GENERALIZED ON-POLICY SELF-DISTILLATION FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalar image rewards can guide diffusion model training, but translating them into effective parameter updates remains challenging. DiffusionOPSD offers an explicit target-fitting route, yet reward-gradient-based target construction still requires differentiable evaluators. Thus, we present GENOPSD, an on-policy self-distillation method for score-only feedback. It converts scores of decoded images from completed original trajectories and projected self-SDE branches into detached query-level targets, then fits the clean-latent targets with uniform positive clean-prediction MSE and refreshes the policy. Rewards affect target construction only, without gradients through the evaluator or decoder or a reward-weighted policy-gradient loss. On single-reward SD3.5-M, despite using no reward gradients, GenOPSD improves over DiffusionOPSD by 1.5% on ImageReward, with relative gaps of 3.0%, 5.6%, and 4.8% on PickScore, CLIPScore, and HPSv2.1. It also achieves the best GenEval and OCR results in the matched single-reward comparison, surpassing DiffusionNFT by 4.1% and 0.9%, respectively. Under multi-reward training, its scores remain within 7.8% of DiffusionNFT across measures. On Z-Image-Turbo, it improves over the base by 2.5% on GenEval and 1.2% on OCR. On Qwen-Image-Bench, it improves over the SD3.5-M base by 34.2% overall, with gains ranging from 8.9% to 62.5% across five categories, including 62.5% on Quality. Ablations support the roles of both state sources, prompt-population-relative signed preferences, coverage over native noise levels, and policy refresh; guidance tests favor the CFG 1 actor. We hope GenOPSD offers the community an effective method for training diffusion models.

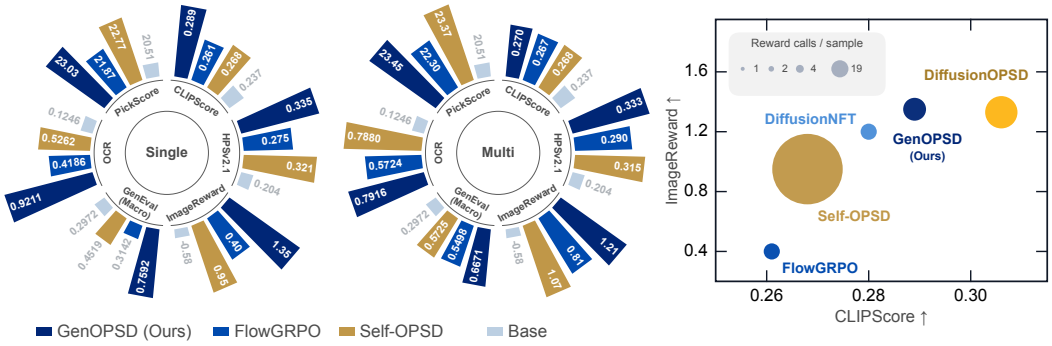


Figure 1: **Overview of GenOPSD.** Left: Per-metric results under single- and multi-reward training, compared with Base and representative baselines. Right: ImageReward versus CLIPScore, with marker size denoting reward calls per sample.

1 INTRODUCTION

Diffusion and flow models (Ho et al., 2020; Lipman et al., 2022; Yu et al., 2025) make a sequence of denoising predictions, yet many post-training objectives score only the decoded image (Zhang et al., 2026a). The evaluator may be a learned preference model, a discrete compositional check, an OCR system, or a remote API (Kirstain et al., 2023; Wu et al., 2023; Xu et al., 2023; Ghosh et al.,

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.

2023; Du et al., 2020). Its scalar output identifies desirable outcomes but provides no derivative with respect to an intermediate prediction. The resulting question is not only how to rank decoded images, but how to turn that ranking into a *target* that the denoiser can fit.

DiffusionOPSD (Zhou et al., 2026a) offers one answer when the reward is differentiable. It rolls out the current model, uses a reward gradient through the predicted clean image to construct an improved target, fits that target by supervised regression, and refreshes the policy that supplies the next states. This construct–fit–refresh structure is attractive because the feedback becomes explicit supervision rather than a reward-weighted likelihood update. Its target-construction step, however, depends on a derivative through the decoder and evaluator. For discrete rules or API-only scores, that derivative does not exist or is unavailable. Replacing the reward function with a different task score therefore does not make DiffusionOPSD applicable.

Black-box reward optimization is not new. DDPO and Flow-GRPO (Black et al., 2024; Liu et al., 2026) propagate outcome rewards through sampled denoising transitions with policy-gradient objectives. DiffusionNFT (Zheng et al., 2026) and Self-OPD (Zhang et al., 2026b) also use scalar feedback without differentiating the evaluator, through forward-process fitting and rewarded stochastic branches, respectively. These methods establish that black-box rewards can train diffusion models; they do not resolve the narrower design question here: *how can completed outcomes replace reward gradients while preserving an explicit detached-target and policy-refresh interface for self-distillation?*

We answer this question with GenOPSD. The generalization is operational: it retains fresh original trajectories and projected self-SDE branches, explicit detached query-level targets, supervised fitting, and policy refresh, while changing target construction from a reward gradient to scalar outcome scores. Because a scalar score alone does not identify a denoising direction, the method resolves three linked subproblems.

First, *which outcomes should supply feedback?* Each fresh original trajectory yields a final latent state. We obtain a second final latent state by sampling a projected self-SDE action at an early state and completing the sampler suffix (Zhang et al., 2026b). Both decoded images are scored only after their trajectories complete; no score is assigned to an unexecuted intermediate action. Second, *how should those scores reach model queries?* We normalize scores across the decoded images from each prompt occurrence to obtain detached prompt-population-relative signed preferences, and forward-noise the corresponding final latent states to form target-query rows. The current-point transport uses each selected state's signed preference together with detached current-actor and fixed-base velocity predictions to construct a detached target (Zheng et al., 2026). Third, *how should that target update the generator?* We fit the clean-latent targets with one uniform positive clean-prediction MSE step, then perform policy refresh and collect a fresh population for the next iteration. The evaluator and decoder are not differentiated, no reward-weighted policy-gradient loss is used, and no reward-selected parameter deployment is required.

Extensive experiments demonstrate the effectiveness of our method. On single-reward SD3.5-M, despite using no reward gradients, GenOPSD improves over DiffusionOPSD by 1.5% on ImageReward, with relative gaps of 3.0%, 5.6%, and 4.8% on PickScore, CLIPScore, and HPSv2.1. It also achieves the best GenEval and OCR results in the matched single-reward comparison, surpassing DiffusionNFT by 4.1% and 0.9%, respectively. Under multi-reward training, its results remain within 7.8% of DiffusionNFT across measures. On Z-Image-Turbo, it improves over the base by 2.5% on GenEval and 1.2% on OCR. On Qwen-Image-Bench, multi-reward GenOPSD improves over the SD3.5-M base by 34.2% overall and by 62.5% on Quality, 30.2% on Aesthetics, 28.3% on Alignment, 8.9% on Real-world Fidelity, and 27.4% on Creative Generation. Ablations show that removing a state source, changing preferences, or reducing noise coverage lowers reward by up to 42.3%, 93.3%, and 83.2%, respectively. Clipping negative preferences and freezing the sampler lower reward by 100% and 91.5%. CFG 1 training improves held-out GenEval accuracy by 26.3% and 15.0% at inference CFG 1 and 4.5.

Our contributions are:

- We introduce GenOPSD, an on-policy self-distillation method that trains diffusion models with scalar reward feedback.
- We convert paired final state scores into prompt relative preferences and detached query targets, then fit them with uniform positive MSE and refresh the policy.

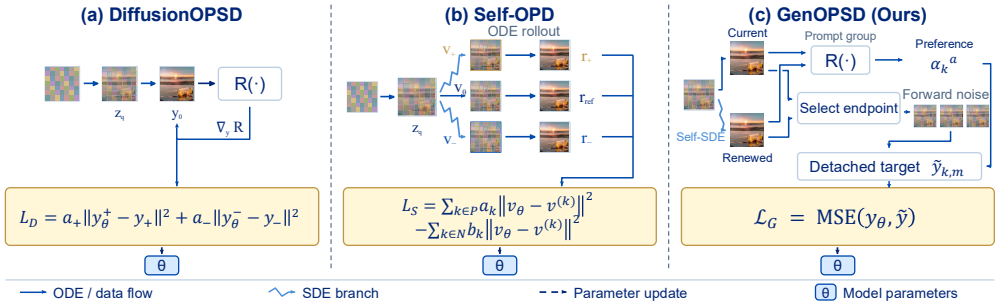


Figure 2: **Comparison of reward feedback.** DiffusionOPSD uses reward-gradient targets; Self-OPD scores completed SDE branches; GenOPSD scores decoded images from original trajectories and projected self-SDE branches, then forms detached targets at forward-noised queries.

- Matched comparisons and single-factor ablations evaluate GenOPSD and isolate the effects of projected self SDE branches, prompt relative preferences, and native noise level coverage.

2 Related Works

Reinforcement learning for diffusion models. Reinforcement learning provides a direct way to adapt diffusion and flow generators to preference- and task-based rewards. One family treats reverse denoising as a sequence of decisions: DDPO (Black et al., 2024) and Flow-GRPO (Liu et al., 2026) optimize sampled trajectories using terminal rewards and policy-gradient updates. Another family exploits the forward corruption or flow-matching formulation to turn reward-bearing outputs into regression signals. DiffusionNFT (Zheng et al., 2026) develops a forward-process objective, while Reinforce Adjoint Matching (RAM) (Bergmeister et al., 2026) and centered reward distillation (Zhu et al., 2026) derive reward-informed matching losses. ReFL (Xu et al., 2023) represents a complementary differentiable-feedback approach, propagating preference gradients through a predicted clean image or a truncated sampler. However, prior approaches either rely on trajectory-likelihood updates or differentiable reward gradients, or restrict score-only feedback to branch-level updates; GenOPSD instead converts black-box scores of completed on-policy outcomes into detached targets at student-dependent queries and fits them with a positive-MSE objective.

