REINFORCING FEW-STEP GENERATORS VIA REWARD-TILTED DISTRIBUTION MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in few-step diffusion distillation have enabled efficient image generation, yet aligning these models with human preferences remains challenging. We propose *Reward-Tilted Distribution Matching Distillation* (RTDMD), a two-stage framework that unifies distribution matching distillation with reward-guided reinforcement learning for few-step flow generators. We show that minimizing the KL divergence to a reward-tilted teacher distribution naturally decomposes into a distribution matching term and a reward maximization term. In the first stage, we introduce *Ambient-Consistent Distribution Matching Distillation* (AC-DMD), which performs subinterval-wise distribution matching and augments the fake score objective with a consistency regularizer to help the fake score model track the shifting generator distribution under limited updates. In the second stage, we jointly optimize both terms: for the reward maximization term, we derive a hybrid policy gradient that combines a GRPO-style estimator for the stochastic intermediate transitions with direct reward backpropagation through the deterministic final step, and further introduce *step-subset GRPO* (SubGRPO) to reduce variance. Experiments on SD3, SD3.5, and FLUX.2 demonstrate that RTDMD establishes new state-of-the-art results across preference, aesthetic, and compositional metrics with only 4 inference steps, outperforming previous few-step text-to-image generation methods.

1 INTRODUCTION

Diffusion (Ho et al., 2020; Song et al., 2021b) and flow-based generative models (Lipman et al., 2023; Liu et al., 2023) have achieved remarkable progress in text-to-image generation. Modern diffusion and rectified-flow systems (Esser et al., 2024; Black Forest Labs, 2024; Podell et al., 2024) can synthesize realistic and semantically aligned images, but their iterative sampling procedures typically require tens of denoising or flow-integration steps (Song et al., 2021a; Ho et al., 2020). This high sampling cost limits their deployment in latency-sensitive applications such as interactive content creation, on-device generation, and real-time visual systems.

To improve efficiency, recent works distill pretrained multi-step models into few-step generators (Salimans & Ho, 2022; Luo et al., 2023a; Sauer et al., 2024b;a; Luo et al., 2025; Chen et al., 2025a; Fan et al., 2026b). Among them, Distribution Matching Distillation (DMD) (Yin et al., 2024b;a; Liu et al., 2026a; Ge et al., 2025) trains a student to match the teacher's output distribution via a learned fake score model. Orthogonally, reinforcement learning (RL) aligns generative models with human preferences (Black et al., 2024; Fan et al., 2023; Liu et al., 2026b; Xue et al., 2025b; Xu et al., 2023; Wu et al., 2023; Clark et al., 2024; Wallace et al., 2024). Recent efforts combine distribution matching with reward optimization (Jiang et al., 2025; Luo et al., 2026; Dong et al., 2026), aiming to retain the teacher's generative prior while steering the student toward higher-reward outputs.

However, reward-guided few-step generation remains challenging for two reasons. First, in few-step generation, the intermediate latents at non-terminal timesteps are inherently noisy. The fake score model in DMD must therefore be trained on these noisy intermediates rather than clean samples. Moreover, the generator distribution shifts at every training iteration, requiring the fake score to continuously track a moving target under a limited compute budget, which makes the cold-start distillation signal unreliable. Second, reward optimization must respect the hybrid nature of the sampling dynamics: intermediate steps are stochastic due to injected noise, while the final step is

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: Visual generations produced by our RTDMD method under 4 NFE on FLUX.2 4B (Black Forest Labs, 2025) without applying classifier-free guidance (CFG) (Ho & Salimans, 2021). More visual results can be found in Sec. P.

deterministic (terminal noise level is zero). Optimizing only the stochastic steps (Liu et al., 2026b; Li et al., 2025) or only the deterministic final mapping (Xu et al., 2023; Jiang et al., 2025) are both suboptimal; a tailored estimator that accounts for the full trajectory is needed.

In this work, we propose *Reward-Tilted Distribution Matching Distillation* (RTDMD), a two-stage framework for training high-quality few-step flow generators. We show that minimizing the KL divergence to a *reward-tilted teacher distribution* (defined in Eq. (6)) naturally decomposes into a distribution matching term and a reward maximization term, providing a principled unification of distillation and RL. In the first stage, we introduce *Ambient-Consistent DMD* (AC-DMD) as a stable cold start. AC-DMD performs distribution matching on each time subinterval independently, and augments the fake score objective with a consistency regularizer (Daras et al., 2023; 2024) that couples predictions across timesteps. This helps the fake score model track the shifting generator distribution more effectively under limited updates. In the second stage, we jointly optimize both terms via a hybrid policy gradient that combines GRPO-style updates for stochastic intermediate transitions with direct reward backpropagation through the deterministic final step, and further introduce step-subset GRPO (SubGRPO) with shared noise to reduce variance.

Comprehensive experiments on SD3-M (Esser et al., 2024), SD3.5-M (Stability AI, 2024), and FLUX.2 4B (Black Forest Labs, 2025) demonstrate that RTDMD achieves state-of-the-art few-step generation quality under 4-step sampling. Notably, our distilled FLUX.2 4B surpasses the full FLUX.2 9B (50-step) across most benchmarks.

2 Preliminaries

2.1 Diffusion and Flow Models

Diffusion and flow-based generative models (Song et al., 2021a; Ho et al., 2020) define a continuous probability path $\{p_t\}_{t\in[0,1]}$ that connects the data distribution p_0 to a simple prior p_1 (typically a standard Gaussian). A common Gaussian interpolation is $q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\alpha_t \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})$, where $\boldsymbol{x}_0 \sim p_0$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, so that $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ and $p_t(\boldsymbol{x}_t) = \int q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0) p_0(\boldsymbol{x}_0)\, d\boldsymbol{x}_0$. Here α_t and σ_t specify the noise schedule.

Sampling is described by the probability-flow Ordinary Differential Equation (PF-ODE) (Lipman et al., 2023) $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{x}_t, t)$, where $\boldsymbol{v}(\boldsymbol{x}, t)$ is the *marginal velocity field* transporting the density p_t. Its relation (Ma et al., 2024) to the *score function* $s(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ is

$$\boldsymbol{v}(\boldsymbol{x}, t) = \frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{x} + \left(\frac{\dot{\alpha}_t}{\alpha_t} \sigma_t^2 - \dot{\sigma}_t \sigma_t \right) s(\boldsymbol{x}, t). \tag{1}$$

Thus, the score function indicates how the density changes locally, while the marginal velocity determines how samples move along the probability path.

In this work, we adopt flow matching with the rectified schedule (Lipman et al., 2023; Esser et al., 2024), namely $\alpha_t = 1 - t$ and $\sigma_t = t$, which gives the linear path $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$. For a fixed pair $(\boldsymbol{x}_0, \boldsymbol{\epsilon})$, the conditional velocity is simply $\boldsymbol{v}_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{\epsilon}) = \boldsymbol{\epsilon} - \boldsymbol{x}_0$, and the marginal velocity satisfies

$$\boldsymbol{v}(\boldsymbol{x}, t) = \mathbb{E}[\boldsymbol{\epsilon} - \boldsymbol{x}_0 \mid \boldsymbol{x}_t = \boldsymbol{x}] = -\frac{1}{1-t}\boldsymbol{x} - \frac{t}{1-t}s(\boldsymbol{x}, t). \tag{2}$$

Flow matching trains a neural velocity field $\boldsymbol{v}_\theta$ by regressing it to the conditional target velocity (Lipman et al., 2023):

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\big[w(t)\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - (\boldsymbol{\epsilon} - \boldsymbol{x}_0)\|_2^2\big], \quad \boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}, \tag{3}$$

where $w(t)$ is an optional weighting function. This objective is commonly referred to as the conditional flow matching (CFM) loss.

2.2 Distribution Matching Distillation

Sampling from a pretrained flow model typically requires many function evaluations, motivating distillation into a few-step generator. Let $\boldsymbol{v}_\psi$ denote the pretrained *teacher* velocity field, and let p_ψ be its induced distribution.

Distribution Matching Distillation (DMD) (Yin et al., 2024b;a) trains a few-step *student* generator G_θ with $\boldsymbol{x}_0 = G_\theta(\boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, so that its induced distribution p_θ matches p_ψ. A natural objective is the reverse Kullback–Leibler (KL) $\mathbb{D}_{\mathrm{KL}}(p_\theta \,\|\, p_\psi)$. Because this divergence can be difficult to optimize directly in data space when the two distributions have limited overlap, DMD instead compares their noised marginals in an ambient space. Specifically, for $t \sim \mathrm{Unif}[0, 1]$ and an independent $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, it defines $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}'$, which induces marginals $p_{\theta,t}$ and $p_{r,t}$ for the student and teacher, respectively. The resulting time-averaged reverse KL yields a generator gradient proportional to the difference between the student and teacher scores at $\boldsymbol{x}_t$.

Using Eq. (2) to convert between velocities and scores, DMD writes the generator update as

$$\nabla_\theta \mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_{\boldsymbol{\epsilon},t,\boldsymbol{\epsilon}'}\big[w_t \alpha_t \big(s_\phi(\boldsymbol{x}_t, t) - s_\psi(\boldsymbol{x}_t, t)\big)\nabla_\theta G_\theta(\boldsymbol{\epsilon})\big], \tag{4}$$

where $w_t > 0$ is a time-dependent weight and $\alpha_t = 1 - t$ comes from $\partial \boldsymbol{x}_t / \partial \boldsymbol{x}_0$.

Since the student score is not available in closed form, DMD introduces an auxiliary *fake* velocity field $\boldsymbol{v}_\phi$ to track the current student distribution p_θ. Via Eq. (2), $\boldsymbol{v}_\phi$ defines the fake score s_ϕ, which serves as a surrogate for the student score in Eq. (4). The fake velocity is trained with the conditional flow-matching objective

$$\mathcal{L}_f(\phi) = \mathbb{E}_{\boldsymbol{\epsilon},t,\boldsymbol{\epsilon}'}\big[\|\boldsymbol{v}_\phi(\boldsymbol{x}_t, t) - (\boldsymbol{\epsilon}' - \boldsymbol{x}_0)\|^2\big]. \tag{5}$$

At optimum, $\boldsymbol{v}_\phi$ recovers the marginal velocity field of the current student distribution, providing the score estimate required by the DMD update. DMD therefore alternates between updating the student generator G_θ and training the fake model $\boldsymbol{v}_\phi$ to track it.

3 Method

We present *Reward-Tilted Distribution Matching Distillation* (RTDMD), a principled framework for training high-quality few-step generators (an overall algorithm can be found in Sec. I). Let p_ψ denote the distribution induced by the pretrained teacher model. While DMD (Yin et al., 2024b;a) aims to replicate p_ψ, the teacher distribution itself is not necessarily aligned with human preferences, which means it can assign equal probability to both high-reward and low-reward samples. A natural remedy is to up-weight high-reward regions of p_ψ while down-weighting low-reward ones. Therefore, we define the *reward-tilted teacher distribution* as

$$\tilde{p}_\psi(\boldsymbol{x}) = \frac{p_\psi(\boldsymbol{x}) \exp(\beta r(\boldsymbol{x}))}{Z}, \tag{6}$$

where $r(\boldsymbol{x})$ is a scalar reward function, $\beta \geq 0$ controls the reward strength, and Z is the normalizing constant. We optimize the few-step generator G_θ by minimizing $\mathbb{D}_{\mathrm{KL}}(p_\theta \| \tilde{p}_\psi)$. Since $\log \tilde{p}_\psi(\boldsymbol{x}) =$

$\log p_\psi(\boldsymbol{x}) + \beta r(\boldsymbol{x}) - \log Z$, and Z is independent of θ, we have $\mathbb{D}_{\mathrm{KL}}(p_\theta \| \tilde{p}_\psi) = \mathbb{D}_{\mathrm{KL}}(p_\theta \| p_\psi) - \beta \mathbb{E}_{\hat{\boldsymbol{x}}_0 \sim p_\theta}[r(\hat{\boldsymbol{x}}_0)] + \log Z$ and

$$\nabla_\theta \mathbb{D}_{\mathrm{KL}}(p_\theta \| \tilde{p}_\psi) = \underbrace{\nabla_\theta \mathbb{D}_{\mathrm{KL}}(p_\theta \| p_\psi)}_{\text{distribution matching}} - \beta \underbrace{\nabla_\theta \mathbb{E}_{\hat{\boldsymbol{x}}_0 \sim p_\theta}[r(\hat{\boldsymbol{x}}_0)]}_{\text{reward maximization}}. \tag{7}$$

This decomposition shows that minimizing the KL to the reward-tilted distribution naturally separates into a *distribution matching* term and a *reward maximization* term. This motivates our two-stage framework: we first perform distribution matching as a cold start via *Ambient-Consistent DMD* (AC-DMD, Sec. 3.1), and then jointly optimize both terms using a *hybrid policy gradient* with *step-subset GRPO* for the reward term (Sec. 3.2).

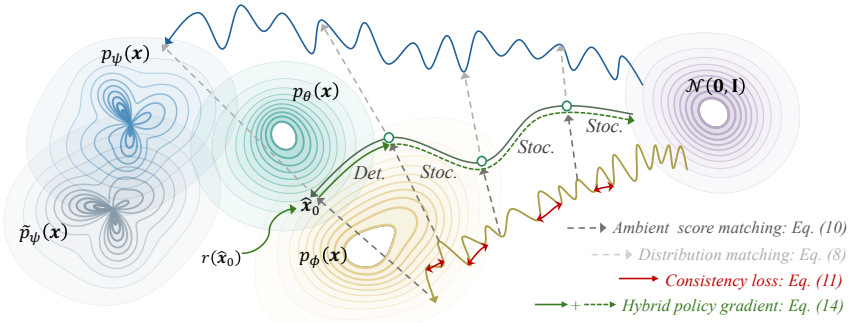


Figure 2: Overview of the proposed RTDMD. "*Det.*" means the final deterministic step and "*Stoc.*" denotes the stochastic steps (see Sec. 3.2). Blue, green, and yellow trajectories represent the denoising trajectory of the pretrained-teacher, few-step generator, and fake score model, respectively.

