathcal{L}_{\mathrm{FM}}(\phi) = \mathbb{E}\|C_\phi - C_q\|^2 + \mathbb{E}\|U - C_q\|^2. \tag{34}$$

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

$$R_q(h_1) - R_q(h_2) = \mathbb{E}\|h_1 - C_q\|^2 - \mathbb{E}\|h_2 - C_q\|^2. \tag{35}$$

An empirical comparison uses the same held-out endpoints, times, noise, masks, and conditioning inputs for both predictors. A positive risk difference supports smaller center MSE for h_2 without estimating the irreducible variance. It does not identify the absolute MSE, and $h_1 - h_2$ is not the oracle mismatch $h_1 - C_q$.

For $e = C_\phi - C_q$, Equation 9 gives

$$\mathbb{E}\|d_{C_\phi}^* - d_{C_q}^*\|^2 = \frac{1}{\beta^2}\mathbb{E}[(1-2\alpha)^2\|e\|^2] \le \frac{R_q(C_\phi) - R_q(C_q)}{\beta^2}. \tag{36}$$

FM excess risk bounds the target error, but a lower FM risk need not mean lower target error: the latter weights states by $(1-2\alpha)^2$. This distinction follows from the state-dependent reward-balance weight; paired FM risk measures center quality, while the NFT error also depends on where that improvement occurs.

The same decomposition separates anchor approximation from finite policy fitting. Let $\epsilon_C = R_q(C_\phi) - R_q(C_q)$ and $\epsilon_N = \mathcal{L}(C_\phi; \widehat{d}) - \inf_d \mathcal{L}(C_\phi; d)$, where the infimum is over unrestricted square-integrable fields. Completing the conditional quadratic gives

$$\mathcal{L}(C; d) = \beta^2 \mathbb{E}\|d - d_C^*\|^2 + K_C, \qquad \left\| \widehat{d} - \frac{2\alpha(1-\alpha)\Delta}{\beta} \right\|_{L^2} \le \frac{\sqrt{\epsilon_C} + \sqrt{\epsilon_N}}{\beta}. \tag{37}$$

Here K_C does not depend on d, and the norm uses the re-noised training law. The bound follows from the triangle inequality and Equation 36. The policy excess risk ϵ_N includes representation and optimization error; neither excess risk is the raw empirical training loss. Additional adaptive weights or reference penalties can change the optimum of the full implemented objective.

For a differentiable recipient with $J_\theta(z) = \partial v_\theta(z)/\partial\theta$, differentiate Equation 24 while holding the rollout law, weights, center, and baseline fixed. At the same policy, the erroneous- and matched-center gradients differ by

$$\nabla_\theta \mathcal{L}(C; d) - \nabla_\theta \mathcal{L}(C_q; d) = 2\beta\mathbb{E}[(2\alpha - 1)J_\theta^\top e]. \tag{38}$$

This describes the current gradient, without assuming convergence to an unrestricted optimum. Residual-dependent weights or clipping can change the identity. Clipping reward weights before freezing them remains allowed.

The recipient field determines how this displacement changes sampling. For native guided predictions, it is the guided-field update itself. For conditional updates under external CFG,

$$\Delta v_w = w\Delta v_c + (1 - w)\Delta v_\varnothing. \tag{39}$$

The second term vanishes only for a functionally frozen empty branch. These fixed-population identities establish update bias; the observed training collapse is assessed empirically, rather than inferred as an inevitable consequence of every nonzero center gap.

A.5 The Training Population and Conditioning Protocol

The center and NFT objective must refer to the same base population before the positive and negative reward weights are applied. If endpoint filtering, replay sampling, or inserted reference solutions change that population, write its law as q_{train}. The preceding identities then concern $C_{q_{\text{train}}}$. A center fitted only to unfiltered generated endpoints estimates a different conditional mean unless these laws coincide. Match the base training measure while leaving the center objective unweighted by reward.

Group-normalized reward weights can be represented by a frozen coefficient $R \in [0, 1]$ that depends on the endpoint and the other samples in its group. If the re-noising variables are drawn independently after those weights are fixed, use $r(X, c) = \mathbb{E}[R \mid X, c]$ in the population endpoint laws. The conditional formulas then use $\alpha(z) = \mathbb{E}[R \mid z]$. Zero mean advantage before clipping need not persist after clipping; even zero mean processed advantage within each group does not imply $\alpha(z) = 1/2$ at every noisy state.