On-policy distillation. On-policy distillation trains a student with teacher guidance on states from its own trajectories. In language models, GKD trains on student-generated sequences with teacher feedback, while MiniLLM minimizes reverse KL on student rollouts (Agarwal et al., 2024; Gu et al., 2023). For diffusion and flow generators, DiffusionOPD distills task-specific teachers along student trajectories, DanceOPD routes student samples to capability fields, and D-OPSD and STEP-OPD extend distillation to continuous tuning and step distillation (Li et al., 2026b; Zhou et al., 2026b; Jiang et al., 2026; Wei et al., 2026). Teacher-free methods include DiffusionOPSD, which forms targets from image-reward gradients, and Self-OPD, which uses rewards from completed stochastic branches in a pull–push velocity objective (Zhou et al., 2026a; Zhang et al., 2026b). GenOPSD instead converts prompt-relative scores of decoded images from completed original trajectories and projected self-SDE branches into detached targets at queries formed by forward-noising selected final latent states, enabling positive-MSE fitting without reward gradients or branch-level updates.

3 Methodology

GenOPSD extends on-policy self-distillation to score-only feedback, turning scores of decoded images from completed original trajectories and projected self-SDE branches into detached query-level targets. Sections 3.1–3.3 describe the update interface and target construction; Section 3.4 covers fitting and policy refresh. Figure 3 and Algorithm 1 summarize the process for GenOPSD.

3.1 Score-only self-distillation interface

Let c be a text condition, θ the trainable model parameters, $\sigma \in (0, 1]$ a rectified-flow noise level, and $z_\sigma \in \mathbb{R}^d$ a latent with d scalar coordinates. The model predicts a velocity $v_\theta(z_\sigma, \sigma, c) \in \mathbb{R}^d$ (Lipman et al., 2022). At a fixed query $s = (c, z_\sigma, \sigma)$, abbreviate this prediction as $v_\theta(s)$ and define the

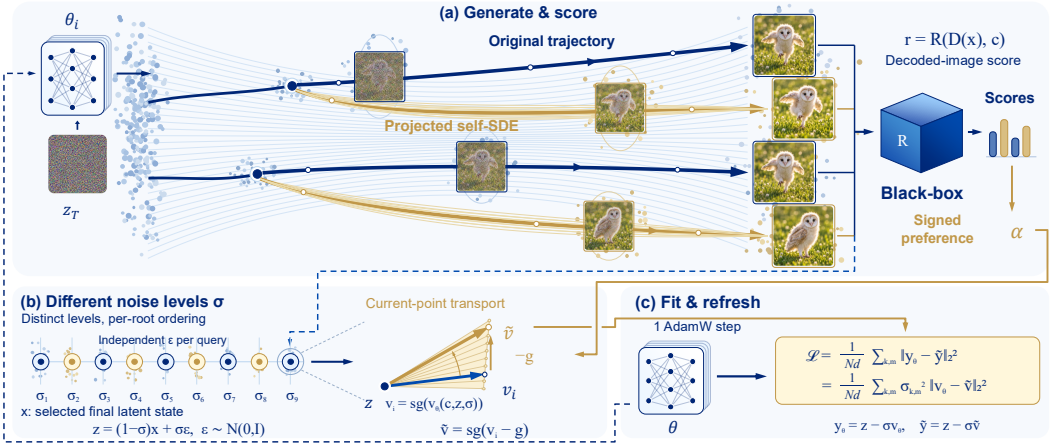


Figure 3: **Overview of GenOPSD.** GenOPSD scores decoded images from original trajectories and projected self-SDE branches, then fits detached preference-based targets in one step at queries from final latents forward-noised at nine native levels.

associated clean-latent prediction as

$$y_\theta(s) = z_\sigma - \sigma v_\theta(s). \tag{1}$$

A velocity target can therefore be converted to a clean-latent target at the same query. Appendix A.1 gives the corresponding map for a general affine path. Below, D maps a final latent state to a decoded image, and $\mathcal{R}(\cdot, c)$ returns a scalar reward for that image and text condition.

DiffusionOPSD constructs an explicit target using the derivative of the terminal reward with respect to a clean prediction (Zhou et al., 2026a). If the evaluator returns only a scalar $\mathcal{R}(D(x), c)$ for final latent state x, this derivative is unavailable. We retain the update structure, not the gradient-based target equation. At iteration $i \geq 0$, let θ_i be the current actor and π_{θ_i} its sampling policy. From B fresh original trajectories indexed by $k \in \{1, \ldots, B\}$, the collection operator produces paired final latent states $\mathcal{E}_i = \{(h(k), c_k, x_k^0, x_k^1)\}_{k=1}^B$, where $h(k)$ identifies the prompt occurrence and c_k its text. The two state sources are specified in Section 3.2. The score-only constructor uses these records, the evaluator, the current actor, and fixed pretrained parameters θ_{base} to form the query and target dataset:

$$\mathcal{D}_i = \mathcal{T}_{\text{score}}(\mathcal{E}_i, \mathcal{R}, \theta_i, \theta_{\text{base}}). \tag{2}$$

The dataset contains pairs $(s_{k,m}, \tilde{v}_{k,m})$ for each root k and row $m \in \{1, \ldots, L\}$. One AdamW step with persistent optimizer state gives $\theta_{i+1} = \text{Fit}_1(\theta_i; \text{sg}(\mathcal{D}_i))$, after which $\mathcal{E}_{i+1} = \text{Collect}(\pi_{\theta_{i+1}})$. Here sg stops gradients through the stored records, and Collect samples completed original trajectories and their projected self-SDE branches. The query states $s_{k,m}$ are obtained by forward-noising final latent states and are not recorded rollout states. This update operationally generalizes OPSD's feedback interface, rather than algebraically reducing its reward-gradient formula.

3.2 Completed outcomes and within-prompt preferences

For each fresh original trajectory k, let $x_k^0 \in \mathbb{R}^d$ be its final latent state and c_k its text condition. Starting from an early native state, we sample a projected self-SDE action and complete the sampler suffix to obtain a second final latent state $x_k^1 \in \mathbb{R}^d$ (Zhang et al., 2026b). We score the decoded image from each trajectory only after that trajectory is complete, using the same prompt-specific evaluator:

$$r_k^a = \mathcal{R}(D(x_k^a), c_k), \qquad a \in \{0, 1\}. \tag{3}$$

Thus $a = 0$ denotes the original trajectory's final latent state and $a = 1$ the SDE branch's final latent state. We do not score intermediate latents or assign one trajectory's reward to another.

For prompt occurrence h, let $\mathcal{G}_h$ be the multiset of all r_j^a with $h(j) = h$ and $a \in \{0, 1\}$. Write μ_h and s_h for its empirical mean and population standard deviation, counting each scored image once. The detached signed preference is

$$\alpha_k^a = \text{clip}\left(\frac{r_k^a - \mu_{h(k)}}{s_{h(k)} + \epsilon_A}, -\kappa, \kappa\right), \qquad \alpha_k^a \in [-\kappa, \kappa], \tag{4}$$

Algorithm 1 GenOPSD with score-only target construction

Require: Initial actor θ_0; fixed base reference θ_{base}; scalar evaluator $\mathcal{R}$; decoder D; prompt groups;
Initialize frozen behavior parameters $\theta_{\text{old}} \leftarrow \theta_0$.
for outer iteration $i = 0, 1, \ldots$ **do**
  Collect fresh trajectories under $\pi_{\theta_{\text{old}}}$, where $\theta_{\text{old}} = \theta_i$.
  for each behavior root k **do**
    Retain final latent state x_k^0; run one projected self-SDE branch to obtain x_k^1.
    Score decoded images $D(x_k^0)$ and $D(x_k^1)$ with the same prompt-specific evaluator.
  Compute detached within-prompt preferences α_k^a using Eq. (4).
  for each root k and target-query row m **do**
    Select one of the two final latent states and a native time; forward-noise the state using Eq. (5).
    Construct detached $\tilde{v}_{k,m}$ and $\tilde{y}_{k,m}$ using Eqs. (7)–(9).
  Fit Eq. (10) for one AdamW step to obtain θ_{i+1}; set $\theta_{\text{old}} \leftarrow \theta_{i+1}$.

where $\kappa > 0$ sets the clipping threshold, $\eta > 0$ is the independent transport gain used below, $\epsilon_A > 0$ prevents division by zero, and $\text{clip}(w, \ell, u) = \max(\ell, \min(w, u))$. This is a prompt-population-relative signal rather than a per-root winner label: both final latent states remain available even when one decoded image has a lower reward. Clipping and all subsequent target construction occur outside.

3.3 Query-level target construction

A scalar preference has no spatial derivative. We construct training queries by forward-noising a final latent state whose decoded image was scored. Each original trajectory yields L target-query rows. For row m, the source indicator $a_{k,m} \in \{0, 1\}$ selects one of the two final latent states, $\sigma_{k,m} \in (0, 1]$ is a native noise level, and independent Gaussian noise constructs

$$z_{k,m} = (1 - \sigma_{k,m}) x_k^{a_{k,m}} + \sigma_{k,m} \epsilon_{k,m}, \qquad \epsilon_{k,m} \sim \mathcal{N}(0, I_d), \tag{5}$$

Here I_d is the d-dimensional identity matrix. The resulting query is $s_{k,m} = (c_k, z_{k,m}, \sigma_{k,m})$. The reward is not evaluated at $z_{k,m}$; the score of the decoded image remains associated with the selected final latent state. Let T denote the number of distinct native noise levels among the L rows. Our selected configuration uses $L = T = 9$. If $\mathcal{J}_9$ is the set of nine selected native time indices, σ_j is the noise level at native index j, and $\rho_k : \{0, \ldots, 8\} \to \mathcal{J}_9$ is a root-specific ordering, then row m uses $\sigma_{k,m} = \sigma_{\rho_k(m-1)}$ and independently sampled forward noise. The sources alternate as

$$a_{k,m} = (k + i + m) \bmod 2, \qquad \big\{\#\{m : a_{k,m} = 0\}, \#\{m : a_{k,m} = 1\}\big\} = \{4, 5\}. \tag{6}$$

