PRECISE: SDE-Consistent Stochastic Sampling for RL Post-Training of Flow-Matching Models

Bo Peng[*†] Peking University
Tao Huang[*†] Tencent Hunyuan
Weijie Kong[*] Tencent Hunyuan
Junzhe Li[†] Peking University
Yue Wu Tencent Hunyuan
Qi Tian Tencent Hunyuan
Jiangfeng Xiong Tencent Hunyuan
Jianwei Zhang[‡] Tencent Hunyuan
Liefeng Bo Tencent Hunyuan
Zhao Zhong[§] Tencent Hunyuan

Abstract

Reinforcement learning (RL) has become an effective way to improve prompt alignment and perceptual quality in diffusion and flow-matching generators. A critical step for applying online RL to flow matching is turning the deterministic sampling trajectory into a stochastic policy, typically by replacing the reverse-time Ordinary Differential Equation (ODE) with a Stochastic Differential Equation (SDE). The stochastic sampler, controlling the exploration behavior and denoising dynamics, is thus part of the policy, and its design can significantly affect the reward optimization performance. We break down the sampler design into two interdependent components: choosing the right amount of stochastic exploration, and discretizing the resulting SDE faithfully at the small step counts used in RL. To address the first component, we analyze the inherent tension between exploration and stability in denoising and derive an SDE schedule that balances the two. Turning to the discretization challenge, we use a toy example to show that existing samplers can deviate from the flow-matching process, either by introducing excessive discretization noise or by relying on heuristic rules that do not guarantee convergence to the data distribution. To address these issues, we propose PRECISE, a new stochastic sampler that balances effective exploration with stability. Crucially, PRECISE keeps the denoising trajectory SDE-consistent through a novel approximation that freezes the clean-latent posterior mean, resolving the excess noise issue in standard samplers. Extensive experiments demonstrate that this formulation leads to significantly faster reward optimization via reinforcement learning, achieving state-of-the-art scores on the optimized rewards (e.g., PickScore, HPSv2.1) while requiring 13.1–53.2% less wall-clock training time to match the best in-domain performance of prior samplers.

1 Introduction

Diffusion and flow-matching models are central to visual generation. Diffusion models and score-based SDEs sample by reversing a gradual noising process [Ho et al., 2020, Song et al., 2020b], while flow matching and rectified flow cast generation as continuous transport [Lipman et al., 2022, Liu et al., 2022]. Although these pretrained generators provide strong visual priors, they do not

[*]Equal contribution.
[†]Work done during internship at Tencent Hunyuan.
[‡]Project Leader.
[§]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

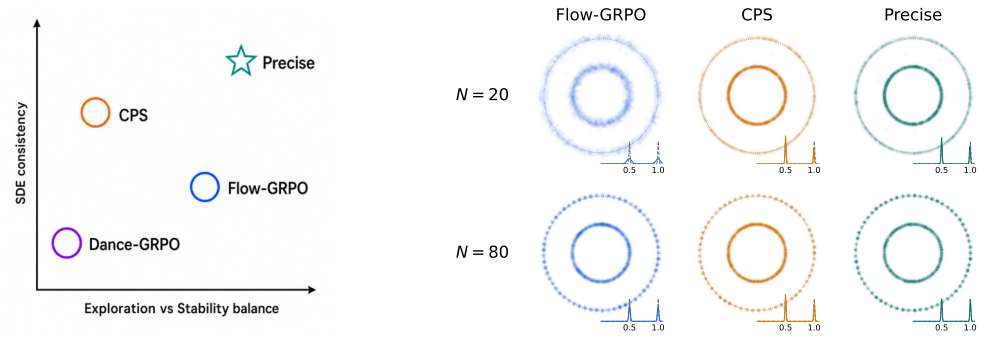

Figure 1: Left (conceptual schematic): Sampler design has two coupled axes: the exploration-stability balance and SDE consistency. Existing stochastic samplers occupy different trade-offs, while PRECISE improves both. Right: The double-ring example illustrates failure modes of existing samplers: Flow-GRPO [Liu et al., 2025] introduces excess noise, while CPS [Wang and Yu, 2025] biases the marginal distribution.

directly optimize downstream criteria such as prompt following, perceptual quality, and task-specific constraints. RL post-training addresses this gap by optimizing image-level rewards or preferences via reward supervision, preference optimization, and policy gradients [Xu et al., 2023, Wallace et al., 2024, Black et al., 2023, Fan et al., 2023], with recent extensions to online RL for flow-matching and visual generation models [Liu et al., 2025, Xue et al., 2025, Zheng et al., 2026b].