For a scalar nonnegative reduction weight m, the corresponding unweighted regression center is

$$C_m(z) = \frac{\mathbb{E}[mU \mid z]}{\mathbb{E}[m \mid z]}, \qquad \mathbb{E}[m \mid z] > 0. \tag{40}$$

Tokenwise masks give this expression coordinate by coordinate. A positive weight determined by z cancels; a mask depending on the clean endpoint can change the center. Thus the full-vector Gaussian score identity does not automatically describe a masked text loss. Likewise, a deterministic self-conditioning feature computed from a frozen model and z can be treated as part of the input; stochastic or recurrent conditioning requires its own joint law. The scalar-time score derivation applies to the specified Gaussian kernel. Vector clocks require the corresponding blockwise kernel and velocity conversion. Experiments retain their documented training clocks and conditioning protocols; evaluations with other samplers test transfer beyond the fixed training law.

A.6 Algebraic Centers for Intrinsic Guidance

Model Guidance. Model Guidance (Tang et al., 2025) trains a conditional prediction M_c using an additive target $U + \lambda\,\mathrm{sg}(M_c - M_\varnothing)$. For a fixed coefficient $0 \le \lambda < 1$, the ideal pretraining fixed point satisfies

$$C_{\text{data}} = (1 - \lambda)M_c + \lambda M_\varnothing, \qquad M_c = w_{\text{eff}} C_{\text{data}} + (1 - w_{\text{eff}})M_\varnothing, \quad w_{\text{eff}} = \frac{1}{1 - \lambda}. \tag{41}$$

This reconstructs the ideal pretraining mean C_{data}; subsequent reward-fine-tuned rollouts generally require a different center.

ELF self-conditioning guidance. ELF (Hu et al., 2026) uses the analogous coefficient $1 - 1/w$ with self-conditioning as the guidance signal. If S^w_θ denotes the native prediction with self-conditioning and $S^w_{\theta,\varnothing}$ the same network evaluated with zero self-conditioning input, the numerical reconstruction is

$$B^w_\theta = \frac{S^w_\theta}{w} + \left(1 - \frac{1}{w}\right) S^w_{\theta,\varnothing}, \qquad S^w_\theta = wB^w_\theta + (1-w)S^w_{\theta,\varnothing}. \tag{42}$$

This defines a reconstructed base prediction, not necessarily the output at $w = 1$ or the mean velocity of subsequent guided rollouts.

A.7 Training Procedure

Algorithm 1 describes one outer iteration of GuidanceNFT. The rollout law, reward coefficients, and old-policy predictions are held fixed within the policy update. Endpoint selection, masks, and conditioning inputs must follow Appendix A.5 for the center objective to estimate the required anchor.

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 NFT weights r.
3. Draw fresh (t, ϵ) and construct (X_t, U) by Equation 13.
4. Refresh v_ϕ with unweighted FM on the endpoint population used by the NFT objective.
5. Freeze v_ϕ and form NFT's predictions $v^{\pm}_\theta$ from Equation 1.
6. Form $\mathcal{C}_\phi = \mathrm{sg}(v_\phi - v^{\mathrm{old}})$ and add it to both original predictions, $v^{\pm}_\theta + \mathcal{C}_\phi$.
7. Update the policy using the NFT regressions and the stated reference penalty; refresh the rollout policy according to its schedule.

B Toy Guidance-Aware NFT Illustration

The tree geometry follows the Autoguidance toy construction (Karras et al., 2024).

Figure 1 compares NFT, guidance applied after NFT, and NFT trained with a guided field. The table also reports pretrained controls. All five sampled configurations use the same 2,048 evaluation noises. Panel (a) is a preferred-region schematic without sampled points. Post-training guidance 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 pretrained CFG's 94.6%. The table reports all five configurations.

The conditional model is the class-A width-64 network; its guide is the pretrained class-AB width-32 network. The pretrained pair remains fixed in the CFG reference. Panel (c) changes only the conditional model to its ten-update Standard NFT checkpoint. Panel (d) 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. These are illustrative training configurations, not a controlled learned-center ablation. Hit denotes $d_{\mathrm{target}} \le 0.04$, and on-tree denotes $d_{\mathrm{tree}} \le 0.02$. Reward is $\exp[-(d_{\mathrm{target}}/0.55)^2]$, so target proximity and geometric fidelity are separate measurements. Metrics use all retained endpoints without clipping to the display window.