This balances source exposure across roots and updates. At each query, let $v_i = v_{\theta_i}(s_{k,m})$ and $v_{\text{ref}} = v_{\theta_{\text{base}}}(s_{k,m})$ be detached current-actor and fixed-base velocity predictions. The residual between the current clean-latent prediction and the selected final latent state, together with a detached scale, is

$$e_{k,m} = z_{k,m} - \sigma_{k,m} v_i - x_k^{a_{k,m}}, \qquad \omega_{k,m} = \max\left(\frac{\|e_{k,m}\|_1}{d}, \tau\right), \quad \tau > 0, \tag{7}$$

The quantity $\|e_{k,m}\|_1 / d$ is the mean absolute residual over the d latent coordinates, and $\tau > 0$ is its scale floor. The current-point transport direction $g_{k,m}$, base-reference coefficient $\lambda \geq 0$, and fixed velocity and clean-latent targets are

$$g_{k,m} = -\eta \sigma_{k,m} \alpha_k^{a_{k,m}} \frac{e_{k,m}}{\omega_{k,m}} + \lambda (v_i - v_{\text{ref}}), \qquad \lambda \geq 0, \tag{8}$$

$$\tilde{v}_{k,m} = \text{sg}(v_i - g_{k,m}), \qquad \tilde{y}_{k,m} = z_{k,m} - \sigma_{k,m} \tilde{v}_{k,m}. \tag{9}$$

For positive preference, the target moves the clean-latent prediction toward the selected final latent state; for negative preference, it moves away, subject to the residual scale and base-reference term. Appendix A.3 derives it from the full implicit-branch expression and states when the simplification holds. In particular, the reward enters through the detached α_k^a; it is not differentiated and does not multiply a policy likelihood.

Table 1: **Main results**. Under *specific*, ✓ denotes reward-specific training and × denotes one checkpoint evaluated across rewards. ‡ means the value is from another paper.

Model / Method	Specific	Trajectories (k)	Differentiable evaluators				Non-differentiable evaluators				
			PickScore	CLIPScore	HPSv2.1	ImageReward	GenEval				OCR
							Micro	Macro	Strict	Soft	
• SDXL‡	×	–	22.42	0.287	0.280	0.76	0.5104	0.5260	0.3318	0.1828	0.1177
• SD3.5-L‡	×	–	22.91	0.289	0.288	0.96	0.6935	0.7034	0.5759	0.6777	0.6955
• FLUX.1-dev‡	×	–	22.84	0.295	0.274	0.96	0.6302	0.6464	0.5448	0.6640	0.6054
• SD3.5-M w/o CFG‡	×	–	20.51	0.237	0.204	-0.58	0.2857	0.2972	0.1894	0.2314	0.1246
• + CFG‡	×	–	22.34	0.285	0.279	0.85	0.6569	0.6665	0.5307	0.6588	0.5761
• + FlowGRPO‡	✓	2,880	22.51	0.293	0.274	1.06	0.9344	0.9368	0.9060	0.9472	0.6424
• + FlowGRPO‡	✓	1,152	22.41	0.290	0.280	0.95	0.6713	0.6815	0.5457	0.6704	0.9087
• + FlowGRPO‡	✓	2,304	23.50	0.280	0.316	1.29	0.7197	0.7304	0.5149	0.5555	0.6920
• + DiffusionNFT‡	×	1,958	23.80	0.293	0.331	1.49	0.9299	0.9313	0.8843	0.9282	0.9176
SD3.5-M: single reward, 25 comparison units											
• + FlowGRPO	✓	28	21.87	0.261	0.275	0.40	0.3006	0.3142	0.2315	0.3442	0.4186
• + DiffusionOPSD	✓	28	**23.73**	**0.306**	**0.352**	1.33	-	-	-	-	-
• + DiffusionNFT	✓	28	23.00	0.280	0.323	1.20	0.7188	0.7293	**0.6844**	**0.8284**	0.9132
• + Self-OPD	✓	28	22.77	0.268	0.321	0.95	0.4376	0.4519	0.3495	0.4866	0.5262
• **GenOPSD (ours)**	✓	28	23.03	0.289	0.335	**1.35**	**0.7509**	**0.7592**	0.6745	0.7879	**0.9211**
SD3.5-M: multi-reward, 100 comparison units											
• + FlowGRPO	×	115	22.30	0.267	0.290	0.81	0.5344	0.5498	0.4227	0.5763	0.5724
• + DiffusionNFT	×	115	22.99	**0.271**	0.316	1.17	**0.6623**	**0.6761**	**0.5416**	**0.6730**	**0.8584**
• + Self-OPD	×	115	23.37	0.268	0.315	1.07	0.5574	0.5725	0.4543	0.5914	0.788
• **GenOPSD (ours)**	×	115	**23.45**	0.270	**0.333**	**1.21**	0.6533	0.6671	0.5118	0.6415	0.7916
Z-Image-Turbo											
• Z-Image-Turbo	×	–	22.86	0.276	0.296	0.96	0.7220	0.7304	0.5931	0.7166	0.8013
• + FlowGRPO	✓	14	22.90	0.275	0.300	1.01	0.7298	0.7383	0.6104	0.7415	0.7970
• + DiffusionOPSD	✓	14	**24.15**	**0.297**	**0.368**	**1.49**	-	-	-	-	-
• + DiffusionNFT	✓	14	22.48	0.281	0.267	0.70	0.7098	0.7209	**0.6524**	**0.8036**	0.7260
• + Self-OPD	✓	14	23.05	0.276	0.310	1.02	0.7224	0.7311	0.6067	0.7329	0.7989
• **GenOPSD (ours)**	✓	14	23.55	0.287	0.325	1.05	**0.7405**	**0.7484**	0.6117	0.7471	**0.8110**

3.4 Target fitting and policy refresh

With $N = BL$ root–query records and d output coordinates, we minimize a uniform positive clean-prediction MSE:

$$\mathcal{L}_{\text{GenOPSD}}(\theta) = \frac{1}{Nd}\sum_{k=1}^{B}\sum_{m=1}^{L}\|y_\theta(s_{k,m}) - \tilde{y}_{k,m}\|_2^2 = \frac{1}{Nd}\sum_{k=1}^{B}\sum_{m=1}^{L}\sigma_{k,m}^2\,\|v_\theta(s_{k,m}) - \tilde{v}_{k,m}\|_2^2\,. \tag{10}$$

Thus the σ-scaled velocity loss is exactly clean-prediction MSE at a fixed query; it is not unweighted velocity MSE. Only the current policy prediction in Eq. 10 carries gradients. One AdamW step is applied with persistent optimizer state. The fitted policy is copied to the frozen behavior slot with parameters $\theta_{\text{old}} = \theta_{i+1}$, old targets are discarded, and the next iteration collects a fresh population.

4 Experiments

To prove the effectiveness of GenOPSD, we compare held-out performance across differentiable and score-only reward, Qwen-Image-Bench results, and qualitative comparisons in Section 4.1. Section 4.2 examines the contributions of GenOPSD's outcome sources, preference construction, noise-level coverage, sampler refresh, and guidance. Appendix B details the backbones, objectives, checkpoints, and evaluation protocols.

4.1 Main Results

Main comparison. The matched single-reward SD3.5-M comparison directly tests how well GenOPSD preserves DiffusionOPSD's differentiable-reward performance under score-only feedback. Relative to DiffusionOPSD, GenOPSD is within 3.0% on PickScore, 5.6% on CLIPScore, and 4.8% on HPSv2.1, and is 1.5% higher on ImageReward. It also yields GenEval and OCR results from scalar feedback, for which DiffusionOPSD's gradient-based target construction is unavailable. Under multi-reward training, GenOPSD's results remain within 7.8% of DiffusionNFT across the reported measures. On Z-Image-Turbo, relative to the base model, GenOPSD improves GenEval by 2.5% and OCR by 1.2%. Table 1 gives the full comparison.

Training dynamics. Figure 4 shows the SD3.5-M training curves. GenOPSD improves steadily, although its PickScore and CLIPScore training rewards do not lead the baselines. Training rewards

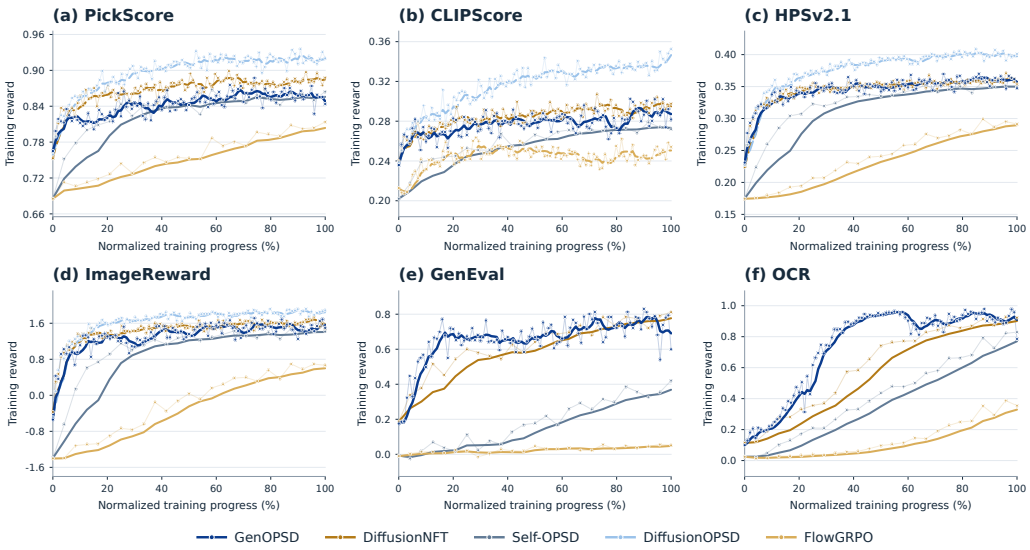


Figure 4: Training dynamics of GENOPSD compared with other methods.

alone do not determine held-out performance: GenOPSD remains within 3.0% on PickScore and 5.6% on CLIPScore of DiffusionOPSD, and exceeds it by 1.5% on ImageReward. These results highlight why training curves are not a substitute for held-out evaluation.