3.1 Ambient-Consistent Distribution Matching Distillation

Existing DMD methods adopt either the deterministic Euler ODE sampler (Fan et al., 2026b) or the consistency model (CM) sampler (Yin et al., 2024b;a) for few-step generation. To unify these choices under a single framework and facilitate the subsequent policy-gradient derivation (Sec. 3.2), we first employ coefficient-preserving sampling (CPS) (Wang & Yu, 2025), which encompasses both as special cases by a predefined hyperparameter η (see Sec. C for the full formula). Under CPS, each generation step consists of a denoising prediction followed by noise injection, and it also ensures that the noise level of the latent variable remains consistent with the predefined scheduler at every timestep.

To be more specific, we use a K-step generator with $K = 4$ and a decreasing timestep schedule $0 = t_K < \cdots < t_1 < t_0 = 1$. Starting from $\hat{\boldsymbol{x}}_{t_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, step k takes the current latent $\hat{\boldsymbol{x}}_{t_{k-1}}$ and outputs an x-prediction, $\hat{\boldsymbol{x}}^{(k)}_{\mathrm{pred}} = G_\theta(\hat{\boldsymbol{x}}_{t_{k-1}}, t_{k-1})$, $k = 1, \ldots, K$, which is the sampler output under the x-parameterization, rather than a clean sample itself. The next latent $\hat{\boldsymbol{x}}_{t_k}$ is a linear combination of the x-prediction $\hat{\boldsymbol{x}}^{(k)}_{\mathrm{pred}}$, the current latent $\hat{\boldsymbol{x}}_{t_{k-1}}$, and a freshly sampled Gaussian noise $\boldsymbol{\epsilon}_k$. Here, $\eta \in [0, 1]$ controls the sampling stochasticity: $\eta = 0$ recovers the deterministic Euler sampler, while $\eta > 0$ injects noise at each step.

Ambient distribution matching distillation. Since CPS ($\eta > 0$) injects noise at intermediate steps, the generator output $\hat{\boldsymbol{x}}_{t_k}$ at $t_k > 0$ could no longer be a clean sample but a noisy latent at noise level t_k. The standard DMD (Yin et al., 2024b;a), which assumes clean samples and performs score matching over the full interval $[0, 1]$, is therefore no longer directly applicable. We re-derive the distribution matching objective on the subinterval $[t_k, 1]$ conditioned on the noisy intermediate, and term this *Ambient Distribution Matching Distillation* (A-DMD).

Concretely, let $p^{(k)}_\theta$ denote the distribution of $\hat{\boldsymbol{x}}_{t_k}$ after k steps. To train step k, we match the teacher distribution on the subinterval $[t_k, 1]$. Under the rectified schedule (Lipman et al., 2023; Esser et al., 2024), we re-noise $\hat{\boldsymbol{x}}_{t_k}$ to any level $t \in [t_k, 1]$ via $\boldsymbol{x}^{(k)}_t = \alpha_k(t)\hat{\boldsymbol{x}}_{t_k} + \sigma_k(t)\boldsymbol{\epsilon}$, where $\alpha_k(t) = \frac{1-t}{1-t_k}$, $\sigma_k(t) = \frac{t-t_k}{1-t_k}$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Let $p^{(k)}_{\theta,t}$ denote the resulting student marginal at noise level t. We minimize the reverse KL:

$$\mathcal{L}^{(k)}_{\mathrm{gen}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[t_k, 1]}\left[\lambda_k(t)\, \mathrm{KL}(p^{(k)}_{\theta,t} \,\|\, p_{\psi,t})\right], \tag{8}$$

where $p_{\psi,t}$ is the teacher marginal and $\lambda_k(t)$ is a timestep-dependent weight.

Since the student score $\nabla_{\boldsymbol{x}} \log p^{(k)}_{\theta,t}(\boldsymbol{x})$ is intractable, following DMD (Yin et al., 2024b;a), we introduce a fake score model $s_\phi(\boldsymbol{x}, t)$ to approximate it, yielding the practical generator gradient

$$\nabla_\theta \mathcal{L}^{(k)}_{\text{gen}} \approx -\mathbb{E}_{t,\boldsymbol{\epsilon},\hat{\boldsymbol{x}}_{t_k}} \left[\lambda_k(t)\alpha_k(t) \left(s_\psi(\boldsymbol{x}^{(k)}_t, t) - s_\phi(\boldsymbol{x}^{(k)}_t, t) \right)^\top \frac{\partial \hat{\boldsymbol{x}}_{t_k}}{\partial \theta} \right]. \tag{9}$$

This form (see Sec. B for the detailed derivation) makes the training signal entirely local to the subinterval $[t_k, 1]$: the teacher score $s_\psi(\boldsymbol{x}^{(k)}_t, t)$ provides the target direction, while the fake score $s_\phi(\boldsymbol{x}^{(k)}_t, t)$ compensates for the intractable student marginal score.

To train s_ϕ, we fit it on the same interval $[t_k, 1]$ using denoising score matching (DSM):

$$\mathcal{L}^{(k)}_{\text{fake}}(\phi) = \mathbb{E}_{t\sim\mathcal{U}[t_k,1],\, \hat{\boldsymbol{x}}_{t_k},\, \boldsymbol{\epsilon}} \left[\omega_k(t) \left\| s_\phi(\boldsymbol{x}^{(k)}_t, t) - \nabla_{\boldsymbol{x}} \log q^{(k)}_t(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k}) \right\|_2^2 \right], \tag{10}$$

where $\nabla_{\boldsymbol{x}} \log q^{(k)}_t(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k}) = \left(\alpha_k(t)\hat{\boldsymbol{x}}_{t_k} - \boldsymbol{x}\right)/\sigma_k(t)^2$ is the conditional score of the Gaussian corruption kernel.

Stabilizing fake score training via consistency regularization. However, when $t_k > 0$, the fake score model s_ϕ is trained on corrupted intermediate latents $\hat{\boldsymbol{x}}_{t_k}$ rather than clean samples. Although the DSM objective in Eq. (10) is theoretically unbiased (its optimal solution is the true student marginal score $\nabla_{\boldsymbol{x}} \log p^{(k)}_{\theta,t}$ as proved in Sec. D), a practical challenge arises: the generator G_θ is updated concurrently, so the target distribution $p^{(k)}_\theta$ shifts at every training iteration. With only a limited number of fake score updates per generator step, s_ϕ must track this moving distribution under a tight sample and compute budget, making accurate estimation difficult.

To stabilize fake score training, we introduce a consistency regularizer (Daras et al., 2023; 2024). The key insight is that the optimal fake score model satisfies a self-consistency property (see Sec. E for a detailed proof): for any $t'' < t'$, its x-prediction at t' must equal the expected x-prediction after one reverse-diffusion step to t'', *i.e.*, $\hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_{t'}, t') = \mathbb{E}_{\tilde{\boldsymbol{x}}_{t''}\sim p_\phi(\boldsymbol{x}_{t''}\mid\boldsymbol{x}_{t'})}[\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}, t'')]$, where $\hat{\boldsymbol{x}}_\phi(\boldsymbol{x}, t)$ denotes the fake score model's x-prediction. This couples the fake score predictions across different timesteps, reducing the effective degrees of freedom and lowering the overall estimation variance.

Concretely, writing the fake score model in its x-prediction form, we penalize violations of this property:

$$\mathcal{L}^{(k)}_{\text{cons}}(\phi) = \mathbb{E}_{t',t'',\hat{\boldsymbol{x}}_{t_k}\sim p^{(k)}_\theta,\, \boldsymbol{x}_{t'}\sim q^{(k)}_{t'}(\cdot\mid\hat{\boldsymbol{x}}_{t_k})} \left[\left\| \hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_{t'}, t') - \mathbb{E}_{\tilde{\boldsymbol{x}}_{t''}\sim p_\phi(\boldsymbol{x}_{t''}\mid\boldsymbol{x}_{t'})}\left[\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}, t'')\right] \right\|_2^2 \right], \tag{11}$$

where $p_\phi(\boldsymbol{x}_{t''} \mid \boldsymbol{x}_{t'})$ is the fake score model's reverse transition kernel from t' to t''. In practice, we use an approximated estimator following Daras et al. (2024) (see Sec. F). We choose t' and t'' to be close so that the consistency term remains local and can be estimated efficiently with a single transition step. The final fake-score objective is

$$\mathcal{L}^{(k)}_{\text{fake-total}}(\phi) = \mathcal{L}^{(k)}_{\text{fake}}(\phi) + \gamma\, \mathcal{L}^{(k)}_{\text{cons}}(\phi). \tag{12}$$

Intuitively, Eq. (10) provides pointwise score supervision at each noise level, while the consistency term couples nearby timesteps by requiring them to predict the same underlying clean sample, thereby reducing the variance of ambient fake-score training.

Overall, we refer to our method as *Ambient-Consistent Distribution Matching Distillation* (AC-DMD), reflecting that the fake score model is trained on noisy intermediate latents (the "ambient" setting) and regularized by a consistency loss to improve estimation quality.

3.2 Reinforcing the Few-step Generator

After the cold start with AC-DMD, we proceed to the second stage: jointly optimizing both terms in Eq. (7). The distribution matching term is handled by AC-DMD as before; we now focus on deriving efficient gradient estimators for the reward maximization term.

Few-step generator as a policy. The few-step generator induces a K-step policy over the latent trajectory $\hat{\boldsymbol{x}}_{t_0} \to \hat{\boldsymbol{x}}_{t_1} \to \cdots \to \hat{\boldsymbol{x}}_{t_K}$. At each step, the CPS update combines the generator's x-prediction with the current latent and injected Gaussian noise [1]. As a result, for the first $K-1$ steps, the transition defines a Gaussian policy

$$\pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}) = \mathcal{N}\big(\hat{\boldsymbol{x}}_{t_k};\, \boldsymbol{\mu}_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_{k-1}}),\, \sigma_k^2 \mathbf{I}\big), \qquad k = 1, \ldots, K-1, \tag{13}$$

where $\boldsymbol{\mu}_\theta^{(k)}$ is determined by the CPS update (Sec. C) and $\sigma_k = t_k \sin(\eta\pi/2)$. The final step is deterministic: since $t_K = 0$, the noise term vanishes and $\hat{\boldsymbol{x}}_0 = G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1})$. Therefore, the few-step generative process is a hybrid policy consisting of $K-1$ stochastic Gaussian steps followed by one deterministic step.

Hybrid policy gradient. As a result, the reward gradient (*i.e.*, $\nabla_\theta \mathbb{E}_{\hat{\boldsymbol{x}}_0 \sim p_\theta}[r(\hat{\boldsymbol{x}}_0)]$ in Eq. (7)) naturally decomposes into a contribution from the stochastic intermediate transitions and a contribution from the deterministic final mapping.

Specifically, let $\tau = (\hat{\boldsymbol{x}}_{t_0}, \hat{\boldsymbol{x}}_{t_1}, \ldots, \hat{\boldsymbol{x}}_{t_{K-1}}, \hat{\boldsymbol{x}}_0)$ denote a generated trajectory. Then

$$\begin{aligned} \nabla_\theta \mathbb{E}_{\hat{\boldsymbol{x}}_0 \sim p_\theta}[r(\hat{\boldsymbol{x}}_0)] = &\sum_{k=1}^{K-1} \mathbb{E}_{\tau \sim p_\theta}\Big[r(\hat{\boldsymbol{x}}_0)\, \nabla_\theta \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}})\Big] \\ &+ \mathbb{E}_{\tau \sim p_\theta}\Big[\big(\nabla_{\hat{\boldsymbol{x}}_0} r(\hat{\boldsymbol{x}}_0)\big)^\top \partial_\theta G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1})\Big]. \end{aligned} \tag{14}$$

The first term is a REINFORCE-style estimator and accounts for how the parameters affect the distribution of the stochastic intermediate states, while the second term differentiates the deterministic final denoising step (a formal derivation is provided in Sec. G; see Proposition G.1). Since the reward is typically differentiable, we estimate the second term by directly backpropagating through G_θ:

$$\nabla_\theta \mathcal{L}_{\text{det}} = \frac{1}{N} \sum_{i=1}^{N} \big(\nabla_{\hat{\boldsymbol{x}}_0} r(\hat{\boldsymbol{x}}_0^{(i)})\big)^\top \partial_\theta G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}^{(i)}, t_{K-1}). \tag{15}$$

For the first term, directly using the REINFORCE-style term leads to high variance. Following GRPO (Shao et al., 2024), we reduce variance of it by sampling a group of N trajectories $\{\tau_i\}_{i=1}^N$ per prompt and replacing the raw reward r with a group-normalized advantage $A_i = (r_i - \bar{r})/\operatorname{std}(\{r_j\}_{j=1}^N$, where $r_i = r(\hat{\boldsymbol{x}}_0^{(i)})$ is the reward of the i-th trajectory and $\bar{r} = \frac{1}{N}\sum_{j=1}^N r_j$.

Step-subset GRPO with shared noise. However, naive GRPO (see Sec. H for more details) uses independent noise at every step, so reward differences across trajectories conflate contributions from all steps. Inspired by MixGRPO (Li et al., 2025), we propose *step-subset GRPO with shared noise* (SubGRPO) to further reduce variance by isolating the effect of selected steps. For each prompt, we uniformly sample a subset of stochastic steps $\mathcal{S} \subset \{1, \ldots, K-1\}$ with $|\mathcal{S}| = M < K-1$. The full K-step trajectory is still rolled out, but only the steps in $\mathcal{S}$ use independent noise across trajectories; the remaining steps share noise within the group:

$$\tilde{\boldsymbol{\epsilon}}_k^{(i)} = \begin{cases} \boldsymbol{\epsilon}_k^{(i)}, & k \in \mathcal{S}, \\ \boldsymbol{\epsilon}_k^{\text{grp}}, & k \notin \mathcal{S}, \end{cases} \qquad \boldsymbol{\epsilon}_k^{(i)},\, \boldsymbol{\epsilon}_k^{\text{grp}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{16}$$

where $\{\boldsymbol{\epsilon}_k^{(i)}\}_{i=1}^N$ are independent across trajectories, while $\boldsymbol{\epsilon}_k^{\text{grp}}$ is shared by all trajectories in the same group at step k. Only the selected steps $k \in \mathcal{S}$ contribute the gradients, yielding

$$\nabla_\theta \mathcal{L}_{\text{stoc}}^{\text{SubGRPO}} = \frac{K-1}{M} \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{k \in \mathcal{S}} A_i\, \nabla_\theta \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k}^{(i)} \mid \hat{\boldsymbol{x}}_{t_{k-1}}^{(i)}). \tag{17}$$

Under the same gradient sample budget, SubGRPO can be viewed as a Rao–Blackwellized variant of the corresponding independent-noise estimator under mild assumptions (Casella & Robert, 1996). Therefore, its gradient estimator typically has a smaller variance.