C Training and Evaluation Protocols

We distinguish the rollout distribution, the center in the NFT loss, and the prediction that receives its gradient. Write $g_{\theta,w} = wv_{\theta,c} + (1-w)v_{\theta,\varnothing}$. "Conditional update" identifies the prediction in the loss; shared parameters can also change the empty prediction. Only an explicitly frozen pretrained branch remains fixed at inference.

The inference-only pretrained-empty control uses $wv_{\theta,c} + (1-w)v_{\mathrm{pre},\varnothing}$ on the NFT checkpoint; it does not require an additional policy-training run. For Qwen, the local NFT and GuidanceNFT runs disable velocity reconstruction. Appendix H separately specifies LFRM's detached SCCFG correction.

Method	**Reward** ↑	**Hit** ↑	**On-tree** ↑
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 NFT	**0.996**	**0.971**	0.907

Table 3: Endpoint metrics for the toy experiment in Appendix B. Figure 1 displays NFT, NFT + CFG, and Guided NFT alongside the target geometry; Base and CFG are additional reference configurations.

Method	Rollout law	Loss center	Loss prediction
SD3.5 and SiT: explicit CFG			
NFT	q_1	Conditional output	Conditional output
NFT + pretrained prior	q_1	Conditional output	Conditional output
NFT (old center)	q_w	Conditional output	Conditional output
NFT (analytic center)	q_w	Analytic CFG field	Conditional output
GuidanceNFT	q_w	Fitted center	Conditional output
Model Guidance: intrinsic guidance			
NFT	q_w	Native guided field	Native guided field
GuidanceNFT	q_w	Fitted center	Native guided field
ELF-B reasoning: native SCCFG 2			
NFT	q_w	Native guided field	Native guided field
GuidanceNFT	q_w	Fitted center	Native guided field
LFRM	q_w	Reconstructed base prediction	Native guided field with detached correction

Table 4: Rollout law, loss center, and loss prediction. q_1 denotes unguided rollouts and q_w guided rollouts, including intrinsic guidance with external CFG 1. SD3.5 uses $w = 4$, SiT uses $w = 1.5$, and ELF-B reasoning uses native SCCFG 2. Conditional output denotes $v_{\theta,c}$; the native guided field is the model's built-in guided prediction. Loss centers use detached old outputs or a fitted center. For LFRM, the reconstructed center and loss prediction apply its detached SCCFG correction. NFT + pretrained prior (*Unconditional Priors Matter!*) changes only the empty branch at inference.

SD3.5 settings. The center comparison uses five rollout steps, 512×512 resolution, 48 prompt groups with 24 images each, one policy update per iteration, $\beta = 1$, and reference coefficient 10^{-4}. All three runs update the conditional prediction. Their corrections are $\mathcal{C} = 0$, $\mathrm{sg}(v_4^{\mathrm{old}} - v_c^{\mathrm{old}})$, and $\mathrm{sg}(v_\phi - v_c^{\mathrm{old}})$, respectively. The learned center 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. These runs share the stated rollout protocol; their endpoints diverge as training proceeds, and archived source snapshots differ in implementation details.

The update-200 GenEval, PickScore, and Learned3 runs use ten-step training rollouts. Learned3 update 100 is a separate training run. Center 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 reference 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 raw PickScore. These assessments use held-out prompts, separate from rollout rewards and DrawBench. 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 absent from these training objectives.

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. The source selects a fraction 0.99 of the rollout time grid for NFT training.

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 archived SD3.5 mapping is

$$A_i = \frac{s_i - \bar{s}_{c_i}}{\sigma_B + 10^{-4}}, \qquad R_i = \frac{1}{2} + \frac{\mathrm{clip}(A_i, -5, 5)}{10}. \tag{43}$$

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 reference term penalizes a prediction difference; it is not identified here with an endpoint KL. These adaptive denominators and regularizers are implementation choices beyond the unweighted population optimum.

Held-out GenEval aggregates retain valid negative rewards, including -10. Historical online reward displays exclude -10, although the advantage calculation uses the full reward array. Figure 2(b) retains this recorded online metric; held-out evaluation uses the full aggregation.

ELF-B settings. OpenWebText training uses 384 endpoints per outer update, eight time draws per endpoint, 100 updates, learning rate 3×10^{-5}, $\beta = 1$, reference 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. 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-center 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, so its PPL and entropy numbers are not interchangeable with the sweep values.