Qwen-Image-Bench. The multi-reward GenOPSD checkpoint improves Qwen-Image-Bench results relative to SD3.5-M base under the matched evaluation protocol. The relative gains are 34.2% overall, 62.5% on Quality, 30.2% on Aesthetics, 28.3% on Alignment, 8.9% on Real-world Fidelity, and 27.4% on Creative Generation. Figure 6, panel c, shows the comparison.

Qualitative results. The twenty matched cases show that GenOPSD produces clear, broadly prompt-aligned images across all seven evaluator groups. In *A fluffy baby sloth with a knitted hat trying to figure out a laptop*, GenOPSD visibly includes the laptop, whereas the other methods mostly show close-up sloth portraits; in *a photo of a cow right of a laptop*, it places the cow on the requested side, while Base and FlowGRPO blur the relation and DiffusionNFT reverses it. GenOPSD also renders *Street Art Rules* legibly in the graffiti mural, unlike the malformed lettering in Base. For *A picnic blanket with food and drinks, but no one around*, the blanket, food, and empty outdoor setting are all clear; its *retro travel poster with 1950s fonts, hand-drawn illustrations and soft pastels, with no modern UI or photographic elements* also has readable lettering and a coherent illustrated style. The grid nevertheless shows remaining errors: *A black apple and a green backpack* is too dark for reliable object and color binding, and *In late afternoon in January in New England, a man stands in the shadow of a maple tree* still shows autumn foliage. Overall, the visualizations are clear and show strong results across tasks, while exact color and seasonal constraints remain challenging; these selected cases complement, rather than replace, the held-out quantitative results.

4.2 ABLATION STUDY

Mechanism and time coverage. Figure 6(a,b) reports raw-actor training rewards at the end of training. Runs use the same original SD3.5-M base, 24 prompt groups with 12 roots each, a matched training-compute budget, 9 target-query rows, and 2 decoded-image scores per root; mechanism runs use sampling CFG 1. All reductions in these mechanism and time-coverage controls are relative to the shared 9-time reference. (i) **Using both final-state sources improves reward:** original-only and self-SDE-only controls are 42.3% and 13.3% lower. (ii) **Reward-aligned preferences are essential:** shuffled and uniform preferences are 93.3% and 72.8% lower. (iii) **Counterbalanced assignment helps:** random-balanced assignment is 19.4% lower. (iv) **Nine distinct times perform best:** using 1, 3, or 5 times lowers reward by 76.3%, 83.2%, or 60.6%.

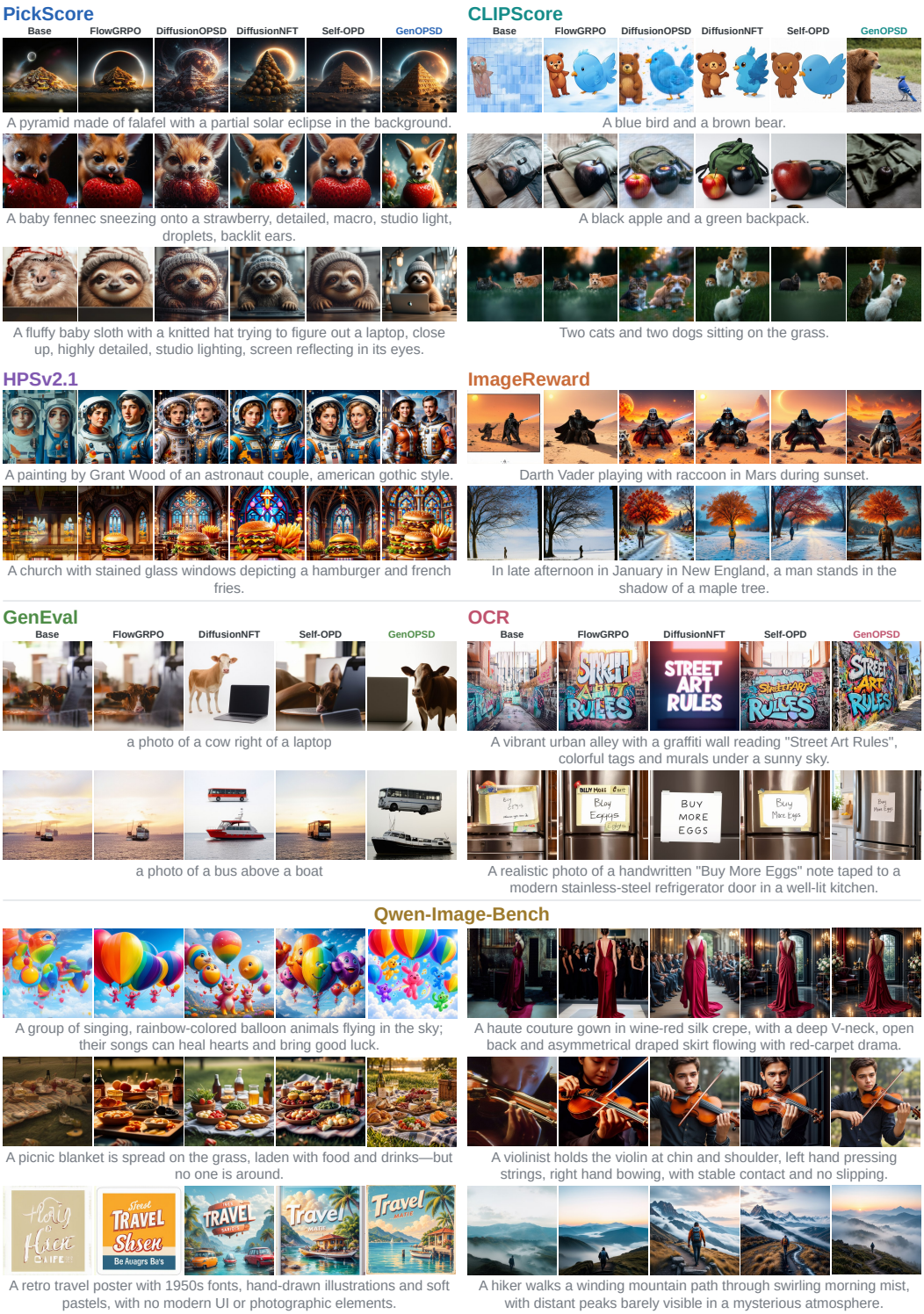


Figure 5: Qualitative comparison with other methods for SD3.5-M across different settings.

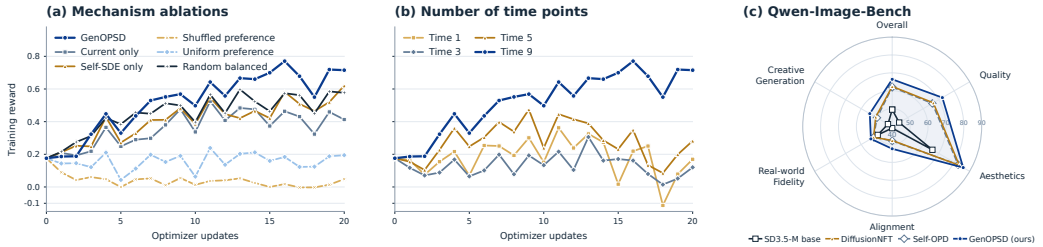


Figure 6: Ablations and Qwen-Image-Bench results of GENOPSD. (a) and (b) compare source, preference, assignment, and time-support controls. (c) shows Qwen-Image-Bench scores.

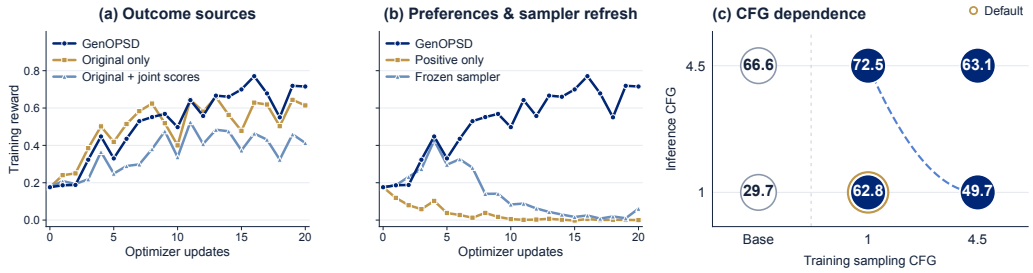


Figure 7: **Additional ablations and guidance.** (a) and (b) compare source, preference, and sampler controls at sampling CFG 1 against the shared 9-time reference. (c) shows held-out six-task GenEval accuracy for final actors across training and inference CFG.

Matched controls and guidance. Figure 7 reports additional controls and held-out GenEval results for the final actors trained under matched compute budgets. (i) **Both outcome sources contribute:** matched original-only transport is 14.2% below full GENOPSD. (ii) **Negative preferences matter:** clamping them to zero lowers reward by 100% relative to the signed-preference reference. (iii) **Refreshing the sampler supports learning:** freezing the U0 collector lowers the fitted actor reward by 91.5% relative to the refreshed reference. (iv) **Training at CFG 1 gives the stronger actor:** held-out GenEval accuracy is 26.3% higher than training at CFG 4.5 when inference CFG is 1, and 15.0% higher when inference CFG is 4.5. (v) **The CFG 1 actor improves on Base:** GenEval accuracy is 111.2% higher at inference CFG 1 and 8.8% higher at inference CFG 4.5; raising its inference CFG from 1 to 4.5 adds 15.5%.

5 CONCLUSION

We propose GENOPSD, an on-policy self-distillation method for score-only feedback. It uses prompt-population-relative signed preferences from decoded images of completed original trajectories and projected self-SDE branches to construct detached query-level targets, which are fit with uniform positive clean-prediction MSE before policy refresh. In matched single-reward SD3.5-M experiments, without reward gradients, GenOPSD outperforms DiffusionOPSD by 1.5% on ImageReward, while its relative gaps on PickScore, CLIPScore, and HPSv2.1 are 3.0%, 5.6%, and 4.8%, respectively. It also achieves the best GenEval and OCR results in the matched single-reward comparison, improving over DiffusionNFT by 4.1% and 0.9%, respectively. Under multi-reward training, it remains within 7.8% of DiffusionNFT across the reported measures. On Z-Image-Turbo, it improves GenEval by 2.5% and OCR by 1.2% over the base. On Qwen-Image-Bench, it improves over the SD3.5-M base by 34.2% overall, with gains ranging from 8.9% to 62.5% across five categories, including the largest gain on Quality. Ablations support the roles of both state sources, prompt-population-relative signed preferences, coverage across native noise levels, and policy refresh. We hope this work provides the community with an effective method for training diffusion models with score-only feedback.