Total objective. Combining Eqs. (15), (17), and (9), the generator in the second stage is updated by descending along:

$$\nabla_\theta \mathcal{L}_{\text{total}} = \nabla_\theta \mathcal{L}_{\text{AC-DMD}} - \beta\big(\nabla_\theta \mathcal{L}_{\text{stoc}}^{\text{SubGRPO}} + \nabla_\theta \mathcal{L}_{\text{det}}\big), \tag{18}$$

[1] We set $\eta > 0$ throughout this work, as it naturally introduces stochasticity into the sampling trajectory, which is essential for exploration in reinforcement learning.

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

Models. Our experiments are conducted on two open-source state-of-the-art (SOTA) text-to-image diffusion models: Stable Diffusion 3-Medium (`SD3-M`) (Esser et al., 2024), Stable Diffusion 3.5-Medium (`SD3.5-M`) (Stability AI, 2024) and FLUX.2 4B (Black Forest Labs, 2025). We use 512^2 as the default resolution unless otherwise specified.

Rewards. For `SD3-M`, following prior work (Jiang et al., 2025; Luo et al., 2025), we train with HPSv2 (Wu et al., 2023) and CLIPScore (Hessel et al., 2021) rewards on prompts from `t2i-2M` (Data, 2024), and evaluate on prompts sampled from ShareGPT-4o-Image (Chen et al., 2025b), reporting CLIPScore, Aesthetic Score (Schuhmann et al., 2022), PickScore (Kirstain et al., 2023), and HPSv2. Besides, we further validate on the non-differentiable GenEval reward (Ghosh et al., 2023) for `SD3.5-M`. For FLUX.2 4B, we train with HPSv2, CLIPScore, PickScore, and GenEval rewards, and additionally evaluate on OCR Score, Aesthetic Score, GenEval2 (Kamath et al., 2025), ImageReward (Xu et al., 2023), and HPSv3 (Ma et al., 2025b) for a thorough assessment.

Training. We finetune the generator initialized from its corresponding pre-trained teacher without CFG (Ho & Salimans, 2021) using LoRA (Hu et al., 2022) ($\alpha = 32, r = 64$) and adopt CPS (Wang & Yu, 2025) with $\eta = 0.9$ (see Sec. K for more discussion). For the cold start stage, we adopt 1.5×10^3 training iterations for the generator and $\gamma = 0.01$. In the second stage, we use 1×10^3 iterations, and each consists of 48 groups with a group size of 24 for `SD3-M` and `SD3.5-M`, and 64 groups with the same group size for FLUX.2 4B. All the experiments are conducted on 8 or 16 NVIDIA H20 GPUs. More details can be found in Sec. J.

Baselines. We compare our method against SOTA few-step RL approaches, including GDMD (Dong et al., 2026), DMDR (Jiang et al., 2025), R_{dm} (Fan et al., 2026a), TDM-R1 (Luo et al., 2026), and Hyper-SD (Ren et al., 2024). To cover a wider range of baselines, we further include foundational multi-step base models (Black Forest Labs, 2025; Z-Image Team et al., 2025; Stability AI, 2024), RL-only approaches (Xu et al., 2023), and few-step distillation methods (Luo et al., 2023a; Yin et al., 2024a; Chadebec et al., 2025; Luo et al., 2025) as baselines. We reproduce the closed-source R_{dm} and evaluate the open-source baselines with their official checkpoints, while the remaining results are directly taken from GDMD (Dong et al., 2026) and TDM-R1 (Luo et al., 2026).

Table 1: Quantitative comparison with SOTA approaches on `SD3-M` with 1024^2 resolution. We highlight the best and second-best scores in bold and underlined formats, respectively. Visual results can be found in Fig. 9

Method	NFE	CLIPScore		Aesthetic		PickScore		HPSv2		ImageReward	
		Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
`SD3-M` (w/o CFG) (Esser et al., 2024)	50	0.2535	−0.0401	5.2840	−0.2871	20.5660	−1.7576	0.2010	−0.0800	−0.3762	−1.4521
`SD3-M` (w/ CFG) (Esser et al., 2024)	100	0.2936	—	5.5711	—	22.3236	—	0.2810	—	1.0759	—
Few-Step Distillation											
DMD (Yin et al., 2024b)	4	0.2861	−0.0075	5.5598	−0.0113	21.6216	−0.7020	0.2891	+0.0081	0.9704	−0.1055
DMD2 (Yin et al., 2024a)	4	0.2914	−0.0022	5.7704	+0.1993	22.1442	−0.1794	0.2951	+0.0141	1.1689	+0.0930
Flash Diffusion (Chadebec et al., 2025)	4	0.2864	−0.0072	5.5236	−0.0475	22.0558	−0.2678	0.2636	−0.0174	0.8752	−0.2007
TDM (Luo et al., 2025)	4	0.2848	−0.0088	5.7070	+0.1359	22.1806	−0.1430	0.2940	+0.0130	1.0932	+0.0173
Few-Step RL											
Hyper-SD (w/ CFG) (Ren et al., 2024)	8	0.2827	−0.0109	5.5715	+0.0004	21.2576	−1.0660	0.2608	−0.0202	0.6562	−0.4197
DMDR (Jiang et al., 2025)	4	0.2901	−0.0035	5.5123	−0.0588	21.6528	−0.6708	0.2931	+0.0121	1.0120	−0.0639
GDMD (Dong et al., 2026)	4	0.2930	−0.0006	5.8728	+0.3017	22.4614	+0.1378	<u>0.3076</u>	<u>+0.0266</u>	1.2702	+0.1943
R_{dm} (Fan et al., 2026a)	4	<u>0.2936</u>	<u>+0.0000</u>	<u>5.8769</u>	<u>+0.3058</u>	<u>22.5783</u>	<u>+0.2547</u>	0.2957	+0.0147	<u>1.2897</u>	<u>+0.2138</u>
RTDMD (Ours)	4	**0.3161**	+0.0225	**5.9642**	+0.3931	**22.8593**	+0.5357	**0.3211**	+0.0401	**1.3024**	+0.2265

4.2 PERFORMANCE ANALYSIS

Comparison with baselines. We first evaluate our RTDMD on `SD3-M` with 4-step generation and report results in Tab. 1. RTDMD achieves the best performance across all five evaluation metrics, establishing a new state of the art for few-step generation. Specifically, we attain a CLIPScore (Hessel et al., 2021) of 0.3161, a PickScore (Kirstain et al., 2023) of 22.86, and an HPSv2 (Wu et al., 2023) of 0.3211, outperforming the strongest prior methods R_{dm} (Fan et al., 2026a) and GDMD (Dong et al., 2026) by a large margin. Notably, these gains extend beyond training-time rewards to unseen evaluation metrics: our model reaches an Aesthetic Score of 5.9642 and an ImageReward of 1.3024, surpassing the 100-NFE teacher with CFG (Ho & Salimans, 2021) by substantial margins (+0.39 and +0.23, respectively) while using $25\times$ fewer NFE. We further compare with representative baselines

on `SD3.5-M` in Sec. N.1, and validate our framework with the non-differentiable GenEval (Ghosh et al., 2023) reward (Tab. 5 in Sec. L), where our approach generalizes effectively.

Table 2: Performance on larger-scale diffusion models. We evaluate HPSv3, OCR, and GenEval following Liu et al. (2026b) and Xue et al. (2025c), and use the official setting for GenEval2. The remaining metrics are evaluated on DrawBench (Saharia et al., 2022) following Zheng et al. (2026).

Model	ImageReward	CLIPScore	Aesthetic	PickScore	HPSv2	HPSv3	GenEval	GenEval2	OCR
Diffusion Models									
FLUX.2 4B (Black Forest Labs, 2025)	0.8538	0.2834	5.3333	22.3938	0.2771	11.7025	0.7631	0.2207	0.6133
FLUX.2 9B (Black Forest Labs, 2025)	1.0021	0.2962	5.2030	22.6382	0.2800	11.6883	0.7568	0.3557	0.7432
Z-Image 6B (Z-Image Team et al., 2025)	0.7841	0.2841	5.2488	22.2118	0.2714	10.0857	0.6563	0.3012	0.7373
Few-Step Diffusion Models (4 NFE)									
Z-Image-Turbo 6B (Z-Image Team et al., 2025)	0.9696	0.2764	5.2894	22.7994	0.2954	12.9136	0.7562	0.3530	0.7539
FLUX.2 4B (Black Forest Labs, 2025)	1.0506	0.2864	5.2658	22.7370	0.2890	12.9295	0.7722	0.2403	0.6375
FLUX.2 9B (Black Forest Labs, 2025)	1.1998	0.2919	5.3730	23.0178	0.2991	13.2955	0.7814	0.3570	0.7566
Z-Image 6B w/ TDM-R1 (Luo et al., 2026)	1.1543	0.2836	5.2450	22.8202	0.3064	13.4349	0.7737	**0.4073**	**0.7665**
FLUX.2 4B w/ RTDMD (Ours)	**1.3712**	**0.3219**	**5.7746**	**23.9642**	**0.3516**	**15.5772**	**0.9046**	0.2755	0.6858



(a) FLUX.2 4B w/ Ours **(b)** FLUX.2 9B **(c)** FLUX.2 4B **(d)** Z-Image 6B w/ TDM-R1 **(e)** Z-Image-Turbo 6B

Figure 3: Qualitative comparison for few-step diffusion models (4 NFE). Using identical noise inputs, our method outperforms others in both quality and prompt alignment, showing strong performance.

Scaling to more advanced models. In Tab. 2, we further apply our RTDMD to FLUX.2 4B (Black Forest Labs, 2025), a SOTA transformer-based flow model. Our approach sets new best results in seven out of 9 metrics with substantial absolute gains over the 50-step FLUX.2 4B baseline. Notably, our 4B model surpasses the considerably larger FLUX.2 9B (50-step) on the majority of metrics, demonstrating that RL-guided distillation can effectively close the quality gap introduced by model scale reduction. While Z-Image 6B (Z-Image Team et al., 2025) with TDM-R1 (Luo et al., 2026) achieves higher absolute scores on GenEval2 (Kamath et al., 2025) and OCR (Liu et al., 2026b), this advantage stems primarily from the Z-Image base model itself being inherently stronger on these two benchmarks; in contrast, the relative improvement brought by our method over its own baseline is more pronounced on OCR (+0.0483 *vs.* +0.0126) and comparable on GenEval2. Qualitative comparisons in Fig. 3 and Fig. 10 further corroborate these findings.

4.3 Ablation Studies

Here, we perform ablation using `SD3.5-M` (Stability AI, 2024) with CLIPScore (Hessel et al., 2021), PickScore (Kirstain et al., 2023), and HPSv2 (Wu et al., 2023) as training rewards on DrawBench (Saharia et al., 2022) following Zheng et al. (2026). Other settings are the same as Sec. 4.1. Additional ablations and visual results are presented in Sec. M and Sec. O, respectively.

Effect of AC-DMD. In A-DMD, the fake score model is trained on noisy intermediate latents rather than clean samples, making denoising score matching less accurate under limited updates. As shown

Table 3: Ablation results for different distribution matching strategies in our two-stage training. All methods adopt the CPS scheduler (Wang & Yu, 2025) with $\eta = 0.9$. Blue row denotes the default setting of our RTDMD. DMD2 (Yin et al., 2024a) and more results are shown in Sec. M.1.

Method	CLIPScore		PickScore		HPSv2	
	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
A-DMD	0.2753	—	22.0116	—	0.3188	—
AC-DMD ($\gamma=0.1$)	**0.2846**	+0.0093	23.2105	+1.1989	0.3362	+0.0174
AC-DMD ($\gamma=0.01$)	0.2843	+0.0090	**23.5690**	+1.5574	**0.3456**	+0.0268
AC-DMD ($\gamma=0.005$)	0.2837	+0.0084	22.8546	+0.8430	0.3324	+0.0136
AC-DMD ($\gamma=0.001$)	0.2831	+0.0078	22.2378	+0.2262	0.3294	+0.0106

Table 4: Ablation results for strategies applied after cold start. "w/ $\mathcal{L}_{\text{det}}$" means we adopt Eq. (15) to optimize the generator. "w/ GRPO" signifies that we use naive GRPO to optimize the first $K-1$ stochastic steps. Blue row denotes the default setting of our RTDMD.