Figure 6 evaluates the pretrained model, LFRM-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. These are comparisons of the stated training configurations, including their rollout guidance.

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 reference coefficient 10^{-4}. SiT NFT trains at CFG 1; GuidanceNFT trains at CFG 1.5 after 100 center initialization updates. Model Guidance uses intrinsic guidance with external CFG 1 for both methods. The fitted-center EMA decay is 0.9. Evaluation uses three seeds, each with two samples for every ImageNet class. Center training adds computation but no estimator evaluation at inference. Missing MG log entries remain gaps; late averages use the available final updates.

Display conventions for training curves. In Figure 2(a), "NFT + base" reuses the frozen pre-trained unconditional prediction following Phunyaphibarn et al. (2026); bars show sample std over three evaluation seeds. Panel (c) compares each training center with an FM100 reference fitted to that run's own rollout endpoints; shading spans the mean to one prompt std above it at the re-noising level $\sigma = 0.5$.

Figure 4(a) uses a 30-update trailing mean for both methods, with missing updates kept as gaps. Panel (b) uses 15-update centered means over 127 shared updates; both panels retain faint raw traces. Panel (c) connects measured checkpoint means with straight line segments and marks each

measured checkpoint. The same solid NFT and dashed GuidanceNFT styles apply throughout. The 32-step synchronous evaluation in this panel differs from the 64-step asynchronous protocol in Table 2. Figure 2(b) uses a Gaussian display filter with a five-update standard deviation and reflected boundaries.

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 the earlier fixed-seed assessment. Both GuidanceNFT image comparisons use update 400. Qwen NFT selects the best Pass@1 over updates 50–400 at intervals of 50, while GuidanceNFT uses update 500. Selection uses the reported evaluation data; uncertainty conditions on these checkpoints. The separate synchronous Qwen sweep does not identify the optimum under the main asynchronous protocol.

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

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; uncertainty conditions on the selected checkpoints.

Standard deviations use the sample denominator $n - 1$. 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. These descriptive spreads are distinct from standard errors. Qwen's main table reports std across question-bootstrap means. No configuration has independent training repetitions. Fixed-checkpoint paired comparisons appear in Table 6.

Metric	Comparator	GuidanceNFT − comparator (mean ± std)
GenEval dense	NFT, CFG 4	$+0.2462 \pm 0.3182$
GenEval dense	NFT, CFG 1	$+0.0143 \pm 0.1737$
GenEval dense	NFT + base guide, 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 + base guide, CFG 4	$+0.1337 \pm 0.0406$
Learned3 sum	NFT, CFG 4	$+0.0176 \pm 0.0357$
Learned3 sum	NFT, CFG 1	-0.0019 ± 0.0398
Learned3 sum	NFT + base guide, CFG 4	-0.0049 ± 0.0408

Table 6: SD3.5 at update 200. Standard deviations 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 descriptive SDs do not measure training variation or uncertainty of the mean.

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 included in training. All scores are higher-is-better, and bold includes ties at the displayed precision.

Objective	Method	CFG	Train 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
GenEval	NFT	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
OCR	NFT	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
Learned3	NFT	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

Table 7: **SD3.5 rewards and evaluation-seed variation.** Each cell shows the mean on the first line and ± 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.

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. Matching evaluator names or seeds does not imply identical prompts or generated images across evaluation suites.

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. We use saved `accuracy` flags from the official all-condition success rule (Ghosh et al., 2023); the local `strict_accuracy` field instead uses an exact-count rule. The detector and color-model inference are those of the saved evaluation. An evaluator excluded from training tests reward transfer; prompt holdout is specified separately by each evaluation suite.

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.

D.1 Qualitative comparisons

Figure 3 combines three Learned3 DrawBench samples at update 300 with three OCR online-evaluation previews at update 200. Learned3 candidates were ranked by GuidanceNFT's combined reward, $\mathrm{PickScore}/26 + \mathrm{HPSv2.1} + \mathrm{CLIPScore}$, 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 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. These are selected qualitative comparisons; aggregate performance is reported in Table 1.

Task	Prompts	NFT (%)	GuidanceNFT (%)	Difference ± 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	$+34.09 \pm 40.00$

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

Figures 7 and 8 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. These selected examples use paired noise within each column.