In online RL, the stochastic sampler is not merely an inference-time choice. It defines the policy whose action probabilities enter the policy-gradient objective [Liu et al., 2025, Xue et al., 2025], determines the exploration available for within-prompt advantage estimation, and controls the denoising trajectory on which the model makes predictions. Thus, for flow-matching models, converting the deterministic ODE into a stochastic sampler creates a sampler-design problem: the sampler must explore enough to support reward optimization while remaining faithful to the denoising dynamics learned during pretraining. This balance is especially delicate because online RL for diffusion and flow-matching models typically restricts the rollout to a small number of denoising steps (e.g., 6-30) to mitigate the massive computational bottleneck of iterative generation. Maintaining faithful trajectory exploration in this extreme low-NFE regime is not merely an efficiency trick, but a fundamental prerequisite for scaling RLHF to high-dimensional continuous domains [Liu et al., 2025, Xue et al., 2025].

To systematically resolve this tension and enable stable exploration under restricted budgets, we deconstruct the stochastic sampler design into two interdependent components. First, the exploration schedule must balance diversity and stability: if the injected noise is too small, sampled groups have limited reward variation and policy improvement is slow; if it is too large, the trajectory can become unstable and reward estimates become less reliable. Second, the finite-step discretization must remain consistent with the intended reverse SDE at a small step count, which becomes harder as the injected noise increases. If the discretization is inconsistent, the model can be forced to extrapolate beyond its training distribution, leading to poor sample quality.

Viewed through the lens of these two requirements, existing samplers expose critical shortcomings. For instance, the Euler-style sampler used in prior flow-matching RL pipelines [Liu et al., 2025, Xue et al., 2025] introduces stochasticity, but can add excess noise under coarse discretization. CPS [Wang and Yu, 2025] reduces this artifact by preserving the nominal signal and noise coefficients, but coefficient preservation alone does not ensure convergence to the correct data distribution, and the exploration-stability tradeoff still requires careful tuning. As shown in Figure 1, existing samplers occupy different tradeoff points in the design space.

To transcend these suboptimal tradeoffs, we propose PRECISE, a principled stochastic sampler explicitly designed for online RL post-training. Our framework resolves the tension through two key observations. First, exploration should respect the local geometry of the denoising process. A log signal-to-noise ratio (logSNR) analysis gives a principled time-varying schedule: inject more noise where the model can reliably return to the data manifold, and less where extra randomness would destabilize the trajectory. Second, the finite-step transition should account for the fact that injected noise immediately changes the latent state on which the velocity and score are predicted. Euler-style samplers freeze these quantities within a step, therefore missing this feedback. We instead observe that the clean-latent posterior mean $\hat{\mathbf{z}}_0(t) = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t]$ is more stable under noise injection, because estimating $\hat{\mathbf{z}}_0(t)$ is naturally a denoising task. This motivates freezing $\hat{\mathbf{z}}_0(t)$ rather than the velocity

or score, and computing the exact transition under this approximation. The resulting transition rule remains faithful to the intended SDE while avoiding excess discretization noise.

We empirically evaluate PRECISE on Stable Diffusion 3.5 Medium (SD3.5-M) [Esser et al., 2024] and FLUX.2 Klein 4B Base [Black Forest Labs, 2025, 2026]. PRECISE achieves state-of-the-art scores on the optimized rewards (e.g., PickScore, HPSv2.1) while requiring 13.1–53.2% less wall-clock training time to match the best in-domain performance of prior samplers, and remains competitive on held-out metrics. Additional ablations show that it is robust to hyperparameter and NFE changes, and that both the exploration schedule and the novel approximation contribute to its performance. Code is open-source at `https://github.com/Tencent-Hunyuan/Precise`, including the PRECISE sampler and reproduction scripts.

Our contributions are summarized as follows:

- We identify stochastic sampler design in RL as a fundamentally coupled problem of exploration scheduling and few-step reverse-SDE discretization, establishing it as a critical component of the RL policy.
- We analyze the systematic biases of prior samplers. Building on CPS [Wang and Yu, 2025], which showed that Euler-style transitions inject excessive noise at small step counts, we prove that coefficient-preserving (CPS) rules suffer from covariance contraction that biases the marginal distribution.
- We introduce PRECISE, featuring a theoretically principled logSNR-derived exploration schedule and an SDE-consistent closed-form transition. Under a novel "frozen clean-latent posterior mean" approximation, it eliminates excess discretization noise.
- Extensive experiments demonstrate that PRECISE achieves state-of-the-art scores on the optimized rewards (e.g., PickScore, HPSv2.1). Crucially, it matches the best prior baselines while requiring 13.1–53.2% less wall-clock training time, remaining highly robust even in extreme low-budget regimes.