AI USE STATEMENT

We used AI tools to assist with language editing and polishing of the manuscript, including improving clarity, grammar, and readability. We did not use generative AI tools for research tasks requiring disclosure under the ICLR AI policy, such as generating research data, developing the proposed method, designing or conducting experiments, or interpreting results. The authors reviewed and revised all AI-assisted text and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work studies reward-guided optimization of pretrained text-to-image diffusion and flow-matching models. It does not involve human-subject experiments or the collection of new personal data. Training and evaluation rely on existing model checkpoints and benchmarks. Like other image-generation models, the resulting models may produce inaccurate, biased, harmful, or misleading images. Optimizing against learned rewards or benchmark-based evaluators may also amplify their limitations. Our evaluation measures image quality and prompt alignment, including text rendering and compositional correctness; it is not a comprehensive assessment of safety, fairness, privacy, or broader societal impact.

REPRODUCIBILITY STATEMENT

We describe GenOPSD's reward-conditioned target construction and forward-only reward interface, together with model checkpoints, training configurations, prompt and evaluation-data provenance, preprocessing, sampler settings, and evaluation protocols in Appendices A, B, and B.7. Full result tables and supplementary comparisons are in Section 4.1.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

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.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

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

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

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

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. *arXiv preprint arXiv:2106.09685*, 2021.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, et al. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Niantong Li, Guangzheng Hu, Weixu Qiao, Ying Ba, Qichen Hong, Shijun Shen, Jinlin Wang, Fan Zhou, Jianye Kang, Xin Shang, et al. Qwen-image-bench: From generation to creation in text-to-image evaluation. *arXiv preprint arXiv:2605.28091*, 2026a.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026b.

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

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. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Stability AI. Introducing stable diffusion 3.5, 2024. URL https://stability.ai/news/introducing-stable-diffusion-3-5.

Qingyan Wei, Guangzhao Li, Xiaobing Tu, Yinggui Wang, Xiantao Zhang, Jinkui Ren, Xiaohong Liu, and Linfeng Zhang. Step-opd: Rethinking output targets and internal dynamics in on-policy distillation for diffusion models. *arXiv preprint arXiv:2608.04887*, 2026.

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

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26125–26135. IEEE, 2025.

Lei Zhang, Junjiao Tian, Zhipeng Fan, Kunpeng Li, Jialiang Wang, Weifeng Chen, Markos Georgopoulos, Felix Juefei-Xu, Manling Li, Julian McAuley, et al. Think in strokes, not pixels: Process-driven image generation via interleaved reasoning. In *European Conference on Computer Vision*, pp. 390–407. Springer, 2026a.

Shiyi Zhang, Mushui Liu, Yunze Tong, Wanggui He, Siyu Zou, Jinlong Liu, Yunlong Yu, Jian Song, Hao Jiang, Pipei Huang, et al. Self-opd: On-policy distillation for flow matching models without teacher. *arXiv preprint arXiv:2608.26872*, 2026b.

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*, volume 2026, pp. 134129–134150, 2026.

Wei Zhou, Xiongwei Zhu, Lingdong Kong, Bo Chen, Lei Zhang, Yongyuan Liang, Xiaoxia Hou, Ye Tian, Xian Sun, Yingshuo Wang, et al. On-policy self-distillation in diffusion models. *arXiv preprint arXiv:2608.24646*, 2026a.

Wei Zhou, Xiongwei Zhu, Zelin Xu, Bo Dong, Lixue Gong, Yongyuan Liang, Meng Chu, Leigang Qu, Lingdong Kong, Wei Liu, et al. Danceopd: On-policy generative field distillation. *arXiv preprint arXiv:2606.27377*, 2026b.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Diffusion reinforcement learning via centered reward distillation. *arXiv preprint arXiv:2603.14128*, 2026.

A Mathematical Details

This appendix makes the score-only target interface precise.

Clean prediction for a general affine schedule. Let $t \in [0,1]$ be continuous path time, $y \in \mathbb{R}^d$ a clean latent state, $\epsilon \sim \mathcal{N}(0, I_d)$ a Gaussian noise vector, and α_t, σ_t differentiable scalar path coefficients. The noised state z_t and its path velocity $v_t = \mathrm{d}z_t/\mathrm{d}t$, with y and ϵ fixed, are

$$z_t = \alpha_t y + \sigma_t \epsilon, \qquad v_t = \dot{\alpha}_t y + \dot{\sigma}_t \epsilon, \tag{11}$$

Here a dot denotes differentiation with respect to t. The path coefficient α_t is distinct from the reward preference α_k^a in Eq. 4. Define the determinant

$$\Delta_t = \alpha_t \dot{\sigma}_t - \dot{\alpha}_t \sigma_t. \tag{12}$$

Proposition A.1 (Affine clean-prediction map)**.** *If $\Delta_t \neq 0$, then a state and its velocity uniquely determine the clean prediction:*

$$y_t(z_t, v_t) = \frac{\dot{\sigma}_t z_t - \sigma_t v_t}{\Delta_t}. \tag{13}$$

For rectified flow, $\alpha_t = 1 - t$ and $\sigma_t = t$, giving $y_t = z_t - t v_t$. Thus t equals the main-text noise level σ, and replacing v_t with $v_\theta(s)$ recovers Eq. 1.

Proof. Equation 11 is the linear system

$$\begin{bmatrix} z_t \\ v_t \end{bmatrix} = \begin{bmatrix} \alpha_t & \sigma_t \\ \dot{\alpha}_t & \dot{\sigma}_t \end{bmatrix} \begin{bmatrix} y \\ \epsilon \end{bmatrix}. \tag{14}$$

The determinant is Δ_t. Inverting the matrix gives

$$\begin{bmatrix} y \\ \epsilon \end{bmatrix} = \frac{1}{\Delta_t} \begin{bmatrix} \dot{\sigma}_t & -\sigma_t \\ -\dot{\alpha}_t & \alpha_t \end{bmatrix} \begin{bmatrix} z_t \\ v_t \end{bmatrix}. \tag{15}$$

This yields Eq. 13. In the rectified-flow case, $\dot{\alpha}_t = -1$, $\dot{\sigma}_t = 1$, and $\Delta_t = 1$. □

For any two candidate velocities $v, v' \in \mathbb{R}^d$ at the same state z_t and time t, Eq. 13 gives

$$\|y_t(z_t, v) - y_t(z_t, v')\|_2 = \left|\frac{\sigma_t}{\Delta_t}\right| \|v - v'\|_2 . \tag{16}$$

Hence the native σ-scaled velocity MSE used by GenOPSD is exactly clean-prediction MSE for rectified flow. Target construction operates in velocity coordinates and is mapped to the clean target only after detachment.

Prompt-population-relative signed preferences. For prompt occurrence h, rewards r_j^a with $h(j) = h$ define the clipped signed preferences α_j^a in Eq. 4. For each target-query row, define the normalized preference $\bar{\alpha}_{k,m} = \alpha_k^{a_{k,m}}/\kappa$ and positive-branch mixing weight $q_{k,m}$. Thus

$$\alpha_k^a \in [-\kappa, \kappa], \qquad \bar{\alpha}_{k,m} \in [-1, 1], \qquad q_{k,m} = \frac{1 + \bar{\alpha}_{k,m}}{2} \in [0, 1]. \tag{17}$$

Rewards from both decoded images in the prompt occurrence determine these detached scalars. The mixing weight $q_{k,m}$ is not an additional reward or a sampling probability. Both final latent states remain eligible target sources regardless of reward.

Current-point transport identity. Fix row (k, m) and abbreviate $x = x_k^{a_{k,m}}$, $z = z_{k,m}$, $\sigma = \sigma_{k,m}$, $\alpha = \alpha_k^{a_{k,m}}$, and $q = q_{k,m}$. Write $v_i = v_{\theta_i}(s_{k,m})$, $v_{\text{old}} = v_{\theta_{\text{old}}}(s_{k,m})$, and $v_{\text{ref}} = v_{\theta_{\text{base}}}(s_{k,m})$ for detached predictions from the current actor, frozen behavior copy, and fixed base model. The frozen parameters θ_{old} are synchronized with θ_i before target construction. For a branch extrapolation parameter $\beta > 0$, distinct from AdamW's moment coefficients β_1, β_2, define

$$p^+ = \beta v_i + (1 - \beta) v_{\text{old}}, \qquad p^- = (1 + \beta) v_{\text{old}} - \beta v_i. \tag{18}$$

The branch residuals and their floored scales are

$$e^\pm = z - \sigma p^\pm - x, \qquad \omega^\pm = \max\left(d^{-1}\|e^\pm\|_1, \tau\right). \tag{19}$$

The full detached velocity-space transport direction and velocity target are

$$g_{\text{full}} = \eta\kappa\sigma \left(-q\frac{e^+}{\omega^+} + (1 - q)\frac{e^-}{\omega^-}\right) + \lambda(v_i - v_{\text{ref}}), \qquad \tilde{v} = \text{sg}(v_i - g_{\text{full}}). \tag{20}$$

Proposition A.2 (Synchronized one-step form)**.** *In the reported one-step protocol, the frozen behavior copy is synchronized with the current actor after every fit and before the next target construction. Therefore $v_{\text{old}} = v_i$ at target construction, and Eq. 20 reduces exactly to Eq. 8.*

Proof. Synchronization gives $p^+ = p^- = v_i$, hence $e^+ = e^- = e$ and $\omega^+ = \omega^- = \omega$. Since $-q + (1 - q) = 1 - 2q = -\alpha/\kappa$, the branch term becomes $-\eta\sigma\alpha e/\omega$. The base-reference term is unchanged. □

The reward appears only through α. Write $y_i = z - \sigma v_i$ for the current clean-latent prediction and $\tilde{y} = z - \sigma\tilde{v}$ for its detached target. For $\alpha > 0$ and $\lambda = 0$, their difference satisfies

$$\tilde{y} - y_i = -\eta\sigma^2\alpha\,\frac{y_i - x}{\omega}, \tag{21}$$

making the movement toward the selected final latent state explicit. The reference term adds a separate regularizing displacement when $\lambda > 0$.