The full prompts for Figure 3, 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 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.
5. 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.
6. 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.

Figure 9 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. The six paired cases complement the aggregate GenEval comparison.

E Empirical Center 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:

$$D_q(F) = L_q(F) - L_q(C_{\mathrm{FM100}}), \qquad G_q(F) = \mathbb{E}_q \|F - C_{\mathrm{FM100}}\|^2 / d, \tag{44}$$

where d is the field dimension. The paired risk difference D_q cancels irreducible conditional variance and compares center MSEs (Equation 35); G_q measures field disagreement. Neither is absolute error to an oracle. Standard deviations use the 48 class or prompt means, averaging paired endpoints within each unit.

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

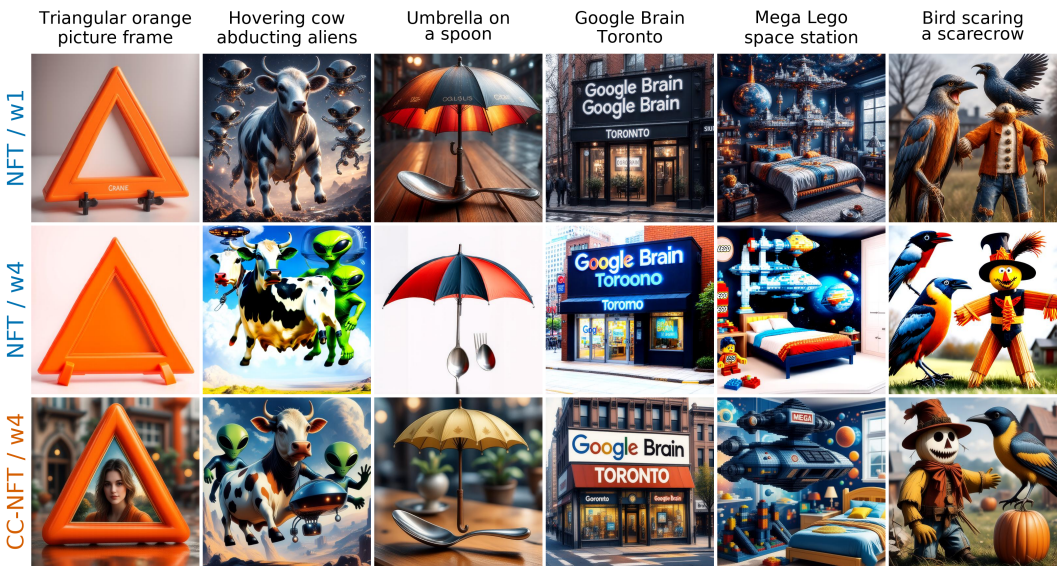


Figure 7: **Six-prompt Learned3 comparison.** Selected SD3.5 DrawBench samples at update 300, using 40 inference steps at 512×512 resolution. Rows show NFT at CFG 1, NFT at CFG 4, and GuidanceNFT at CFG 4 (labeled CC-NFT in the archived montage). Headings summarize each prompt; noise is paired across methods.



Figure 8: **Six-prompt OCR comparison.** Selected SD3.5 online-evaluation previews at update 200, using 40 inference steps. Rows show NFT at CFG 1, NFT at CFG 4, and GuidanceNFT at CFG 4 (labeled CC-NFT in the archived montage). Headings give the target text. The saved previews have native resolution 256×256; noise is paired across methods within each column.