Method	CLIPScore		PickScore		HPSv2	
	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
Cold Start	0.2822	+0.0063	22.5056	−0.8008	0.2942	−0.0326
w/ GRPO	0.2759	—	23.3064	—	0.3268	—
w/ SubGRPO ($M=1$)	0.2798	+0.0039	23.3272	+0.0208	0.3332	+0.0064
w/$\mathcal{L}_{\text{det}}$	0.2802	+0.0043	23.4156	+0.1092	0.3318	+0.0050
w/ SubGRPO ($M=2$)	0.2839	+0.0080	23.4520	+0.1456	**0.3459**	+0.0191
w/$\mathcal{L}_{\text{det}}$	**0.2843**	+0.0084	**23.5690**	+0.2626	0.3456	+0.0188

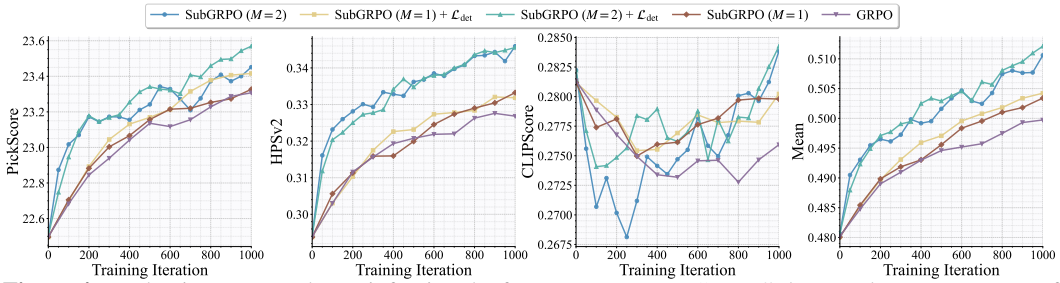


Figure 4: Evaluation curves when reinforcing the few-step generator. "Mean" denotes the average score of normalized PickScore, HPSv2, and CLIPScore.

in Tab. 3, the consistency loss $\mathcal{L}_{\text{cons}}^{(k)}$ improves fake score estimation by enforcing coherent predictions across neighboring timesteps. Notably, the improvement is consistent across all tested values of γ, confirming the robustness of the regularizer. The best result at $\gamma = 0.01$ yields an additional +1.58 PickScore and +0.027 HPSv2 over A-DMD.

Effect of SubGRPO. As shown in Tab. 4 and Fig. 4, replacing naive GRPO with SubGRPO consistently improves all metrics. Naive GRPO samples independent noise at every stochastic step, so reward differences across trajectories conflate contributions from all steps, leading to a noisy gradient signal. SubGRPO instead shares the injected noise at non-selected steps ($k \notin \mathcal{S}$) across the group (Eq. (16)), attributing reward variation primarily to the selected subset $\mathcal{S}$. This shared-noise design reduces gradient variance by isolating the effect of selected steps. Increasing the subset size from $M = 1$ to $M = 2$ brings further gains, with PickScore improving from 23.33 to 23.45 and HPSv2 from 0.3332 to 0.3459. This reflects a trade-off between per-step variance reduction (fewer selected steps yield tighter control) and optimization coverage (more selected steps provide gradient signal to a larger portion of the trajectory); in our setting, $M = 2$ strikes a favorable balance.

Effect of hybrid policy gradient. Existing GRPO-based methods for diffusion models (Liu et al., 2026b; Xue et al., 2025c) only optimize the stochastic denoising steps via the GRPO-estimator, neglecting the final deterministic transition entirely. However, our hybrid policy gradient (Eq. (14)) combines SubGRPO for stochastic intermediate steps with a pathwise gradient through this deterministic final step. Tab. 4 shows that adding this term consistently improves performance: under $M = 2$, PickScore increases from 23.45 to 23.57 and CLIPScore from 0.2839 to 0.2843, confirming that the deterministic final mapping carries a meaningful optimization signal.

5 Conclusions and Limitations

We present RTDMD, a two-stage framework for few-step image generation that unifies distribution matching distillation with reward-guided reinforcement learning. The first stage, AC-DMD employs ambient distribution matching and augments fake score training with a consistency regularizer to stabilize estimation. The second stage minimizes the KL divergence to a reward-tilted teacher distribution, which naturally decomposes into a distribution matching term and a reward maximization term. For the latter, we derive a hybrid policy gradient that combines SubGRPO for stochastic intermediate steps with direct reward backpropagation through the deterministic final step. Comprehensive experiments demonstrate the SOTA performance of our method across baselines. In terms of limitations, this work focuses on text-to-image generation; extending the framework to video synthesis and image editing, where reward design and temporal consistency pose additional challenges, is left for future work.

AI USE STATEMENT

We acknowledge the use of large language models (LLMs), such as OpenAI's `GPT-5.6`, as a writing-assistance tool in this work. Their role was strictly limited to proofreading and rephrasing sentences to enhance linguistic quality, without any contribution to the research ideation or experimental results.

REFERENCES

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 24

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD. 1, 16

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 24

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025. 2, 7, 8, 26, 31

George Casella and Christian P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83 (1):81–94, 1996. 6

Clement Chadebec, Onur Tasar, Eyal Benaroche, and Benjamin Aubin. Flash diffusion: Accelerating any conditional diffusion model for few steps image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 15686–15695, 2025. 7, 16

Guanjie Chen, Shirui Huang, Kai Liu, Jian-Xiang Zhu, Xiaoye Qu, Peng Chen, Yu Cheng, and Yifu Sun. Flash-dmd: Towards high-fidelity few-step image generation with efficient distillation and joint reinforcement learning. *ArXiv*, abs/2511.20549, 2025a. 1, 16

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. NFT: Bridging supervised learning and reinforcement learning in math reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ujBrsQm6Zu. 17

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025b. 7

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling, 2025c. URL https://arxiv.org/abs/2501.17811. 24

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1vmSEVL19f. 1, 16

Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36:42038–42063, 2023. 2, 5

Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=PlVjIGaFdH. 2, 5, 20

Hugging Face Open Data. text-to-image-2m. https://huggingface.co/datasets/jackyhate/text-to-image-2M, 2024. 7

Linwei Dong, Ruoyu Guo, Ge Bai, Zehuan Yuan, Yawei Luo, and Changqing Zou. Guiding distribution matching distillation with gradient-based reinforcement learning, 2026. URL https://arxiv.org/abs/2604.19009. 1, 7, 17, 27

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 1, 2, 3, 4, 7, 16, 24, 29

Jiajun Fan, Shuaike Shen, Chaoran Cheng, Yuxin Chen, Chumeng Liang, and Ge Liu. Online reward-weighted fine-tuning of flow matching with wasserstein regularization. In *The Thirteenth International Conference on Learning Representations*, 2025. 17

Linqian Fan, Peiqin Sun, Tiancheng Wen, Shun Lu, and Chengru Song. r_{dm}: Re-conceptualizing distribution matching as a reward for diffusion distillation, 2026a. URL https://arxiv.org/abs/2603.28460. 7, 17, 27

Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased dmd: Few-step distribution matching distillation via score matching within subintervals, 2026b. URL https://arxiv.org/abs/2510.27684. 1, 4, 16

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023. 1, 16

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Seneflow: Scaling distribution matching for flow-based text-to-image distillation. *arXiv preprint arXiv:2506.00523*, 2025. 1

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. 7, 8, 23

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. 7, 8

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 27

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI. 2, 7, 16, 23, 31

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 16

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 7, 23

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 24

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088. 16

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Zhen Li, Bo Zhang, et al. Distribution matching distillation meets reinforcement learning. *arXiv preprint arXiv:2511.13649*, 2025. 1, 2, 7, 17, 27

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation, 2025. URL https://arxiv.org/abs/2512.16853. 7, 8

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *ICLR*. OpenReview.net, 2024. 16

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. 7, 8

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023. 17

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025. 2, 6

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 16

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014. 27

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t. 1, 2, 3, 4, 16

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Steven HOI, and Hongsheng Li. Decoupled DMD: CFG augmentation as the spear, distribution matching as the shield. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=jBztvOiCKE. 1, 16

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026b. URL https://openreview.net/forum?id=oCBKGw5HNf. 1, 2, 8, 9, 17, 23, 24

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. 1

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. 23

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023a. 1, 7, 16

Weijian Luo. Diff-instruct++: Training one-step text-to-image generator model to align with human preferences, 2025. URL https://arxiv.org/abs/2410.18881. 17

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *NeurIPS*, 2023b. 16

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17719–17728, 2025. 1, 7, 16, 24, 27

Yihong Luo, Tianyang Hu, Weijian Luo, and Jing Tang. Tdm-r1: Reinforcing few-step diffusion models with non-differentiable reward, 2026. URL https://arxiv.org/abs/2603.07700. 1, 7, 8, 17, 23, 24

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024. 2

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7739–7751, 2025a. 24

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15086–15095, 2025b. 7

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eoEmoKoQpJ. 17

Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning timestep-distilled diffusion model using pairwise sample optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fXnE4gB64o. 17

OpenAI. Dalle-2, 2023. URL https://openai.com/dall-e-2. 24

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf. 1, 24

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. 27

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9. 17

Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, XING WANG, and Xuefeng Xiao. Hyper-SD: Trajectory segmented consistency model for efficient image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=O5XbOoi0x3. 7, 16, 17

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 16, 24

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=08Yk-n5l2Al. 8, 24

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI. 1, 16

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024a. 1, 16

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024b. 1, 16

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 7

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 17

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 6, 17, 22, 28

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 27

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP. 1, 2

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS. 1

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023. 16, 18

Stability AI. Sd3.5. https://github.com/Stability-AI/sd3.5, 2024. 2, 7, 8, 24, 27

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024. 1, 17

Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025. 4, 7, 9, 18

Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in neural information processing systems*, 37:83951–84009, 2024a. 16

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024b. URL https://arxiv.org/abs/2409.18869. 24

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/forum?id=HZf7UbpWHuA. 16

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. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023b. 16

Tianhe Wu, Ruibin Li, Lei Zhang, and Kede Ma. Diversity-preserved distribution matching distillation for fast visual synthesis. *arXiv preprint arXiv:2602.03139*, 2026. 27

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. 1, 7, 8, 24

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng YU, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=27hOkXzy9e. 24

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=o6Ynz6OIQ6. 24

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. 1, 2, 7, 16

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a. 17

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b. 1

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025c. 8, 9, 17

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a. 1, 3, 4, 5, 7, 9, 16, 18, 24, 25, 27

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b. 1, 3, 4, 5, 7, 16, 18

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang, Ruikai Zhou, and Shilin Zhou. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025. 7, 8

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VJZ477R89F. 8, 17

Appendix

CONTENTS

A Related Work **16**

B Derivation of the Step-k Training Objective **17**

C Coefficient-Preserving Sampling Formula **18**

D Optimality of Ambient Denoising Score Matching **19**

E Self-Consistency of the Optimal Fake Score **19**

F Practical Consistency Loss Estimator **20**

G Hybrid Policy Gradient **20**
G.1 Derivation 20
G.2 Discussion 21

H Background on GRPO and Its Application to Few-step Generators **22**

I Detailed Algorithm of RTDMD **22**

J More Implementation Details **22**

K Discussion for Sampling Schedule **23**

L GenEval Results **23**

M Additional Ablation Studies **24**
M.1 Disentangling Ambient Matching and Consistency Regularization 24
M.2 Fake-Score Tracking under a Moving Generator Distribution 25
M.3 Comparison of Training Recipes 25
M.4 SubGRPO with More Steps and Different Step Selections 26
M.5 Robustness to the Number of Groups per Iteration 26

N Additional Comparisons and Analyses **27**
N.1 Comparison with Baselines on SD3.5-M 27
N.2 Sample Diversity 27

O Qualitative comparison for Ablation **28**

P More Qualitative Results **29**

A RELATED WORK

Few-step distillation. To alleviate the iterative sampling cost of diffusion and flow-based models (Ho et al., 2020; Rombach et al., 2022; Esser et al., 2024; Lipman et al., 2023), a growing line of work distills multi-step teachers into few-step students, broadly falling into *trajectory-based* and *distribution-based* approaches. The former, including progressive distillation (Salimans & Ho, 2022; Ren et al., 2024; Lin et al., 2024; Chadebec et al., 2025) and consistency models (CM) (Song et al., 2023; Luo et al., 2023a; Kim et al., 2024; Wang et al., 2024a), replicate or enforce self-consistency along the teacher's trajectory, while the latter align output distributions via GAN-based (Wang et al., 2023a; Sauer et al., 2024b;a) or score/VSD variants (Wang et al., 2023b; Luo et al., 2023b). Among these, Distribution Matching Distillation (DMD) (Yin et al., 2024b;a) has become a foundation work for high-fidelity few-step generation, with follow-ups refining it from different angles: Flash-DMD (Chen et al., 2025a) uses a timestep-aware pixel-GAN; TDM (Luo et al., 2025) fuses trajectory and distribution matching without GAN; Decoupled DMD (Liu et al., 2026a) recasts classifier-free guidance (CFG) (Ho & Salimans, 2021) as the generative driver and DMD as a regularizer; and PhasedDMD (Fan et al., 2026b) combines phase-wise distillation with Mixture-of-Experts (MoE) (Jiang et al., 2024) to ease learning.

Reinforcement learning for diffusion models. Reinforcement learning (RL) has emerged as a powerful tool for aligning diffusion and flow-based models with human preferences. One line directly backpropagates reward gradients, as in ReFL (Xu et al., 2023) on one-step predictions and DRaFT (Clark et al., 2024) on multi-step samples via truncated backpropagation. Another line formulates denoising as a multi-step decision process: DDPO (Black et al., 2024) and DPOK (Fan et al.,

2023) derive tractable Gaussian likelihoods via an Euler–Maruyama discretization, enabling PPO-style (Schulman et al., 2017) optimization. Flow-GRPO (Liu et al., 2026b) and DanceGRPO (Xue et al., 2025c) further combine this formulation with GRPO (Shao et al., 2024), thereby simplifying PPO with a group-mean baseline. A third line optimizes the forward process: Lee et al. (2023) and Fan et al. (2025) use reward-weighted denoising losses, Diffusion-DPO (Wallace et al., 2024) adapts DPO (Rafailov et al., 2023) without rollouts, and FMPG (McAllister et al., 2026) and AWM (Xue et al., 2025a) use the ELBO as a likelihood proxy, while DiffusionNFT (Zheng et al., 2026) reinterprets GRPO as an NFT-style (Chen et al., 2026) forward-process variant.