Properties of the supervised update. For fixed stored clean-latent targets $\tilde{y}_{k,m}$, the gradient of Eq. 10 with respect to the trainable parameter vector θ is

$$\nabla_\theta \mathcal{L}_{\text{GenOPSD}} = \frac{2}{Nd}\sum_{k=1}^{B}\sum_{m=1}^{L} J_{\theta,k,m}^\top \left(y_\theta(s_{k,m}) - \tilde{y}_{k,m}\right), \quad J_{\theta,k,m} = \frac{\partial y_\theta(s_{k,m})}{\partial\theta} = -\sigma_{k,m}\frac{\partial v_\theta(s_{k,m})}{\partial\theta}. \tag{22}$$

The decoder, evaluator, sampler suffix, preference calculation, frozen behavior prediction, and base-reference prediction are absent from this fitting gradient.

Proof. Conditioned on the stored targets and query states, Eq. 10 is a squared error in the current clean-latent prediction. Differentiation gives Eq. 22; Eq. 1 gives the Jacobian relation. □

Reward and query counts.

Proposition A.3 (GenOPSD reward-call count)**.** *Let N_{reward} count scored decoded images and $N_{\text{target rows}}$ count forward-noised training queries in one update. Each of B original trajectories supplies its own decoded image and one decoded image from a completed projected self-SDE branch, while each root supplies L target-query rows. Therefore*

$$N_{\text{reward}} = 2B, \qquad N_{\text{target rows}} = LB. \tag{23}$$

The number T of distinct native noise levels among the L rows does not change the reward-call count.

Proof. Each original trajectory contributes two decoded images to the evaluator. Each target-query state is produced by forward-noising one of the corresponding final latent states and is not separately decoded or reward-scored. □

The selected configuration uses $L = T = 9$, with nine target-query rows at nine distinct native noise levels per root. Across one matched comparison unit, SD3.5-M comprises four physical updates of 288 roots each, giving $1,152$ roots, $2,304$ decoded images, and $10,368$ target-query rows. Z-Image-Turbo comprises eight updates of 72 roots each, giving 576 roots, $1,152$ decoded images, and $5,184$ rows. The projected self-SDE branch also requires a stochastic sampler suffix, so reward scoring, sampler-suffix forwards, target-query forwards, and fitting work are accounted for separately.

B Implementation Details

This appendix records the choices needed to reproduce the experiments. We separate model training, target construction, and evaluation so that matching one count, such as reward calls, is not confused with matching total compute.

B.1 Experimental setup

Backbones and training budget. We use SD3.5-M (Stability AI, 2024) at 512^2 and Z-Image-Turbo (Cai et al., 2025) at 1024^2. Fresh-trajectory budgets and prompt/root groupings are detailed in Appendix B.9. Unless otherwise specified, settings follow the official papers.

FlowGRPO samples a full stochastic SDE trajectory before PPO replay (Liu et al., 2026): SD3.5-M uses 10 transitions and replays 9; Z-Image-Turbo uses 9 and replays 8. It scores one endpoint image per trajectory and uses two raw optimizer steps per reported update. Table 1 reports cumulative trajectory counts; per-update grouping and optimizer-step accounting are detailed in Appendices B.9 and B.11.

Evaluators. We divide the training objectives into two blocks. The differentiable block contains PickScore (Kirstain et al., 2023), CLIPScore (Hessel et al., 2021), HPSv2.1 (Wu et al., 2023), and ImageReward (Xu et al., 2023). The non-differentiable block contains GenEval (Ghosh et al., 2023) and OCR (Du et al., 2020). GenEval is evaluated on $2,212$ images and reports image-micro correctness, the equal-weight macro average over its six tasks, strict correctness, and the dense soft score. OCR is evaluated on $1,018$ prompts. PickScore, CLIPScore, HPSv2.1, and ImageReward use DrawBench (Saharia et al., 2022). All matched SD3.5-M rows use deterministic flow sampling with 40 steps, guidance 1.0, and LoRA scale 1.0.

Methods. The matched block contains FlowGRPO (Liu et al., 2026), DiffusionOPSD (Zhou et al., 2026a), DiffusionNFT (Zheng et al., 2026), Self-OPD (Zhang et al., 2026b), and GenOPSD. Each method retains its canonical objective. Canonical DiffusionOPSD requires reward gradients and therefore has no GenEval- or OCR-trained row; these entries are marked not applicable rather than being filled with a different black-box algorithm. GenOPSD constructs nine forward-noised target-query rows at distinct native noise levels from its two final latent states. Prompt-population-relative signed preferences and counterbalanced source assignment define detached targets for one uniform positive clean-prediction MSE AdamW step per physical update. Rewards affect detached target construction only; they are neither differentiated nor used as direct policy-gradient multipliers.

Self-OPD preserves the released objective and branch construction: $K = 8$ stochastic branch samples per selected timestep, two weighted-without-replacement timesteps, global standard deviation per-prompt normalization, zero-advantage filtering and rebalancing, raw reward fusion, learning rate

Table 2: Public checkpoint provenance for the ‡ reference rows.

Reference row	Task or setting	Hugging Face repository	Loading type
SDXL	Base	stabilityai/stable-diffusion-xl-base-1.0	Full checkpoint
SD3.5-L	Base	stabilityai/stable-diffusion-3.5-large	Full checkpoint
FLUX.1-dev	Base	black-forest-labs/FLUX.1-dev	Full checkpoint
SD3.5-M w/o CFG	Guidance 1.0	stabilityai/stable-diffusion-3.5-medium	Full checkpoint
SD3.5-M + CFG	Guidance 4.5	stabilityai/stable-diffusion-3.5-medium	Same full checkpoint
FlowGRPO, ¿5k	GenEval	jieliu/SD3.5M-FlowGRPO-GenEval	SD3.5-M LoRA
FlowGRPO, 2k	OCR/Text	jieliu/SD3.5M-FlowGRPO-Text	SD3.5-M LoRA
FlowGRPO, 4k	PickScore	jieliu/SD3.5M-FlowGRPO-PickScore	SD3.5-M LoRA
DiffusionNFT, 1.7k	Multi-reward	worstcoder/SD3.5M-DiffusionNFT-MultiReward	SD3.5-M LoRA

3×10^{-4}, two raw optimizer steps per reported update, and EMA decay 0.9 at interval 8 (Zhang et al., 2026b). For the matched table, its released 16-image group is adapted to the prompt sizes specified in Appendix B.9. SD3.5-M uses 16 distributed sampling batches with per-rank batch size 9 and accumulation 8; the Z-Image port uses 12 batches with per-rank batch size 6 and accumulation 6. Both consume 48 prompts. Per-prompt statistics are grouped by prompt identity even when one distributed batch contains multiple prompt groups. The changed behavior group, lossless branch micro-batching, CPU staging, and checkpoint telemetry are explicit adaptations. Because Self-OPD has no official Z-Image release, that row is an official-core port.

Public checkpoints for the reference rows. Rows marked with ‡ in Table 1 preserve the source table's differentiable-evaluator scores. Their original iteration labels and checkpoint provenance are retained in Table 2; their GenEval/OCR entries are fixed-protocol re-evaluations. Table 2 records the public weights used for those re-evaluations. The three FlowGRPO entries are distinct reward-specific LoRA adapters (Liu et al., 2026). The SD3.5-M rows without and with CFG use the same base weights and differ only in guidance scale (Ho & Salimans, 2022).

The FlowGRPO model cards load the adapters on stabilityai/stable-diffusion-3.5-medium and use 40 inference steps with guidance 4.5. For the two base SD3.5-M rows we use guidance 1.0 and 4.5, respectively. The DiffusionNFT repository is an author-released *reproduced LoRA*; its model card does not assert that it is bit-identical to the paper's 1.7k checkpoint (Zheng et al., 2026). We therefore label results obtained from this checkpoint as an author-released reproduction.

For our completed re-evaluation, each downloadable checkpoint was regenerated under the evaluator-specific protocol in Appendix B.7. We report all four GenEval aggregates and OCR rather than copying the reference paper's single GenEval scalar. Gated Stability AI and FLUX repositories require accepting their licenses before downloading. Every download and evaluation manifest records the repository identifier, resolved revision, sampler, guidance, resolution, and adapter scale.

GenOPSD final latent states and noise levels. For each fresh original trajectory, GenOPSD retains the final latent state x_k^0 and completes the projected self-SDE branch from an early native state to obtain x_k^1. The decoded images from both trajectories are scored with the same prompt-specific evaluator, giving rewards r_k^0 and r_k^1. These rewards define the prompt-population-relative signed preferences α_k^0 and α_k^1. Both final latent states remain eligible target sources.

Each root has nine target-query rows. The root-specific permutation ρ_k assigns row m to native time index $\rho_k(m-1)$. At $T = 9$, the rows use nine distinct native noise levels and independent forward noise. The two final latent states provide five and four rows in a counterbalanced source assignment across roots and updates. The $T = 1, 3, 5$ controls reuse native noise levels while retaining nine rows and the same decoded-image scoring budget. Query rows are model evaluations, not reward calls.

Rollout and sampler continuation. The SD3.5-M collection path uses deterministic ten-step DPM-Solver++-2M at 512×512 resolution and guidance scale 1.0 (Lu et al., 2025). The implementation records the native trajectory and the solver history needed to resume from an early query. For the projected self-SDE branch, the collector applies the projected stochastic action at that query and completes the sampler suffix using the actual solver state. This yields a second final latent state under the declared proposal rather than scoring an isolated intermediate latent.