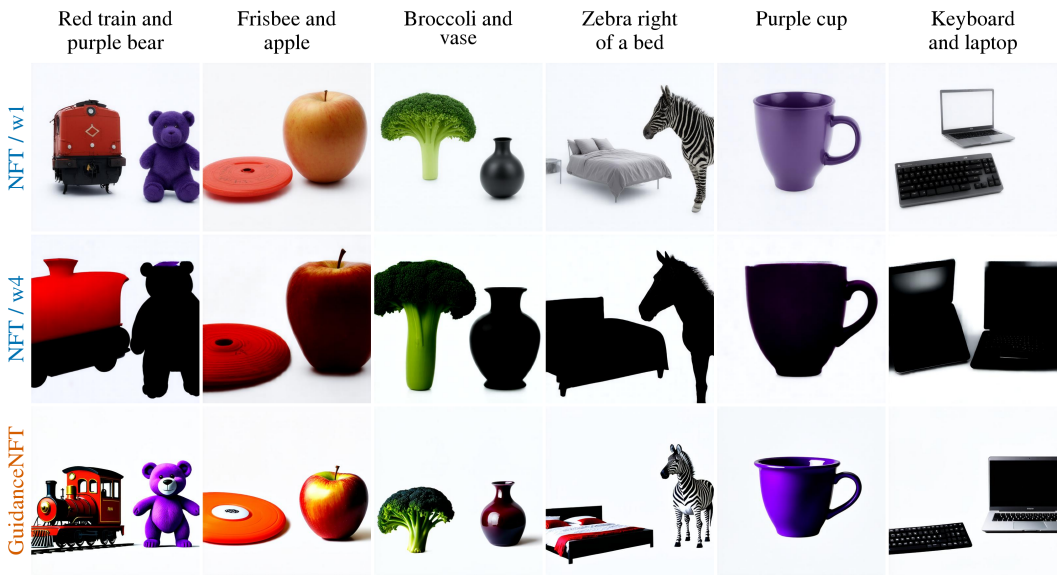


Figure 9: **Center correction preserves color and shading under CFG.** Selected GenEval-trained SD3.5 samples at update 200, with NFT at CFG 1 (top), NFT at CFG 4 (middle), and GuidanceNFT at CFG 4 (bottom). Columns pair latent noise across methods, using 40 steps and 512×512 resolution. Headings abbreviate the prompts.

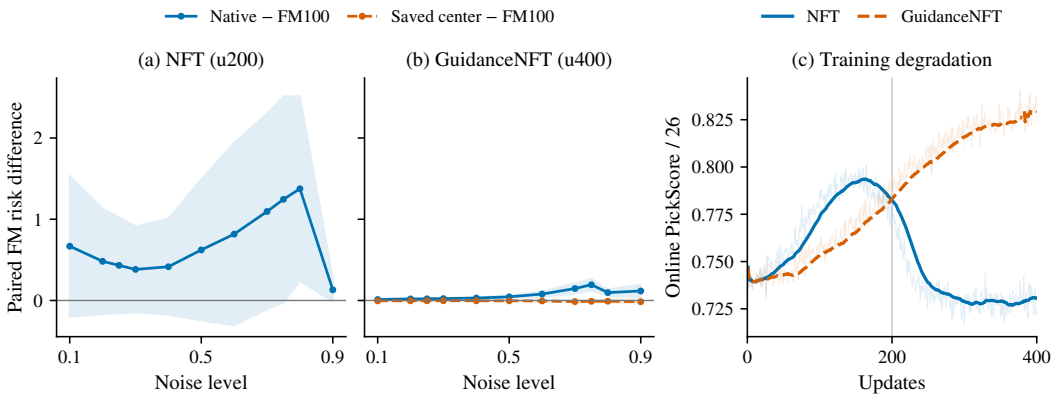


Figure 10: **A fitted center lowers held-out flow-matching risk on Model Guidance.** (a–b) Paired risk differences relative to each checkpoint's FM100 EMA: native field (blue) and saved GuidanceNFT center (orange). Bands show mean $\pm$ sample std over 48 class means. (c) Online rewards, with 30-update trailing means and faint raw traces; the vertical line marks NFT update 200. The two checkpoint populations are evaluated separately.

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 center has risk difference -0.00715 ± 0.00396, outperforming FM100. Directly comparing the native field with the saved center gives 0.07835 ± 0.02287, supporting separate center estimation on the same rollout population.

SD3.5. We evaluate the three five-step CFG-4 runs in Figure 2(b) at updates 0, 50, and 100. Each saved old-policy adapter generates its own deterministic DPM2 endpoints. The tested centers are the old conditional field, the analytic CFG-4 field, and the saved center EMA, respectively. Prompts