Reinforcement learning for few-step generative models. Beyond applying RL to multi-step teachers, a growing body of work integrates RL with few-step distillation to combine efficiency with preference alignment. Early efforts treat RL as an independent post-training stage on top of distilled models: Hyper-SD (Ren et al., 2024) augments trajectory-segmented consistency distillation with human feedback learning, while PSO (Miao et al., 2025) fine-tunes timestep-distilled models with pairwise preference data to avoid costly re-distillation. Diff-Instruct++ (Luo, 2025) further reveals a theoretical link between CFG-based distillation and RL, formulating one-step generator alignment as KL-regularized reward maximization. More recent work (Luo et al., 2026; Jiang et al., 2025) moves toward joint optimization: DMDR (Jiang et al., 2025) unifies DMD and RL so that the two mutually regularize each other, and R_{dm} (Fan et al., 2026a) re-conceptualizes distribution matching itself as a reward, providing a principled bridge between DMD and GRPO-style RL. Besides these two works, GDMD (Dong et al., 2026) first combines NFT-style RL (Zheng et al., 2026) with DMD.

B Derivation of the Step-k Training Objective

Learning the generator after k sampling steps. Let $p_\theta^{(k)}$ denote the distribution of the generated state $\hat{\boldsymbol{x}}_{t_k}$ after the first k sampling steps. For training step k, we define the student marginal at time $t \in [t_k, 1]$ by diffusing $\hat{\boldsymbol{x}}_{t_k}$ forward within the remaining interval.

Under the rectified schedule, the diffusion path is

$$\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

By the Markov property of the Gaussian path, conditioning on an intermediate state $\boldsymbol{x}_{t_k}$ yields

$$\boldsymbol{x}_t = \frac{1-t}{1-t_k}\boldsymbol{x}_{t_k} + \frac{t-t_k}{1-t_k}\boldsymbol{\epsilon}, \qquad t \in [t_k, 1].$$

Therefore, for generated samples $\hat{\boldsymbol{x}}_{t_k} \sim p_\theta^{(k)}$, we define

$$\boldsymbol{x}_t^{(k)} = \alpha_k(t)\hat{\boldsymbol{x}}_{t_k} + \sigma_k(t)\boldsymbol{\epsilon}, \qquad \alpha_k(t) = \frac{1-t}{1-t_k}, \qquad \sigma_k(t) = \frac{t-t_k}{1-t_k},$$

with conditional density

$$q_k(\boldsymbol{x}_t \mid \hat{\boldsymbol{x}}_{t_k}, t) = \mathcal{N}\big(\boldsymbol{x}_t;\, \alpha_k(t)\hat{\boldsymbol{x}}_{t_k},\, \sigma_k(t)^2\mathbf{I}\big)\,.$$

The induced student marginal is

$$p_{\theta,t}^{(k)}(\boldsymbol{x}) = \int q_k(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k}, t)\, p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})\, d\hat{\boldsymbol{x}}_{t_k}.$$

Following the reverse-KL formulation used in distribution matching distillation, we optimize

$$\mathcal{L}_{\text{gen}}^{(k)}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[t_k,1]}\left[\lambda_k(t)\,\mathrm{KL}\Big(p_{\theta,t}^{(k)} \,\|\, p_{\psi,t}\Big)\right].$$

Using the pathwise derivative of the reverse KL gives

$$\nabla_\theta \mathcal{L}_{\text{gen}}^{(k)} = \mathbb{E}_{t,\boldsymbol{\epsilon},\hat{\boldsymbol{x}}_{t_k}}\left[\lambda_k(t)\,\Big(\nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x}_t^{(k)}) - s_\psi(\boldsymbol{x}_t^{(k)}, t)\Big)^\top \frac{\partial \boldsymbol{x}_t^{(k)}}{\partial\theta}\right].$$

Since

$$\boldsymbol{x}_t^{(k)} = \alpha_k(t)\hat{\boldsymbol{x}}_{t_k} + \sigma_k(t)\boldsymbol{\epsilon},$$

we have

$$\frac{\partial \boldsymbol{x}_t^{(k)}}{\partial \theta} = \alpha_k(t) \frac{\partial \hat{\boldsymbol{x}}_{t_k}}{\partial \theta},$$

and hence

$$\nabla_\theta \mathcal{L}_{\text{gen}}^{(k)} = \mathbb{E}_{t,\boldsymbol{\epsilon},\hat{\boldsymbol{x}}_{t_k}} \left[\lambda_k(t)\alpha_k(t) \left(\nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x}_t^{(k)}) - s_\psi(\boldsymbol{x}_t^{(k)}, t) \right)^\top \frac{\partial \hat{\boldsymbol{x}}_{t_k}}{\partial \theta} \right].$$

Because the score of $p_{\theta,t}^{(k)}$ is intractable, we approximate it with a learned fake score

$$\nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x}) \approx s_\phi(\boldsymbol{x}, t),$$

which yields the practical score-difference gradient

$$\nabla_\theta \mathcal{L}_{\text{gen}}^{(k)} \approx -\mathbb{E}_{t,\boldsymbol{\epsilon},\hat{\boldsymbol{x}}_{t_k}} \left[\lambda_k(t)\alpha_k(t) \left(s_\psi(\boldsymbol{x}_t^{(k)}, t) - s_\phi(\boldsymbol{x}_t^{(k)}, t) \right)^\top \frac{\partial \hat{\boldsymbol{x}}_{t_k}}{\partial \theta} \right].$$

Learning the fake score within $[t_k, 1]$. Since clean samples $\boldsymbol{x}_0$ are unavailable when $t_k > 0$, the fake score must be trained using the generated endpoint $\hat{\boldsymbol{x}}_{t_k}$ rather than a global clean target. Restating Eq. (10), the fake-score model s_ϕ is fit on the subinterval $[t_k, 1]$ via the DSM objective

$$\mathcal{L}_{\text{fake}}^{(k)}(\phi) = \mathbb{E}_{t\sim\mathcal{U}[t_k,1],\,\hat{\boldsymbol{x}}_{t_k},\,\boldsymbol{\epsilon}} \left[\omega_k(t) \left\| s_\phi(\boldsymbol{x}_t^{(k)}, t) - \nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x}_t^{(k)} \mid \hat{\boldsymbol{x}}_{t_k}) \right\|_2^2 \right],$$

which is unbiased for the marginal score $\nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}$ (App. D); reusing the full-interval objective with a mismatched conditioning would in general yield a biased estimator.

In practice, we parametrize s_ϕ via an x-prediction model f_ϕ that predicts $\hat{\boldsymbol{x}}_{t_k}$. Under the Gaussian subinterval kernel

$$q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k}) = \mathcal{N}\big(\boldsymbol{x};\, \alpha_k(t)\hat{\boldsymbol{x}}_{t_k},\, \sigma_k(t)^2\mathbf{I}\big),$$

the score estimator follows from Tweedie's formula:

$$s_\phi(\boldsymbol{x}_t^{(k)}, t) = -\frac{\boldsymbol{x}_t^{(k)} - \alpha_k(t)\, f_\phi(\boldsymbol{x}_t^{(k)}, t)}{\sigma_k(t)^2}.$$

C Coefficient-Preserving Sampling Formula

Under the rectified-flow modeling, $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, so $\alpha_t = 1 - t$ and $\sigma_t = t$. Given the current latent $\hat{\boldsymbol{x}}_{t_{k-1}}$ at noise level t_{k-1}, the generator produces an x-prediction $\hat{\boldsymbol{x}}_{\text{pred}}^{(k)} = G_\theta(\hat{\boldsymbol{x}}_{t_{k-1}}, t_{k-1})$ and a corresponding noise estimate

$$\hat{\boldsymbol{\epsilon}}^{(k)} = \frac{\hat{\boldsymbol{x}}_{t_{k-1}} - (1 - t_{k-1})\hat{\boldsymbol{x}}_{\text{pred}}^{(k)}}{t_{k-1}}. \tag{19}$$

coefficient-preserving sampling (CPS) (Wang & Yu, 2025) constructs the next latent $\hat{\boldsymbol{x}}_{t_k}$ by enforcing that the signal and noise coefficients exactly match the scheduler at time t_k:

$$\hat{\boldsymbol{x}}_{t_k} = (1 - t_k)\, \hat{\boldsymbol{x}}_{\text{pred}}^{(k)} + t_k \cos\!\Big(\frac{\eta\pi}{2}\Big)\hat{\boldsymbol{\epsilon}}^{(k)} + t_k \sin\!\Big(\frac{\eta\pi}{2}\Big)\boldsymbol{\epsilon}_k, \tag{20}$$

where $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is fresh noise and $\eta \in [0, 1]$ controls stochasticity. Setting $\eta = 0$ recovers the deterministic Euler ODE scheduler $\hat{\boldsymbol{x}}_{t_k} = (1 - t_k)\hat{\boldsymbol{x}}_{\text{pred}}^{(k)} + t_k\hat{\boldsymbol{\epsilon}}^{(k)}$, while $\eta = 1$ fully replaces the old noise with fresh samples, recovering the consistency model (CM) scheduler (Song et al., 2023). CPS thus provides a unified formulation that subsumes both common DMD (Yin et al., 2024a;b) sampling (*i.e.*, Euler and CM) strategies as special cases.

Substituting $\hat{\boldsymbol{\epsilon}}^{(k)}$ from Eq. (19) into Eq. (20) and collecting terms gives

$$\begin{aligned}
\hat{\boldsymbol{x}}_{t_k} &= (1 - t_k)\, \hat{\boldsymbol{x}}_{\text{pred}}^{(k)} + \frac{t_k}{t_{k-1}} \cos\!\Big(\frac{\eta\pi}{2}\Big) \big[\hat{\boldsymbol{x}}_{t_{k-1}} - (1 - t_{k-1})\hat{\boldsymbol{x}}_{\text{pred}}^{(k)}\big] + t_k \sin\!\Big(\frac{\eta\pi}{2}\Big)\boldsymbol{\epsilon}_k \\
&= \Big[1 - t_k + t_k \cos\!\Big(\frac{\eta\pi}{2}\Big)\Big(1 - \frac{1}{t_{k-1}}\Big)\Big]\hat{\boldsymbol{x}}_{\text{pred}}^{(k)} + \frac{t_k}{t_{k-1}} \cos\!\Big(\frac{\eta\pi}{2}\Big)\hat{\boldsymbol{x}}_{t_{k-1}} + t_k \sin\!\Big(\frac{\eta\pi}{2}\Big)\boldsymbol{\epsilon}_k.
\end{aligned} \tag{21}$$

The transition defines a Gaussian policy with mean $\boldsymbol{\mu}_\theta^{(k)} = \hat{\boldsymbol{x}}_{t_k}\big|_{\boldsymbol{\epsilon}_k = \mathbf{0}}$ and standard deviation $\sigma_k = t_k \sin(\eta\pi/2)$.

D Optimality of Ambient Denoising Score Matching

We show that the denoising score matching (DSM) objective in Eq. (10) has the true student marginal score $\nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x})$ as its unique optimal solution.

Proposition D.1. *Let $p_{\theta,t}^{(k)}(\boldsymbol{x}) = \int q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\, p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})\, d\hat{\boldsymbol{x}}_{t_k}$ denote the student marginal. Then, over functions $s : \mathbb{R}^d \times [t_k, 1] \to \mathbb{R}^d$,*

$$\arg\min_s \; \mathbb{E}_{t\sim\mathcal{U}[t_k,1]}\, \mathbb{E}_{\hat{\boldsymbol{x}}_{t_k}\sim p_\theta^{(k)},\, \boldsymbol{x}\sim q_t^{(k)}(\cdot\mid\hat{\boldsymbol{x}}_{t_k})} \left[\omega_k(t)\, \left\| s(\boldsymbol{x},t) - \nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k}) \right\|^2\right]$$

is achieved at $s^*(\boldsymbol{x},t) = \nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x})$ for almost every $t \in [t_k, 1]$.

Proof. Since s is a function of both $(\boldsymbol{x}, t)$, the outer expectation over t decouples and the minimization can be carried out independently for each t. Indeed, the full objective can be written as

$$\frac{1}{1-t_k}\int_{t_k}^1 \omega_k(t)\, \mathcal{J}_t\big(s(\cdot,t)\big)\, dt, \qquad \mathcal{J}_t(f) := \mathbb{E}_{\hat{\boldsymbol{x}}_{t_k},\,\boldsymbol{x}}\left[\|f(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\|^2\right].$$

For each fixed $t \in [t_k, 1]$, $\omega_k(t) > 0$ is a strictly positive constant and does not affect the inner argmin, so it suffices to minimize $\mathcal{J}_t(f)$ over $f : \mathbb{R}^d \to \mathbb{R}^d$.

Since the joint density of $(\boldsymbol{x}, \hat{\boldsymbol{x}}_{t_k})$ is $p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})\, q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})$, we can write

$$\mathcal{J}_t(f) = \int p_{\theta,t}^{(k)}(\boldsymbol{x})\, \mathbb{E}_{\hat{\boldsymbol{x}}_{t_k}\sim p(\hat{\boldsymbol{x}}_{t_k}\mid\boldsymbol{x})}\left[\|f(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\|^2\right] d\boldsymbol{x},$$

where $p(\hat{\boldsymbol{x}}_{t_k} \mid \boldsymbol{x}) = \frac{q_t^{(k)}(\boldsymbol{x}\mid\hat{\boldsymbol{x}}_{t_k})\, p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})}{p_{\theta,t}^{(k)}(\boldsymbol{x})}$ is the posterior. For each $\boldsymbol{x}$, the inner expectation is minimized pointwise by choosing $f(\boldsymbol{x})$ equal to the conditional mean:

$$f^*(\boldsymbol{x}) = \mathbb{E}_{\hat{\boldsymbol{x}}_{t_k}\sim p(\hat{\boldsymbol{x}}_{t_k}\mid\boldsymbol{x})}\left[\nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\right].$$

It remains to show that this equals the marginal score. By Bayes' rule,

$$\begin{aligned}
\nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} \log \int q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\, p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})\, d\hat{\boldsymbol{x}}_{t_k} \\
&= \frac{\int \nabla_{\boldsymbol{x}}\, q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\, p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})\, d\hat{\boldsymbol{x}}_{t_k}}{p_{\theta,t}^{(k)}(\boldsymbol{x})} \\
&= \int \frac{q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\, p_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k})}{p_{\theta,t}^{(k)}(\boldsymbol{x})}\, \nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\, d\hat{\boldsymbol{x}}_{t_k} \\
&= \mathbb{E}_{\hat{\boldsymbol{x}}_{t_k}\sim p(\hat{\boldsymbol{x}}_{t_k}\mid\boldsymbol{x})}\left[\nabla_{\boldsymbol{x}} \log q_t^{(k)}(\boldsymbol{x} \mid \hat{\boldsymbol{x}}_{t_k})\right] = f^*(\boldsymbol{x}).
\end{aligned}$$

Hence the pointwise optimum $s^*(\boldsymbol{x},t) = \nabla_{\boldsymbol{x}} \log p_{\theta,t}^{(k)}(\boldsymbol{x})$ holds for almost every $t \in [t_k, 1]$. □

E Self-Consistency of the Optimal Fake Score

We prove that the optimal fake score model satisfies the self-consistency property invoked in Sec. 3.1.