The decoded images from the original trajectory and projected self-SDE branch are scored with the same prompt-specific evaluator. Target-query states are then constructed by forward-noising

the selected final latent state at one of nine distinct native noise levels in the selected nine-row configuration. The query states are not decoded or reward-scored. Target-query forwards, stochastic suffix forwards, and reward calls are recorded as different cost categories.

Stored targets and model update. For each root k and row m, the implementation stores the forward-noised state $z_{k,m}$, selected final latent state $x_k^{a_{k,m}}$, detached preference $\alpha_k^{a_{k,m}}$, detached velocity predictions v_{old}, v_i, and v_{ref}, and detached velocity target $\tilde{v}_{k,m}$. The clean-latent target is $\tilde{y}_{k,m} = z_{k,m} - \sigma_{k,m}\tilde{v}_{k,m}$. No decoder or reward activation is retained for fitting.

Target construction uses the synchronized current-point direction in Eqs. 7–9, equivalently the full implicit-branch form in Appendix A.3. It runs without gradients. Fitting applies a σ-scaled velocity MSE, which equals clean-prediction MSE by Eq. 1. One AdamW step is applied per fresh population with optimizer state carried forward. The updated actor is copied to the frozen behavior slot; the next iteration collects new trajectories and their final latent states before constructing new targets.

Main training configuration. Table 3 lists the GenOPSD implementation settings for SD3.5-M. The selected protocol uses nine target-query rows over nine distinct native noise levels per root. The same score-only target construction can use other noise-level supports without changing its feedback interface; backbone-specific behavior-root and image counts are reported separately.

Table 3: Primary SD3.5-M GenOPSD implementation settings.

Item	Setting
Backbone	SD3.5-M
Resolution	512×512
Trainable parameters	LoRA (Hu et al., 2021), rank 32, alpha 64
Training precision	FP16
Decoded images scored / trajectory	one from the original trajectory, one from the projected self-SDE branch
Collection sampler	deterministic DPM-Solver++-2M (Lu et al., 2025), 10 steps
Training guidance scale	1.0
Target-query rows	nine rows across nine distinct native noise levels
Reward preference	prompt-population-relative signed, clipping threshold $\kappa = 5$, transport gain $\eta = 1$
Population stabilizer	$\epsilon_A = 10^{-4}$
Residual-scale floor	$\tau = 10^{-5}$
Branch extrapolation parameter β	1.0
Base-reference coefficient λ	10^{-4}
Optimizer	AdamW, persistent state
Learning rate	3×10^{-4}
AdamW (β_1, β_2)	$(0.9, 0.999)$
Weight decay / optimizer epsilon	10^{-4} / 10^{-8}
Actor steps / fresh population	1
LoRA scale at evaluation	1.0

B.7 Evaluation protocols

Common generation protocol. Final evaluation is performed on fixed held-out prompt sets rather than on the training prompt stream or on training-loop proxy averages. For SD3.5-M, each held-out item produces exactly one decoded image with the deterministic flow sampler using 40 inference steps at 512×512 resolution. The LoRA inference scale is fixed to 1.0. The same generation and evaluator settings are used for all methods within a matched comparison.

GenEval. The GenEval test set contains 553 prompts, with four decoded images per prompt, for a total of $2,212$ images (Ghosh et al., 2023). The metadata cover six tasks: single-object, two-object, counting, colors, position, and color-attribute composition. The evaluator follows the published GenEval scoring stack: it uses the held-out GenEval metadata, the published Mask2Former Swin-S detector checkpoint, and the OpenAI CLIP ViT-L/14 classifier for attribute checks. The implementation wraps this evaluator to generate the images and to preserve per-image records; it does not replace the detector or CLIP-based task rules with a learned judge.

We report four complementary quantities. Image-level micro accuracy is the mean standard correctness over all $2,212$ images. Six-task macro accuracy is the unweighted mean of the six task-level accuracies. Strict accuracy uses the strict per-image correctness field, while soft accuracy is the mean dense score returned by the evaluator. The per-task and per-image records are saved alongside these aggregates, so the reported values can be independently recomputed from the same evaluation output.

Unless otherwise noted, "GenEval" in the abstract and main text denotes the benchmark's official six-task macro accuracy. Table 1 also reports the micro, strict, and soft aggregates.

OCR. The OCR test set contains $1,018$ held-out prompts and one decoded image per prompt. The evaluator uses the English PaddleOCR recognizer without angle classification (Du et al., 2020). For each prompt, let t be the nonempty target string extracted from the quoted text and $\hat{t}$ the concatenated recognized text across detected regions. Both strings are lowercased with whitespace removed. If t is contained in $\hat{t}$, the score is 1. Otherwise, with $d_{\mathrm{Lev}}(\hat{t}, t)$ denoting the Levenshtein distance between the normalized strings and $|t|$ the target length, the score is

$$\mathrm{OCR}(\hat{t}, t) = 1 - \frac{\min\{d_{\mathrm{Lev}}(\hat{t}, t),\ |t|\}}{|t|}.$$

The final OCR result is the mean over all $1,018$ prompts; standard deviation, standard error, and per-image scores are also retained. The OCR score is used only as a detached evaluator output and is never differentiated through during model training.

Evaluation isolation. GenEval and OCR retain evaluator-specific prompt sets, scoring rules, and reward-specific RL checkpoints. They must be trained separately; evaluating a PickScore-, CLIPScore-, HPSv2.1-, or ImageReward-trained checkpoint on GenEval or OCR does not fill the reward-specific GenEval/OCR cells. The four GenEval columns are computed from one GenEval-trained checkpoint. Training-time candidate scores are never substituted for final held-out metrics.

B.8 Qwen-Image-Bench protocol and replacement judge

Qwen-Image-Bench evaluation uses the official $1,000$-example test split, retains its official IDs, and uses the English field `prompt_en` (Li et al., 2026a). The benchmark has five Level-1 dimensions (Quality, Aesthetics, Alignment, Real-world Fidelity, and Creative Generation), nested through the official hierarchy to 56 Level-3 criteria. We pass the official checklist text, user prompt template, and system prompt verbatim, without paraphrasing or regenerating the rubric. Official $0/1/2$ facet scores are mapped to $0/60/100$ and averaged through the published Level-3–Level-2–Level-1 hierarchy.

For every example and checkpoint, SD3.5-M produces one 512×512 decoded image with the deterministic flow sampler for 40 steps, guidance scale 1.0, FP16 inference, and noise level 0.7. The replacement judge uses model revision `gpt-5.5-2026-04-24`, as a strong LLM will be more robust Li et al. (2026a). The $1,000$-example base evaluation was sharded across four independent resumable queues; failed or invalid responses were retried with bounded backoff. This API rejects temperature 0, exposes neither top-k, top-p, nor repetition penalty, and uses its default reasoning behavior.

B.9 Baseline matching

The main table reports cumulative fresh trajectories in thousands. One matched comparison unit contains $1,152$ SD3.5-M or 576 Z-Image-Turbo roots; therefore, the 25- and 100-unit SD3.5-M budgets are $1,152 \times 25 = 28,800$ and $1,152 \times 100 = 115,200$, and the 25-unit Z-Image-Turbo budget is $576 \times 25 = 14,400$.

Following the prompt-grouping convention in the official Flow-GRPO implementation, we use 48 prompt draws with 24 fresh trajectories per draw for SD3.5-M ($48 \times 24 = 1{,}152$) and 48 draws with 12 trajectories per draw for Z-Image-Turbo ($48 \times 12 = 576$) in each reported update. The 48×24 setting follows the official implementation's reference configuration; 48×12 is our adaptation to match the fresh-trajectory budget for Z-Image-Turbo, not a claim about the official default.

GenOPSD's exploration diversity does not rely on a large GRPO-style group of samples for each prompt. Instead, it compares each original trajectory with its completed projected self-SDE branch. We retain GenOPSD's native prompt/root geometry. Each physical SD3.5-M update uses 24 prompt occurrences and 12 roots per occurrence, or 288 fresh trajectories. Four updates form one comparison unit of $1,152$ trajectories. For Z-Image-Turbo, each physical update uses 12 prompt occurrences and 6 roots per occurrence, or 72 fresh trajectories. Eight updates form one comparison unit of 576 trajectories. These are batch sizes for one physical optimizer update, not for a full comparison unit. Each trajectory contributes two decoded images to the evaluator.

All initial and reported-update boundary checkpoints should be retained, together with any native intra-update checkpoints.

B.10 GenOPSD ablation protocol

The ablations use a shared fresh-base nine-level GenOPSD reference and single-factor controls. We distinguish the matched nine-level comparisons from the complementary three-level mechanism sweep so that changes in source or preference are not conflated with changes in noise-level support.

Shared reference and training budget. Every arm starts from the original SD3.5-M base and has fresh optimizer state; no arm resumes from another arm's checkpoint. Each optimizer update samples $G = 24$ prompt occurrences with $H = 12$ fresh original trajectories (roots) per occurrence, for

$$B = GH = 288 \tag{24}$$

fresh original trajectories per physical optimizer update. Four such SD3.5-M batches make the $1,152$-root comparison unit used above. For every root k, we retain the original final latent x_k^0 and complete the projected self-SDE branch to obtain x_k^1. Both decoded images are scored by the same prompt-specific evaluator, as in Eq. 3; the $2H = 24$ decoded-image rewards within each prompt occurrence define the prompt-population-relative signed preferences in Eq. 4. Each update therefore scores $2B = 576$ decoded images and constructs $9B = 2{,}592$ target-query rows. The ablation runs for 20 actor optimizer updates, totaling $5,760$ fitted roots, $11,520$ scored decoded images, and $51,840$ target-query rows per arm.

Query construction and noise levels. All arms retain $L = 9$ rows per root and change only the number T of distinct native noise levels. For each $T \in \{1, 3, 5, 9\}$, let $\mathcal{J}_T$ contain the T selected native time indices, let σ_j be the native noise level at index $j \in \mathcal{J}_T$, and let $\rho_k : \{0, \dots, T-1\} \to \mathcal{J}_T$ give a root-specific ordering. For row $m \in \{1, \dots, 9\}$, the selected noise level is

$$\sigma_{k,m} = \sigma_{\rho_k((m-1) \bmod T)}, \qquad T \in \{1, 3, 5, 9\}. \tag{25}$$

Thus $T = 1, 3, 5$ reuse noise levels across the same nine rows, while $T = 9$ uses nine distinct levels. Every row still receives independent forward noise in Eq. 5; rows are model queries and do not trigger additional decoding or reward scoring. The detached target construction and positive MSE in Eqs. 7–10 are otherwise unchanged.