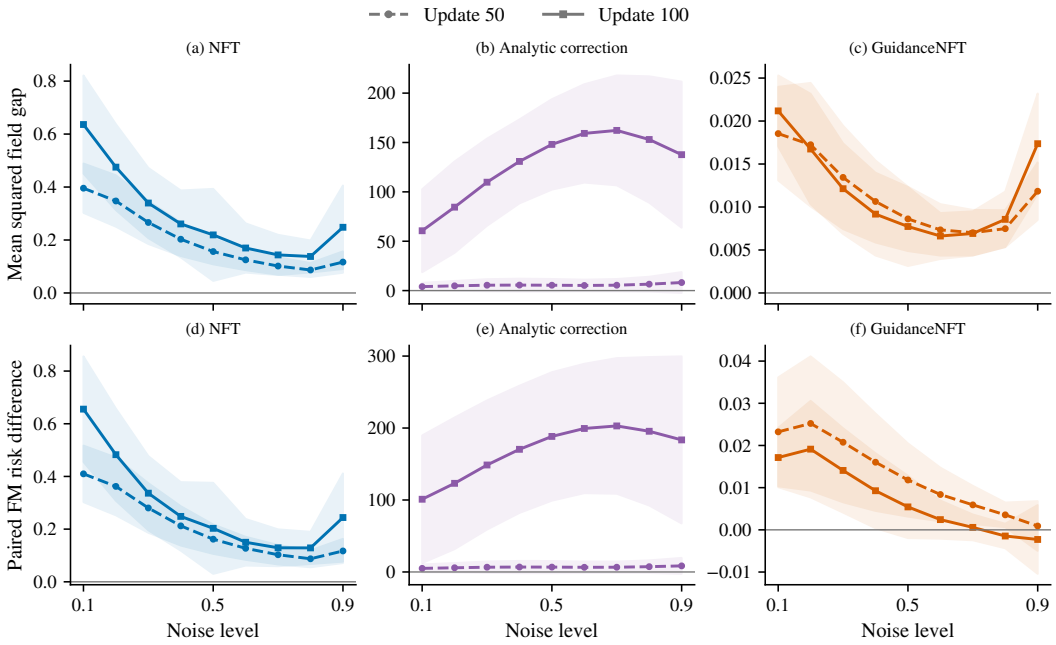


Figure 11: **Center correction controls the growing mismatch in SD3.5.** Columns compare no correction, analytic correction, and learned correction. Each run uses its own endpoints and FM100 EMA at update 50 (dashed) or 100 (solid). Top: squared field disagreement G_q. Bottom: paired FM-risk difference D_q, which can be negative. Bands show mean $\pm$ sample std across 48 prompt means. All policies and saved training centers are frozen during diagnosis.

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 center remain frozen.

Evaluation uses $\sigma \in \{0.1, 0.2, \ldots, 0.9\}$ with equal weights. At initialization the learned center 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 2(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.

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 the analytic center and from 0.3251 to 0.0953 with the old conditional center, 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 center mismatch and degradation. The diagnostic is not a center-only intervention: policies and endpoint populations evolve separately, and FM100 retains fitting error. For example, its held-out risk for analytic correction at update 100 falls from 39.97 to 33.29 between fitting steps 75 and 100. The exact displacement error also weights center error by $(1 - 2\alpha)/\beta$ (Equation 9).

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; unmarked pauses remain. These are typical recorded iteration

latencies, not synchronized total-training measurements. Component medians need not sum to the median total.

Task	Method	n	Rollout/ reward	Center	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×

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, center 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). Initialization and center warm-up are excluded; these are recorded typical latencies, not synchronized GPU timings or total training costs. Matched peak memory and inference throughput were not recorded; deployment requires no auxiliary-center pass.

Center initialization and refresh add training computation. The estimator is discarded at deployment, adding no inference passes. Matched peak-memory and inference-throughput measurements are unavailable.

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.

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

Table 10: Raw PickScore (↑), mean ± sample std. Each checkpoint uses three evaluation seeds and 6,000 images; standard deviations use the sample denominator $n - 1$. NFT uses its best observed checkpoint (200); GuidanceNFT uses 400. MG has external CFG 1. Bold indicates a larger point estimate. Standard deviations condition on the selected checkpoint and benchmark; they do not measure training repeatability or account for selection.

Each SiT checkpoint–CFG coordinate contains 6,000 images over three evaluation seeds. Table 11 reports raw PickScore over the retained checkpoint sequence, complementing the normalized trajectories in Figure 2(a).

Pretrained-empty SiT control. Replacing NFT's empty 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, the historical three-channel SiT guidance, ten SDE steps, batch size eight, and the same three evaluation seeds at updates 100–400 every 50 updates. Source population SD is multiplied by $\sqrt{3/2}$ for sample SD.

Selected PickScore examples. Figure 5 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 raw PickScore, mean

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

Table 11: SiT-XL/2 raw PickScore (↑) at every retained checkpoint (columns 100–400). Each cell shows the mean with ± 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 center for 100 updates and then trains with CFG-1.5 rollouts. Bold marks the best checkpoint within each method and evaluation-CFG row.