Proposition E.1 (Self-consistency of optimal denoiser)**.** *Let $\hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_t, t)$ denote a score model in x-prediction form trained under the flow matching framework. If $\hat{\boldsymbol{x}}_\phi$ is optimal, i.e.,*

$$\hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_t, t) = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t], \qquad \forall\, t,$$

then for any $t'' < t'$:

$$\hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_{t'}, t') = \mathbb{E}_{\tilde{\boldsymbol{x}}_{t''}\sim p(\boldsymbol{x}_{t''}\mid\boldsymbol{x}_{t'})}\left[\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}, t'')\right].$$

Proof. Under the forward process, $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ with $\boldsymbol{x}_0 \sim p_0$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and (α_t, σ_t) specifying the noise schedule (*e.g.*, $\alpha_t = 1 - t$, $\sigma_t = t$ for the rectified-flow schedule). For $t'' < t'$, the Gaussian interpolation implies that $(\boldsymbol{x}_0, \boldsymbol{x}_{t''}, \boldsymbol{x}_{t'})$ forms a Markov chain in the increasing-noise direction, yielding

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_{t''}, \boldsymbol{x}_{t'}] = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_{t''}].$$

Starting from the right-hand side and applying optimality at t'':

$$\begin{aligned}\mathbb{E}_{\tilde{\boldsymbol{x}}_{t''} \sim p(\boldsymbol{x}_{t''} \mid \boldsymbol{x}_{t'})}\big[\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}, t'')\big] &= \mathbb{E}\big[\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_{t''}] \,\big|\, \boldsymbol{x}_{t'}\big] \\ &= \mathbb{E}\big[\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_{t''}, \boldsymbol{x}_{t'}] \,\big|\, \boldsymbol{x}_{t'}\big] \\ &= \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_{t'}] = \hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_{t'}, t').\end{aligned}$$

The second equality uses the Markov property; the third is the tower rule. □

The consistency regularizer $\mathcal{L}_{\text{cons}}^{(k)}$ (Eq. (11)) penalizes deviations from this property, constraining the fake score model toward the optimal solution even as the generator distribution shifts during training.

F Practical Consistency Loss Estimator

Computing $\mathcal{L}_{\text{cons}}^{(k)}$ (Eq. (11)) requires evaluating $\mathbb{E}_{\tilde{\boldsymbol{x}}_{t''}}[\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}, t'')]$, which is intractable in closed form. Following Daras et al. (2024), we draw two independent reverse-step samples $\tilde{\boldsymbol{x}}_{t''}^1, \tilde{\boldsymbol{x}}_{t''}^2 \sim p_\phi(\boldsymbol{x}_{t''} \mid \boldsymbol{x}_{t'})$ to obtain the unbiased estimator:

$$\mathcal{L}_{\text{cons}}^{(k)}(\phi) \approx \mathbb{E}_{\boldsymbol{x}_{t'}, \tilde{\boldsymbol{x}}_{t''}^1, \tilde{\boldsymbol{x}}_{t''}^2}\left[\left(\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}^1, t'') - \hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_{t'}, t')\right)^\top \left(\hat{\boldsymbol{x}}_\phi(\tilde{\boldsymbol{x}}_{t''}^2, t'') - \hat{\boldsymbol{x}}_\phi(\boldsymbol{x}_{t'}, t')\right)\right]. \tag{22}$$

In our experiments, we set $t' = t_k$ and $t'' = t_{k+1}$ (adjacent endpoints of the timestep schedule), following Daras et al. (2024).

G Hybrid Policy Gradient

G.1 Derivation

We provide a formal derivation of the gradient estimator used for the reward term

$$\nabla_\theta \mathbb{E}_{\hat{\boldsymbol{x}}_0 \sim p_\theta}[r(\hat{\boldsymbol{x}}_0)].$$

Since the few-step generator consists of $K-1$ stochastic transitions followed by one deterministic final step, the resulting gradient decomposes naturally into a contribution from the stochastic transitions and a contribution from the deterministic terminal mapping.

Proposition G.1 (Hybrid policy gradient)**.** *Consider the few-step generative process*

$$\hat{\boldsymbol{x}}_{t_0} \to \hat{\boldsymbol{x}}_{t_1} \to \cdots \to \hat{\boldsymbol{x}}_{t_{K-1}} \to \hat{\boldsymbol{x}}_0,$$

where for $k = 1, \ldots, K-1$,

$$\pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}) = \mathcal{N}\big(\hat{\boldsymbol{x}}_{t_k}; \boldsymbol{\mu}_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_{k-1}}), \sigma_k^2 \mathbf{I}\big),$$

and the final output is deterministic:

$$\hat{\boldsymbol{x}}_0 = G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1}).$$

Define

$$J(\theta) := \mathbb{E}_{\hat{\boldsymbol{x}}_0 \sim p_\theta}[r(\hat{\boldsymbol{x}}_0)].$$

Then

$$\begin{aligned}\nabla_\theta J(\theta) = &\sum_{k=1}^{K-1} \mathbb{E}_{\tau \sim p_\theta}\left[r(\hat{\boldsymbol{x}}_0)\, \nabla_\theta \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}})\right] \\ &+ \mathbb{E}_{\tau \sim p_\theta}\left[\left(\nabla_{\hat{\boldsymbol{x}}_0} r(\hat{\boldsymbol{x}}_0)\right)^\top \partial_\theta G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1})\right].\end{aligned} \tag{23}$$

Proof. Because the last step is deterministic, it is convenient to separate the stochastic and deterministic parts of the generation process. Let

$$\zeta := (\hat{\boldsymbol{x}}_{t_1}, \dots, \hat{\boldsymbol{x}}_{t_{K-1}})$$

denote the intermediate random states. Conditioned on $\hat{\boldsymbol{x}}_{t_0}$, their density is

$$p_\theta(\zeta) = \prod_{k=1}^{K-1} \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}). \tag{24}$$

The terminal sample is then determined by

$$\hat{\boldsymbol{x}}_0 = G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1}),$$

and the objective can be written as

$$J(\theta) = \mathbb{E}_{\zeta \sim p_\theta}\left[r\big(G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1})\big)\right]. \tag{25}$$

Differentiating Eq. (25) under the integral sign gives

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \int p_\theta(\zeta)\, r\big(G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1})\big)\, d\zeta \\ &= \int \nabla_\theta p_\theta(\zeta)\, r(\hat{\boldsymbol{x}}_0)\, d\zeta + \int p_\theta(\zeta)\, \nabla_\theta r(\hat{\boldsymbol{x}}_0)\, d\zeta. \end{aligned} \tag{26}$$

Using

$$\nabla_\theta p_\theta(\zeta) = p_\theta(\zeta) \nabla_\theta \log p_\theta(\zeta),$$

we obtain

$$\nabla_\theta J(\theta) = \mathbb{E}_{\zeta \sim p_\theta}[r(\hat{\boldsymbol{x}}_0) \nabla_\theta \log p_\theta(\zeta)] + \mathbb{E}_{\zeta \sim p_\theta}[\nabla_\theta r(\hat{\boldsymbol{x}}_0)]. \tag{27}$$

By Eq. (24),

$$\log p_\theta(\zeta) = \sum_{k=1}^{K-1} \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}),$$

and therefore

$$\nabla_\theta \log p_\theta(\zeta) = \sum_{k=1}^{K-1} \nabla_\theta \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}). \tag{28}$$

Substituting Eq. (28) into Eq. (27) yields

$$\nabla_\theta J(\theta) = \sum_{k=1}^{K-1} \mathbb{E}_{\zeta \sim p_\theta}\left[r(\hat{\boldsymbol{x}}_0)\, \nabla_\theta \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}})\right] + \mathbb{E}_{\zeta \sim p_\theta}[\nabla_\theta r(\hat{\boldsymbol{x}}_0)]. \tag{29}$$

For the second term, $r(\hat{\boldsymbol{x}}_0)$ depends on θ through the final deterministic mapping G_θ. Hence, by the chain rule,

$$\nabla_\theta r(\hat{\boldsymbol{x}}_0) = \big(\nabla_{\hat{\boldsymbol{x}}_0} r(\hat{\boldsymbol{x}}_0)\big)^\top \partial_\theta G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1}). \tag{30}$$

Combining Eqs. (29) and (30) proves Eq. (23). □

G.2 Discussion

Prop. G.1 separates the gradient into two parts. The first part comes from how the parameters affect the distribution of the stochastic intermediate states, and therefore involves the log-derivatives of the transition densities. The second part comes from the explicit dependence of the final deterministic output on the parameters, and is obtained by directly differentiating the terminal mapping.

A subtle point is that the above derivation applies the log-derivative identity only to the stochastic part of the trajectory. One could formally write the full trajectory distribution as

$$p_\theta(\tau) = p(\hat{\boldsymbol{x}}_{t_0}) \prod_{k=1}^{K-1} \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}) \cdot \delta\big(\hat{\boldsymbol{x}}_0 - G_\theta(\hat{\boldsymbol{x}}_{t_{K-1}}, t_{K-1})\big),$$

but this representation contains a Dirac delta corresponding to the deterministic last step. For this reason, directly applying the log-derivative form to the entire trajectory is unnecessary and less clean. By isolating the random intermediate states ζ, the derivation remains fully standard and avoids manipulating the logarithm of a Dirac delta.

Another potential concern is whether the second term in Eq. (23) ignores the dependence of $\hat{\boldsymbol{x}}_{t_{K-1}}$ on θ. It does not. In Eq. (27), the dependence of the random trajectory ζ on θ is already accounted for by the term $\mathbb{E}_{\zeta\sim p_\theta}[r(\hat{\boldsymbol{x}}_0)\nabla_\theta \log p_\theta(\zeta)]$. Therefore, in the remaining term $\mathbb{E}_{\zeta\sim p_\theta}[\nabla_\theta r(\hat{\boldsymbol{x}}_0)]$, the sampled ζ is treated as fixed, and only the explicit parameter dependence of the terminal mapping G_θ is differentiated. Re-introducing the dependence of $\hat{\boldsymbol{x}}_{t_{K-1}}$ on θ in this term would count the same contribution twice.

H Background on GRPO and Its Application to Few-step Generators

Group relative policy optimization. GRPO (Shao et al., 2024) is a variance-reduced policy gradient method originally developed for language model alignment. Given a prompt c, GRPO samples a group of N responses, computes a group-normalized advantage for each, and updates the policy with a clipped objective. The key idea is to use the group mean reward as a baseline, eliminating the need for a separate value network.

Naive GRPO for few-step generators. In our setting, the few-step generator defines a K-step policy π_θ with Gaussian transitions (Sec. 3.2). A direct (naive) application of GRPO proceeds as follows: for each prompt c, sample N independent trajectories $\{\tau_i\}_{i=1}^N$ by drawing independent noise $\boldsymbol{\epsilon}_k^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at every stochastic step $k = 1, \dots, K-1$ of each trajectory i. Compute rewards $r_i = r(\hat{\boldsymbol{x}}_0^{(i)}, c)$ and group-normalized advantages

$$A_i = \frac{r_i - \bar{r}}{\mathrm{std}(\{r_j\}_{j=1}^N)}, \qquad \bar{r} = \frac{1}{N}\sum_{j=1}^N r_j.$$

The naive GRPO gradient estimator for the stochastic part is then

$$\nabla_\theta \mathcal{L}_{\text{stoc}}^{\text{GRPO}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K-1} A_i \, \nabla_\theta \log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k}^{(i)} \mid \hat{\boldsymbol{x}}_{t_{k-1}}^{(i)}), \tag{31}$$

where the Gaussian log-likelihood is

$$\log \pi_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_k} \mid \hat{\boldsymbol{x}}_{t_{k-1}}) = -\frac{1}{2\sigma_k^2}\|\hat{\boldsymbol{x}}_{t_k} - \boldsymbol{\mu}_\theta^{(k)}(\hat{\boldsymbol{x}}_{t_{k-1}})\|^2 + \text{const},$$

with $\boldsymbol{\mu}_\theta^{(k)}$ and $\sigma_k = t_k \sin(\eta\pi/2)$ determined by the CPS update (App. C).

Limitation. Since all $K-1$ steps use independent noise across trajectories, the reward difference between any two trajectories reflects the combined effect of noise injected at *all* steps simultaneously. The advantage A_i is therefore a noisy credit assignment signal, which means it cannot attribute reward variation to any specific step, leading to high gradient variance. This motivates our SubGRPO (Sec. 3.2), which shares noise at non-selected steps to isolate contributions from a chosen subset.

I Detailed Algorithm of RTDMD

We present a detailed algorithm pipeline of RTDMD in Algo. 1.

J More Implementation Details

For the 4-step generator ($K = 4$), we uniformly partition the interval $[0, 1]$ to obtain the pre-shift timestep schedule $[1.0, 0.75, 0.5, 0.25]$. For FLUX.2 4B, the Stage 1 generator is initialized from the FLUX.2 [klein] 4B checkpoint rather than the Base 4B model. All experiments use

Algorithm 1 RTDMD: Reward-Tilted Distribution Matching Distillation

Require: Teacher score s_ψ, generator G_θ, fake score s_ϕ, reward $r(\cdot)$, K-step schedule $\{t_k\}_{k=0}^{K}$, reward weight β.
Ensure: Reward-optimized K-step generator G_θ.