Matched nine-level controls. The reference uses prompt-population-relative signed preferences from both scored decoded images and the counterbalanced source allocation in Eq. 6. The five controls rerun against this $T = 9$ reference retain its prompt/root/noise plan, two-image scoring budget, nine target-query rows, and all target/loss settings except for the named intervention:

- **Current-only:** set $a_{k,m} = 0$ for all nine rows, so every target query is formed from x_k^0. Both decoded images are still scored, and preferences remain normalized over rewards from both final latent states in the prompt occurrence. This removes projected self-SDE branch supervision without changing the scoring or preference population.
- **SDE-only:** set $a_{k,m} = 1$ for all nine rows, so every target query is formed from x_k^1. Both decoded images are still scored and the same prompt-population-relative signed preferences are retained.
- **Shuffled preference:** keep the reference source and noise-level assignments, but permute the detached signed preferences among the $2H$ scored decoded images within each prompt occurrence. This preserves the within-prompt label distribution and score budget while breaking the reward-to-image correspondence.
- **Uniform preference:** set every detached signed preference α_k^a to zero, giving $q_{k,m} = 1/2$ for every target-query row, while preserving both final latent states, source assignment, noise-level support, and query count.
- **Random-balanced source assignment:** randomize the nine row positions assigned to the two final latent states while preserving a four/five split for every root. The number of projected self-SDE branch rows alternates between five and four across roots and updates, keeping aggregate source exposure balanced; both decoded images remain scored.

These controls therefore alter how already-scored outcomes supervise the same detached target interface; none places a reward value in the differentiated fitting loss.

Complementary three-level controls. A separate base/U0 mechanism sweep includes the matched original-only transport control and a fixed five-to-four source allocation. The original-only control uses x_k^0 for all nine rows and recomputes the prompt-population-relative signed preferences using only original-trajectory rewards; both decoded images are still scored to retain the two-image budget. The fixed allocation assigns five rows to x_k^0 and four to x_k^1 at fixed row positions rather than counterbalancing those positions across roots and updates. Both controls use $T = 3$ and are not part of the six-curve matched $T = 9$ comparison. They remain auxiliary mechanism diagnostics and are not interpreted as matched nine-level effects.

Each arm is compared at its declared update boundary and fresh-root budget. Training-population rewards diagnose optimization but do not replace held-out generation metrics. The systems profile separately records decoded-image scores, sampler-suffix denoiser forwards, target-query forwards, fitting work, memory, and elapsed time.

Additional matched nine-level arms. The subsequent source, preference, and refresh controls also start from the SD3.5-M base with the same 20-update. Original-only transport forms every query from the original final latent state and normalizes preferences using only original-trajectory rewards, while still scoring both decoded images. Positive-only clamps negative signed preferences to zero during target construction; its learning rate is unchanged. Frozen-sampler retains a separate U0 sampling adapter while the fitted actor and its current behavior copy continue to update. Its curve uses separate read-only measurements of the current fitted actor on the matched update-specific prompt/root/noise plan, adding $5,760$ measurement images over the twenty non-initial boundaries. These diagnostic images are accounted for separately from the shared fitting budget.

Crossed guidance evaluation. The complete GenOPSD actors trained with sampling CFG 1 and 4.5 are evaluated at their fixed twenty-update boundary, each at inference CFG 1 and 4.5. Training-sampling guidance changes original collection and the completed projected self-SDE branch; fitting remains conditional-only. Each cell uses 553 held-out prompts with four indexed samples each, 512-pixel images, forty-step deterministic flow sampling and raw LoRA scale 1. The training prompt set and held-out prompt set have no normalized exact-prompt overlap. Standard GenEval correctness is averaged equally over the six tasks; micro, strict and dense scores are retained separately. The four new cells generate $8,848$ images.

Profile definitions. Matched profiles record endpoint scoring, sampler-suffix denoiser forwards, nine target-query forwards, actor backward/recompute, peak memory, and elapsed time. Report DiffusionOPSD's reward-backward graph separately; scorer counts alone do not establish a speedup.

B.11 Efficiency accounting and scope

Table 1 reports cumulative fresh trajectories in thousands, while Table 4 gives reward-scored decoded images per trajectory. Component-scorer calls, target queries, denoiser forwards, backward passes, optimizer steps, memory, and runtime are separate units.

FlowGRPO scores only each trajectory's decoded endpoint (Liu et al., 2026); intermediate SDE states are not decoded, and PPO replay does not rescore images. In the current no-cert DiffusionOPSD implementation, each behavior trajectory triggers four reward evaluations (Zhou et al., 2026a): one decoded endpoint and three target-construction evaluations. With `opa_n_ascent=2`, the positive and negative directions share the anchor gradient and each adds one evaluation. DiffusionNFT scores only the original endpoint (Zheng et al., 2026). Self-OPD scores each original endpoint before applying the released zero-advantage filter (Zhang et al., 2026b). Each retained sample adds 18 scores (two timesteps times $K + 1 = 9$ branch/reference outputs). Since filtering depends on rewards, the table gives the integer no-filter maximum; each run manifest records the actual retained-sample and score counts. GenOPSD scores decoded images from two final latent states. Its nine target-query rows and projected self-SDE sampler-suffix denoising are separate costs; multiple component-scorer calls do not add images.

Equal decoded-image counts do not imply equal compute. GenOPSD additionally performs nine target-query forwards per trajectory and sampler-suffix denoising on projected self-SDE branches,

Table 4: Reward-scored image instances per fresh trajectory

Method	Scored images / fresh trajectory
FlowGRPO	1
DiffusionOPSD	4
DiffusionNFT	1
Self-OPD ($K = 8$)	≤ 19
GENOPSD (ours)	2

while DiffusionOPSD constructs reward-gradient graphs. Samplers, proposals, replay schedules, and backward work also differ.

C QUALITATIVE ABLATION COMPARISONS

Figures 8 and 9 complement the ablation diagnostics in Section 4.2 with custom prompts covering landscapes, environmental portraits, illustrations, everyday scenes, and imaginative compositions. The examples are selected from a 40-prompt pool to illustrate different content types; the two figures share several cases and are not independent evaluations.

Generation settings. We use the raw SD3.5-M actors after 20 physical optimizer updates from the GenEval-trained ablations in Appendix B.10. Images are generated at 512×512 resolution using 40-step deterministic flow sampling and LoRA scale 1. Both training-sampling CFG and inference CFG are 1. The short phrases beneath the image groups summarize the full generation prompts. All images are shown without cropping or retouching; noisy and nearly blank outputs are actual model generations rather than missing results.

Controls shown in the figures. *Joint-score original* constructs queries from original-trajectory final latent states while normalizing preferences over rewards from both the original trajectory and its projected self-SDE branch. *Original-only* also uses only original final latent states for queries, but normalizes preferences using only original-trajectory rewards; both decoded images are still scored. Thus these two controls differ in their preference population. *Positive-only* clamps negative signed preferences to zero without retuning the learning rate. *Frozen sampler* keeps the U0 collector fixed while continuing to train the actor; the figure shows outputs from that fitted actor, not from the frozen collector. Full GENOPSD uses both final latent states, prompt-population-relative signed preferences, and population refreshes.

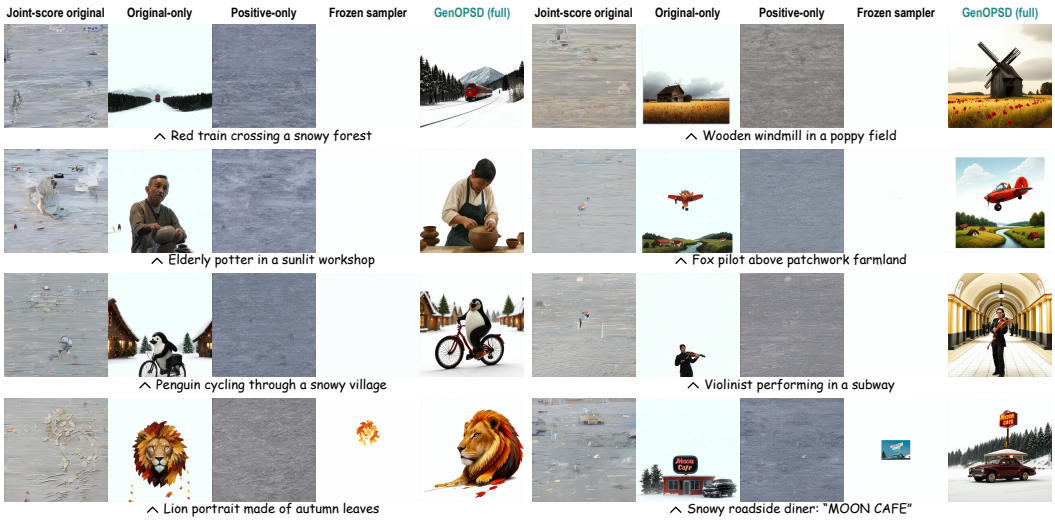


Figure 8: **Qualitative source, preference, and sampler controls.** Columns compare Joint-score original, Original-only, Positive-only, Frozen sampler, and full GENOPSD for the matched condition.

Observed behavior and limitations. In Figure 8, Joint-score original and Positive-only frequently produce repetitive textures, while Frozen sampler often yields nearly empty images. Original-only and full GENOPSD retain recognizable subjects. Full GENOPSD depicts more of the surrounding scene in several examples, including the snowy train, the cycling penguin, and the subway violinist. Figure 9 focuses on the two original-state controls and full GENOPSD to make their differences easier to inspect.



Figure 9: **Qualitative outcome-source controls.** Each group compares Joint-score original, Original-only, and full GENOPSD under matched generation settings.