± sample std over three evaluation seeds (6,000 images), under the separate ten-step benchmark, rather than scores of the displayed samples. These selected examples do not estimate diversity or class coverage.

G ELF-B GUIDANCE SWEEP MEASUREMENTS

Table 12 reports every measured point in Figure 6, including the extended guidance weights. The evaluation protocol is given in Appendix C. These sweep measurements are separate from the 1,000-sample repetition diagnostic shown in the figure's text examples.

SCCFG w	Pretrained		LFRM-style		GuidanceNFT	
	Entropy	PPL	Entropy	PPL	Entropy	PPL
0.5	5.2363	33.3990	5.1210	19.6921	5.2071	20.7896
1	5.2147	29.5693	5.1034	17.5670	5.1923	18.3294
1.5	5.1964	27.4135	5.0902	16.6942	5.1814	17.3293
2	5.1784	26.0441	5.0793	16.0856	5.1727	16.7972
2.5	5.1640	24.9028	5.0705	15.6371	5.1621	16.3889
3	5.1506	24.0447	5.0637	15.3213	5.1572	16.2141
3.5	5.1421	23.5557	—	—	5.1522	16.1056
4	—	—	—	—	5.1490	16.0393
5	—	—	—	—	5.1454	16.1141

Table 12: Measured coordinates of the three ELF-B curves in Figure 6. Entropy is in nats; PPL is evaluated by GPT-2 large. Each point uses 1,024 samples, a shared evaluation seed, 32 SDE steps, $\gamma = 1.5$, and external CFG 1. LFRM-style and GuidanceNFT use update-100 checkpoints trained with SCCFG 1 and 3, respectively. Dashes denote guidance weights not evaluated for that curve; all displayed numbers are measured values, without interpolation.

H GSM8K REASONING PROTOCOLS

The reference checkpoints are from the ELF-B reasoning model used in LFRM (Anonymous, 2026), built on ELF (Hu et al., 2026). It uses Qwen-derived feature groups and a learned prompt encoder; inference does not execute the Qwen teacher Transformer.

LFRM's NFT implementation. The supplied adaptation uses the following detached correction for current, old, and fixed-reference roles p:

$$\widetilde{v}_p = v_p^{\text{main}} - \text{sg}\big[b_{\text{sc}}(1 - g^{-1})(v_p^{\text{main}} - v_p^0)\big], \qquad b_{\text{sc}} \in \{0, 1\}. \tag{45}$$

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 reference penalty with coefficient 0.1, averaged over latent coordinates and valid response tokens. Section 5.1 and Appendix I of Anonymous (2026) give the full objective and gradient routing. Our local native-field NFT and GuidanceNFT runs disable this reconstruction; their training-dynamics comparison concerns those local variants.

Evaluation and training protocols. The main comparison evaluates all 1,319 GSM8K questions with 64 steps, SCCFG 2, feature-group clocks $(2.5, 2, 1.5)$, and eight evaluation seeds. Pretrained 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. This uncertainty conditions on the selected checkpoints and answers.

Our NFT and GuidanceNFT runs use 32-step SCCFG-2 training rollouts and native main-pass velocities, with `training_velocity_reconstruction=false`. The LFRM reference 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. These are distinct training configurations evaluated under the same inference protocol. Pretrained is the supplied reference checkpoint.

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.

Ordinary SCCFG-2 checkpoint sweep. The sweep in Figure 4(c) and Table 13 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 has the higher peak under this synchronous protocol, whereas GuidanceNFT achieves higher accuracy under the main asynchronous evaluation and sustains performance through later training. NFT stops after its training rollouts produce only incorrect answers; missing updates 450/500 denote a stopped run. Checkpoint selection follows the conventions in Appendix C.

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

Table 13: Ordinary SCCFG-2 evaluation: 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 SD 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; update 400 is its last evaluated checkpoint. Dashes at updates 450 and 500 denote the stopped run. This protocol differs from the 64-step asynchronous main-table evaluation.

Endpoint-population matching. Some Qwen source paths fit the center to all generated endpoints while NFT retains mixed-outcome groups. Exact population matching, including reference answers, masks, and clocks, is unverified for every archived checkpoint. The reasoning results compare the documented training configurations; a center-only interpretation additionally requires the population condition in Appendix A.5.