Stage I: Ambient-Consistent Distribution Matching Distillation (cold start)

for $u = 1, \ldots, T_{\text{cold}}$ **do**
  Roll out G_θ from noise; sample step index k and construct noised state $x_t^{(k)}$ for $t \in [t_k, 1]$.
  Update s_ϕ via the fake score objective (Eq. (12)).
  Update G_θ via the AC-DMD generator objective (Eq. (9)).
end for

Stage II: Reinforcing the Few-step Generator

for $u = 1, \ldots, T_{\text{rl}}$ **do**
  for each prompt c_j in batch **do**
    Sample step subset $S \subset \{1, \ldots, K-1\}$.
    Generate N trajectories with shared/independent noise (Eq. (16)).
    Compute rewards $r_i = r(\hat{x}_0^{(i)}, c_j)$ and group-normalized advantages A_i.
  end for
  Update s_ϕ via the fake score objective (Eq. (12)).
  Update G_θ by descending along $\nabla_\theta \mathcal{L}_{\text{total}}$ (Eq. (18)).
end for
return G_θ.

AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a constant learning rate. In Stage 1 (cold start), we use the same learning rate for the fake score model and generator. The learning rate is 1×10^{-6} for FLUX.2 4B and 3×10^{-5} for the SD series; in Stage 2, the fake score model retains its Stage 1 learning rate while the generator learning rate is uniformly set to 3×10^{-6}. LoRA (Hu et al., 2022) is applied to all linear layers within self-attention and cross-attention modules ($r = 64$, $\alpha = 32$). The GRPO clipping range is $[1 - 10^{-5},\ 1 + 10^{-5}]$, and the reward-tilting coefficient β is set to 1.0 throughout. For multi-reward training, each reward is weighted equally. All training and rollout use `bf16` precision. Stage 1 uses 8 NVIDIA H20 GPUs for all models; Stage 2 uses 16 GPUs for FLUX.2 4B and 8 GPUs for the SD series. We update the fake score model 10 times per generator update, and no GAN loss is used in any experiment. For CFG, only the teacher score is computed with CFG during training; the fake score model and the generator do not use CFG, and all samples are generated without CFG at inference.

K Discussion for Sampling Schedule

As shown in Fig. 5, we find that performance is relatively insensitive to the specific value of η as long as sufficient stochasticity is introduced (*i.e.*, $\eta \geq 0.8$), with all such configurations yielding comparable results. We attribute this to the fact that the subsequent GRPO optimization relies on adequate stochasticity in the intermediate transitions to produce diverse trajectories for effective exploration. We therefore set $\eta = 0.9$ throughout our experiments.

L GenEval Results

We further validate our framework using GenEval (Ghosh et al., 2023), a non-differentiable compositional generation benchmark, on `SD3.5-M`. Since the reward signal is non-differentiable, the hybrid policy gradient cannot backpropagate through the final deterministic step; optimization relies solely on AC-DMD and SubGRPO for the stochastic transitions. Despite this, as shown in Tab. 5, our method achieves an overall score of 0.94 with only 4 NFE, nearly matching Flow-GRPO (Liu et al., 2026b) (0.95), which operates on the full multi-step model with 40 NFE and CFG (Ho & Salimans, 2021), while outperforming TDM-R1 (Luo et al., 2026) (0.92) under the same 4-step setting.

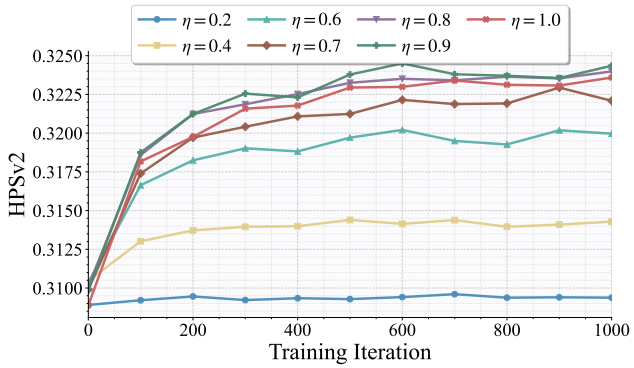


Figure 5: Evaluation curves when reinforcing the few-step generators with different η. We use `SD3.5-M` (Stability AI, 2024) with HPSv2 (Wu et al., 2023) as the sole training reward. Each curve is optimized by A-DMD with different η values combined with GRPO on stochastic transitions. $\eta = 1.0$ denotes original DMD2 (Yin et al., 2024a), which achieves similar performance as $\eta = 0.9$ (default setting in this paper).

Table 5: GenEval results. Results for models other than `SD3.5-M` are from Luo et al. (2026) or original papers.

Model	Overall	Single Object	Two Object	Counting	Colors	Position	Attribution Binding
Autoregressive Models							
Show-o (Xie et al., 2025b)	0.53	0.95	0.52	0.49	0.82	0.11	0.28
Emu3-Gen (Wang et al., 2024b)	0.54	0.98	0.71	0.34	0.81	0.17	0.21
JanusFlow (Ma et al., 2025a)	0.63	0.97	0.59	0.45	0.83	0.53	0.42
Janus-Pro-7B (Chen et al., 2025c)	0.80	0.99	0.89	0.59	0.90	0.79	0.66
GPT-4o (Hurst et al., 2024)	0.84	0.99	0.92	0.85	0.92	0.75	0.61
Diffusion Models							
LDM (Rombach et al., 2022)	0.37	0.92	0.29	0.23	0.70	0.02	0.05
`SD1.5` (Rombach et al., 2022)	0.43	0.97	0.38	0.35	0.76	0.04	0.06
`SD2.1` (Rombach et al., 2022)	0.50	0.98	0.51	0.44	0.85	0.07	0.17
`SD-XL` (Podell et al., 2024)	0.55	0.98	0.74	0.39	0.85	0.15	0.23
DALLE-2 (OpenAI, 2023)	0.52	0.94	0.66	0.49	0.77	0.10	0.19
DALLE-3 (Betker et al., 2023)	0.67	0.96	0.87	0.47	0.83	0.43	0.45
`FLUX.1 Dev` (Black Forest Labs, 2024)	0.66	0.98	0.81	0.74	0.79	0.22	0.45
`SD3.5-L` (Esser et al., 2024)	0.71	0.98	0.89	0.73	0.83	0.34	0.47
SANA-1.5 4.8B (Xie et al., 2025a)	0.81	0.99	0.93	0.86	0.84	0.59	0.65
`SD3.5-M` (Esser et al., 2024)	0.63	0.98	0.78	0.50	0.81	0.24	0.52
w/ Flow-GRPO (Liu et al., 2026b)	0.95	1.00	0.99	0.95	0.92	0.99	0.86
Few-Step Diffusion Models (4 NFE)							
`SD3.5-L-Turbo` (Esser et al., 2024)	0.70	0.94	0.84	0.55	0.79	0.58	0.56
`SD3.5-M` w/ TDM (Luo et al., 2025)	0.61	0.99	0.77	0.49	0.79	0.23	0.44
`SD3.5-M` w/ TDM-R1 (Luo et al., 2026)	0.92	1.00	0.96	0.88	0.85	0.93	0.91
`SD3.5-M` w/ RTDMD (Ours)	**0.94**	**1.00**	**0.98**	**0.95**	**0.90**	**0.87**	**0.94**

M Additional Ablation Studies

Unless otherwise specified, the experiments in this section follow the ablation setting of Sec. 4.3, *i.e.*, `SD3.5-M` (Stability AI, 2024) trained with CLIPScore, PickScore, and HPSv2 as rewards and evaluated on DrawBench (Saharia et al., 2022).

M.1 Disentangling Ambient Matching and Consistency Regularization

This experiment shows that both components of AC-DMD contribute to its advantage. Complementing Tab. 3, which isolates the consistency regularizer under A-DMD, we add full-interval DMD2 (Yin et al., 2024a) baselines with and without $\mathcal{L}_{\text{cons}}^{(k)}$: DMD2 treats each intermediate x-prediction $\hat{\boldsymbol{x}}_{\text{pred}}^{(k)}$ as a clean sample and matches distributions over $[0, 1]$, whereas A-DMD matches the realized CPS state $\hat{\boldsymbol{x}}_{t_k}$ over $[t_k, 1]$ (Sec. 3.1). All configurations share the same settings. We report cold-start metrics,

which reflect distribution matching itself, together with final Stage 2 metrics, which reflect its effect on reward optimization.

Table 6: Separating subinterval (ambient) matching and consistency regularization on `SD3.5-M`. Blue row denotes the default setting of our RTDMD.

Method	Matching interval	$\mathcal{L}_{\text{cons}}^{(k)}$	Cold start			Final (Stage 2)		
			CLIPScore↑	PickScore↑	HPSv2↑	CLIPScore↑	PickScore↑	HPSv2↑
DMD2 (Yin et al., 2024a)	$[0, 1]$	✗	0.2732	21.3219	0.2747	0.2729	21.9123	0.3006
DMD2 + $\mathcal{L}_{\text{cons}}^{(k)}$	$[0, 1]$	✓	<u>0.2804</u>	<u>22.2162</u>	0.2926	<u>0.2836</u>	<u>23.0238</u>	<u>0.3295</u>
A-DMD	$[t_k, 1]$	✗	0.2768	21.6584	**0.2965**	0.2753	22.0116	0.3188
AC-DMD	$[t_k, 1]$	✓	**0.2822**	**22.5056**	<u>0.2942</u>	**0.2843**	**23.5690**	**0.3456**

As shown in Tab. 6, the two components are complementary. Ambient matching improves every metric at both stages, with or without the consistency regularizer, while the consistency regularizer yields the larger gains in the final stage (*e.g.*, $+1.11$ and $+1.56$ PickScore when added to DMD2 and A-DMD, respectively). Combining them, AC-DMD achieves the best final results on all three metrics. In addition to these gains, ambient matching keeps distribution matching valid for any CPS scheduler (Sec. 3.1 and App. B).

M.2 Fake-Score Tracking under a Moving Generator Distribution

This experiment verifies that the gain of the consistency regularizer comes from the mechanism motivated in Sec. 3.1, *i.e.*, better tracking of the shifting generator distribution under a limited number of fake-score updates. We use AC-DMD generator checkpoints of `SD3.5-M` saved at 1,000, 2,000, 3,000, and 4,000 training steps. In the *moving* setting, a fake score model is trained sequentially for 2,000 updates on each checkpoint (8,000 updates in total); in the *stationary* setting, it is trained for 8,000 updates only on the final 4,000-step generator. Each setting is trained with and without $\mathcal{L}_{\text{cons}}^{(k)}$ from the same initialization and with the same data and optimization settings. We report the prediction MSE to a reference fake score model, which shares the initialization and is trained to convergence for 20,000 updates with $\mathcal{L}_{\text{fake}}^{(k)}$ on the frozen final generator, averaged over 1,000 held-out samples from the final generator.

Table 7: Fake-score tracking under stationary and moving generator distributions on `SD3.5-M`. The prediction MSE is measured against a converged reference fake score model.

Target distribution	$\mathcal{L}_{\text{cons}}^{(k)}$	Fake-score updates	Prediction MSE↓
Stationary: $p_{\theta_{4000}}^{(k)}$	✗	8,000	0.1284
Stationary: $p_{\theta_{4000}}^{(k)}$	✓	8,000	**0.1158**
Moving: $p_{\theta_{1000}}^{(k)} \to p_{\theta_{2000}}^{(k)} \to p_{\theta_{3000}}^{(k)} \to p_{\theta_{4000}}^{(k)}$	✗	$4 \times 2{,}000$	0.2875
Moving: $p_{\theta_{1000}}^{(k)} \to p_{\theta_{2000}}^{(k)} \to p_{\theta_{3000}}^{(k)} \to p_{\theta_{4000}}^{(k)}$	✓	$4 \times 2{,}000$	**0.1647**

As shown in Tab. 7, $\mathcal{L}_{\text{cons}}^{(k)}$ reduces the prediction MSE by 42.7% ($0.2875 \to 0.1647$) when the target moves, compared with 9.8% ($0.1284 \to 0.1158$) for a stationary target; the absolute reduction is $9.7\times$ larger in the moving setting. The regularizer thus specifically improves tracking of the evolving generator distribution, which is the regime that online fake-score training operates in.

M.3 Comparison of Training Recipes

This experiment justifies the two-stage recipe of RTDMD, which first performs AC-DMD as a cold start and then jointly optimizes the distribution matching and reward terms. The cold start adapts the teacher-initialized generator to 4-step sampling, providing a strong initialization and a reliable fake score, while Stage 2 retains AC-DMD as a teacher-distribution constraint during reward optimization. We compare this recipe with three alternatives on `SD3.5-M`: (1) jointly optimizing AC-DMD and the reward from initialization, which exposes the reward model to severely degraded

samples from the unadapted generator, so its signal can misdirect the student before a reliable cold start is established; (2) optimizing the reward on the multi-step teacher and then distilling it with AC-DMD, which requires expensive multi-step rollouts and risks losing reward-aligned capabilities during distillation; and (3) performing AC-DMD and then optimizing only the reward, which removes the teacher-distribution constraint and exposes training to drift and reward hacking.

Table 8: Comparison of training recipes on `SD3.5-M`. Blue row denotes the default setting of our RTDMD.

Training recipe	CLIPScore↑	PickScore↑	HPSv2↑
Joint AC-DMD and reward optimization from initialization	0.2528	21.3249	0.2836
Reward optimization on the teacher, then AC-DMD	0.2786	22.9862	0.3147
AC-DMD, then reward optimization only	**0.2847**	<u>23.3853</u>	<u>0.3324</u>
AC-DMD, then joint optimization (Ours)	<u>0.2843</u>	**23.5690**	**0.3456**

As shown in Tab. 8, our recipe achieves the best PickScore and HPSv2. The cold start is essential: compared with joint optimization from initialization, it improves PickScore by 2.24 and HPSv2 by 0.062. Retaining AC-DMD in Stage 2 further improves PickScore from 23.39 to 23.57 and HPSv2 from 0.3324 to 0.3456 over reward-only optimization, showing the benefit of the teacher-distribution constraint. Distilling a reward-optimized teacher reaches only 22.99 PickScore despite its expensive multi-step rollouts.

M.4 SubGRPO with More Steps and Different Step Selections

This experiment shows that SubGRPO scales to more sampling steps and identifies uniform resampling of the step subset as its key design choice. Because $\mathcal{S}$ is uniformly resampled for each prompt, all stochastic steps are covered in expectation for any K. Under a matched training time on `SD3.5-M`, we compare (i) SubGRPO with a uniformly sampled subset of $M = 2$ steps against updating all stochastic transitions ($M = 7$) for an 8-step generator ($K = 8$), and (ii) uniform sampling with $M = 1$ against fixing the selected step to the first or the last stochastic transition for the default 4-step generator.

Table 9: SubGRPO with more sampling steps and different step-selection strategies on `SD3.5-M` under a matched training time. The best results within each K are in bold.

Step selection	K	M	CLIPScore↑	PickScore↑	HPSv2↑
All stochastic steps	8	7	0.2813	23.3274	0.3319
Uniform random	8	2	**0.2899**	**23.4954**	**0.3447**
Uniform random	4	1	**0.2798**	**23.3272**	**0.3332**
Fixed first stochastic step	4	1	0.2751	23.1678	0.3186
Fixed last stochastic step	4	1	0.2747	22.8436	0.3044

As shown in Tab. 9, for $K = 8$, SubGRPO with $M = 2$ outperforms updating all seven stochastic steps on all metrics, so SubGRPO remains effective for longer trajectories. Uniform resampling is essential: fixing the selected step to the first or the last stochastic transition underperforms uniform sampling, because the remaining steps then receive no direct credit assignment, whereas uniform resampling optimizes the model at all noise levels in a balanced way.

M.5 Robustness to the Number of Groups per Iteration

This experiment rules out the number of groups per iteration as a source of the gains on FLUX.2 4B (Black Forest Labs, 2025). The group size is fixed to $N = 24$ in all experiments, and only the number of groups per iteration differs across backbones (48 for the SD series and 64 for FLUX.2 4B; see Sec. 4.1). We train RTDMD on FLUX.2 4B with 32, 48, and 64 groups per iteration while keeping the total number of groups used throughout training identical and all other settings fixed. As shown in Tab. 10, the three configurations perform nearly identically, so RTDMD is robust to this choice. We use 64 groups by default because this configuration more fully utilizes the available GPU memory.

Table 10: Effect of the number of groups per iteration on FLUX.2 4B with a fixed total number of groups (group size $N = 24$). Blue row denotes the default setting of our RTDMD.

Groups per iteration	ImageReward↑	CLIPScore↑	HPSv2↑	GenEval↑
32	1.3714	0.3215	0.3519	0.9038
48	1.3723	0.3223	0.3514	0.9046
64 (default)	1.3712	0.3219	0.3516	0.9046

N Additional Comparisons and Analyses

N.1 Comparison with Baselines on SD3.5-M

Tab. 11 extends the comparison of Tab. 1 to `SD3.5-M` with the same five metrics, covering multi-step references and representative few-step distillation and RL methods. RTDMD outperforms all few-step distillation and RL baselines on all five metrics, *e.g.*, by 0.45 PickScore and 0.021 HPSv2 over the best baseline on each metric. Since it shares the backbone and evaluation setting of Tabs. 3–4, Tab. 11 also serves as the matched reference for the `SD3.5-M` ablations: under these settings, AC-DMD improves over DMD2 on all metrics at both the cold-start and final stages (Tab. 6).

Table 11: Quantitative comparison on `SD3.5-M` under the evaluation setting of Sec. 4.3. NFE counts the conditional and unconditional evaluations separately when CFG is used, so 40 sampling steps with CFG correspond to 80 NFE. The best and second-best few-step results are in bold and underlined, respectively.

Method	NFE	CLIPScore↑	Aesthetic↑	PickScore↑	HPSv2↑	ImageReward↑
`SD3.5-M` (w/o CFG) (Stability AI, 2024)	40	0.2357	5.1777	20.5478	0.2077	−0.5381
`SD3.5-M` (w/ CFG) (Stability AI, 2024)	80	0.2839	5.3905	22.3912	0.2806	0.8954
Few-Step Distillation						
DMD2 (Yin et al., 2024a)	4	0.2732	5.5536	21.3219	0.2747	1.0247
TDM (Luo et al., 2025)	4	0.2689	5.5461	21.3145	0.2751	1.0418
Few-Step RL						
DMDR (Jiang et al., 2025)	4	0.2765	5.5134	23.0391	0.3187	1.4090
GDMD (Dong et al., 2026)	4	0.2764	5.7389	22.9439	<u>0.3249</u>	<u>1.4563</u>
R_{dm} (Fan et al., 2026a)	4	<u>0.2792</u>	<u>5.7436</u>	<u>23.1225</u>	0.3214	1.4395
RTDMD (Ours)	4	**0.2843**	**5.8250**	**23.5690**	**0.3456**	**1.5526**

N.2 Sample Diversity

We evaluate sample diversity and distributional fidelity on COCO-10K 2014 (Lin et al., 2014) for `SD3.5-M`, R_{dm} (Fan et al., 2026a), and RTDMD. Following Wu et al. (2026), we generate nine images per prompt from different initial noises and compute diversity as one minus the average pairwise cosine similarity of DINOv3-ViT-Large (Siméoni et al., 2025) and CLIP-ViT-Large (Radford et al., 2021) features. FID (Heusel et al., 2017) is computed against the COCO-10K 2014 reference images. As shown in Tab. 12, RTDMD outperforms R_{dm} on both diversity metrics and FID while preserving most of the diversity of the multi-step `SD3.5-M`.

Table 12: Sample diversity and FID on COCO-10K 2014. Diversity is one minus the average pairwise cosine similarity among nine images generated per prompt.

Method	DINO Diversity↑	CLIP Diversity↑	FID↓
`SD3.5-M` (Stability AI, 2024)	0.276	0.221	20.47
R_{dm} (Fan et al., 2026a)	0.229	0.187	23.63
RTDMD (Ours)	0.242	0.208	21.88

O QUALITATIVE COMPARISON FOR ABLATION

We provide qualitative visualizations corresponding to each ablation study in the main text. Specifically, Figs. 6–8 present samples generated by the configurations listed in Tabs. 3–4, respectively. None of the samples shown are cherry-picked, and all are produced using randomly drawn prompts and noise seeds. These visualizations offer direct visual evidence for the effectiveness of each proposed component, complementing the quantitative analysis in the main text.



Figure 6: Qualitative comparison of different distillation methods upon completion of cold-start training. From *left* to *right*: AC-DMD ($\gamma = 0.001$), AC-DMD ($\gamma = 0.005$), AC-DMD ($\gamma = 0.01$), AC-DMD ($\gamma = 0.1$), A-DMD. The columns correspond to the configurations in Tab. 3, listed in bottom-to-top order.



Figure 7: Qualitative comparison of different distillation methods upon completion of two-stage training. From *left* to *right*: AC-DMD ($\gamma = 0.001$), AC-DMD ($\gamma = 0.005$), AC-DMD ($\gamma = 0.01$), AC-DMD ($\gamma = 0.1$), A-DMD. The columns correspond to the rows in Tab. 3 in bottom-to-top order.

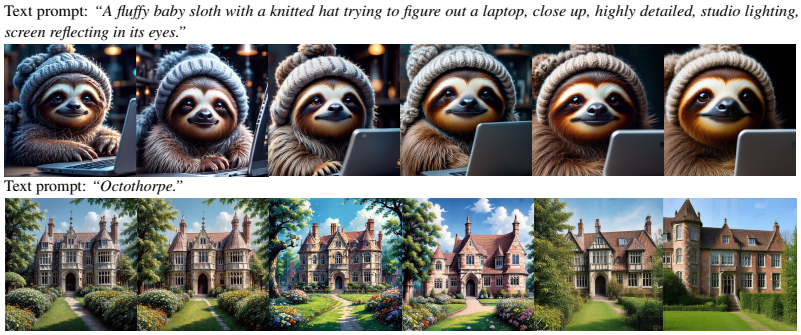
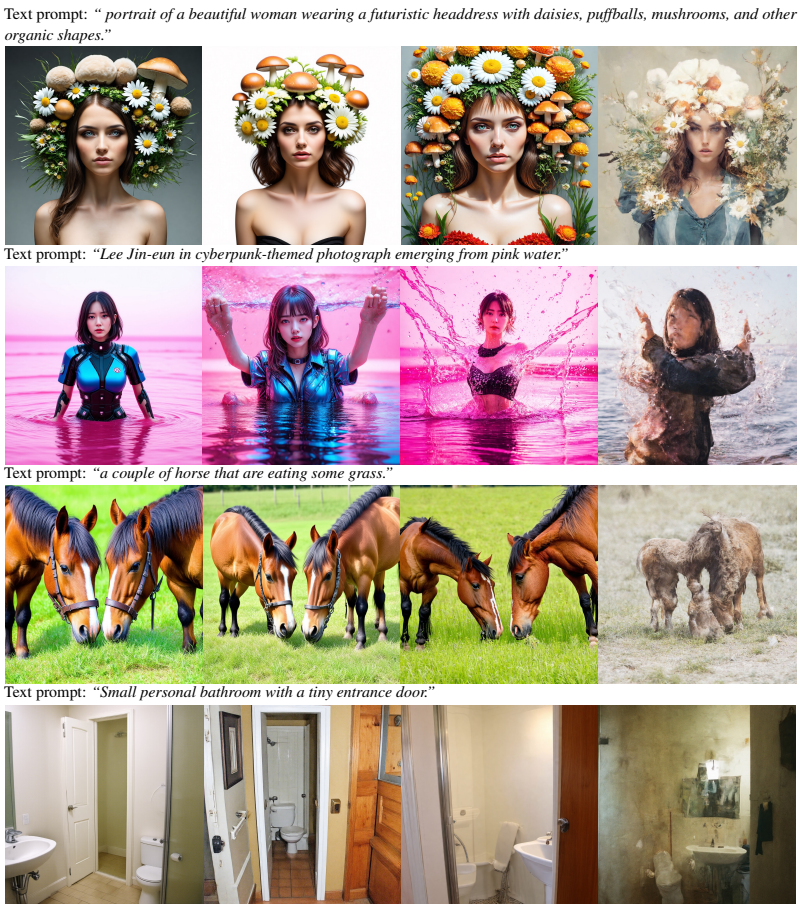


Figure 8: Qualitative comparison of different reinforcement learning methods upon completion of two-stage training. From *left* to *right*: RTDMD ($M = 2$), RTDMD ($M = 2$) w/o $\mathcal{L}_{\text{det}}$, RTDMD ($M = 1$), RTDMD w/o $\mathcal{L}_{\text{det}}$, GRPO (Shao et al., 2024), and $\emptyset$. The columns correspond to the rows in Tab. 4 in bottom-to-top order.

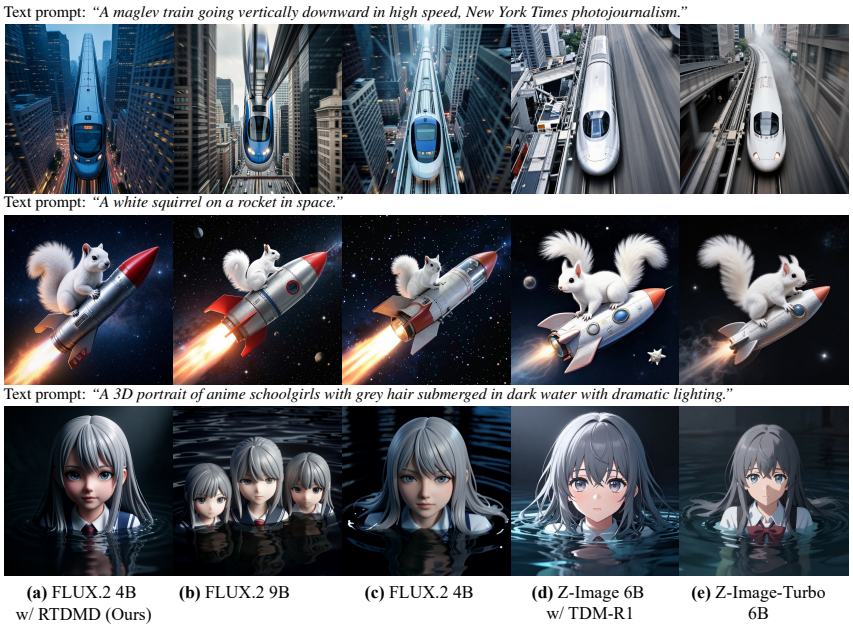
P MORE QUALITATIVE RESULTS

We provide additional visual comparisons in Fig. 9, Fig. 10, and Fig. 11. Across diverse prompts, our RTDMD consistently produces images with superior visual quality and prompt adherence compared to baselines.



(a) RTDMD (Ours) **(b)** R_{dm} **(c)** SD3-M (w/ CFG) **(d)** SD3-M (w/o CFG)

Figure 9: Qualitative comparison for SD3-M (Esser et al., 2024). Using identical noise inputs, our method outperforms others in both quality and prompt alignment, showing strong performance.



(a) FLUX.2 4B w/ RTDMD (Ours) **(b)** FLUX.2 9B **(c)** FLUX.2 4B **(d)** Z-Image 6B w/ TDM-R1 **(e)** Z-Image-Turbo 6B

Figure 10: Qualitative comparison for few-step diffusion models (4 NFE). Using identical noise inputs, our method outperforms others in both quality and prompt alignment, showing strong performance.



Figure 11: Visual generations produced by our RTDMD method under 4 NFE on FLUX.2 4B (Black Forest Labs, 2025) without applying classifier-free guidance (CFG) (Ho & Salimans, 2021).