2 Related Work

Diffusion and flow-matching models. Modern visual generators use denoising diffusion and score-SDE formulations [Ho et al., 2020, Song et al., 2020b] or continuous-transport formulations such as flow matching and rectified flow [Lipman et al., 2022, Liu et al., 2022]. These frameworks underpin many current image and video systems [Esser et al., 2024, Black Forest Labs, 2025, Wan et al., 2025, Gao et al., 2025, Cao et al., 2025, Kong et al., 2024, Wu et al., 2025].

RL post-training for diffusion and flow-matching models. RL post-training aligns visual generators to downstream rewards or preferences via methods such as DDPO, DPOK, ImageReward/ReFL, Diffusion-DPO, and D3PO [Black et al., 2023, Fan et al., 2023, Xu et al., 2023, Wallace et al., 2024, Yang et al., 2024]. Recent work extends these ideas to ODE-based generators through Dance-GRPO and Flow-GRPO [Xue et al., 2025, Liu et al., 2025]; follow-ups improve rollout efficiency [Li et al., 2025a,b, Ding and Ye, 2025] and optimization stability [Wang et al., 2025a]. CPS [Wang and Yu, 2025] is closest to ours: it targets the stochastic sampler used by Flow-GRPO and Dance-GRPO, replacing the Euler transition with a coefficient-preserving rule. Our work further improves the sampler while staying agnostic to the RL objective or training wrapper. Another branch avoids reverse-SDE rollout likelihoods, through forward-process optimization [Zheng et al., 2026a], advantage- or reward-weighted matching losses [Xue et al., 2026, Fan et al., 2025], or ELBO-based likelihoods computed from the final sample [Choi et al., 2026]; PRECISE targets the GRPO-style setting, where the sampler defines the policy and its likelihoods.

Few-step sampler design. Another line of work studies accurate few-step discretization. Fixed-dynamics solvers such as DDIM, DEIS, DPM-Solver, and UniPC reduce inference error without retraining [Song et al., 2020a, Zhang and Chen, 2022, Lu et al., 2022, Zhao et al., 2023], while accelerators learn or distill trajectories into very small NFE [Salimans and Ho, 2022, Song et al., 2023, Luo et al., 2023], for example by choosing training couplings that straighten the learned flow [Lin et al., 2026b] or by conditioning on sub-modes to keep one-step generation diverse [Lin et al., 2026a]. PRECISE is closest in spirit to fixed-dynamics samplers, but targets a different regime: in RL post-training, stochasticity is the exploration mechanism rather than a nuisance, so the transition must be both few-step accurate and stochastic inside policy optimization.

3 Preliminaries

3.1 Flow matching and reverse-time sampling

We write a general diffusion or interpolation path as $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \in [0, 1]$ indexes the noise level, and larger t corresponds to a noisier state. In the flow-matching parameterization, $\alpha_t = 1 - t$ and $\sigma_t = t$:

$$\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\boldsymbol{\epsilon}. \tag{1}$$

The target velocity is $\mathbf{u}_t = \boldsymbol{\epsilon} - \mathbf{z}_0$. A flow-matching model learns to predict $\mathbf{u}_t$ from $(\mathbf{z}_t, t)$. At test time, sampling integrates the probability-flow ODE backward from $t = 1$ to $t = 0$.

For later use, define the clean-latent posterior mean as $\hat{\mathbf{z}}_0(t) \triangleq \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t]$. Under the linear path in Eq. (1), the posterior mean of the noise and the score are

$$\hat{\boldsymbol{\epsilon}}(t) = \frac{\mathbf{z}_t - (1 - t)\hat{\mathbf{z}}_0(t)}{t}, \tag{2}$$

$$\nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t) = -\frac{\hat{\boldsymbol{\epsilon}}(t)}{t} = \frac{(1 - t)\hat{\mathbf{z}}_0(t) - \mathbf{z}_t}{t^2}. \tag{3}$$

This follows by differentiating the Gaussian conditional density $p(\mathbf{z}_t \mid \mathbf{z}_0)$ with respect to $\mathbf{z}_t$ and taking the posterior expectation over $\mathbf{z}_0$.

3.2 RL post-training for flow matching

RL post-training requires stochastic rollouts. Flow-GRPO [Liu et al., 2025] and Dance-GRPO [Xue et al., 2025] therefore replace the deterministic reverse-time ODE with the reverse-time SDE

$$\mathrm{d}\mathbf{z}_t = \left(\mathbf{u}_t - \tfrac{1}{2}\varepsilon_t^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)\right) \mathrm{d}t + \varepsilon_t \, \mathrm{d}\mathbf{w}_t, \tag{4}$$

where $\varepsilon_t \geq 0$ controls exploration and $\mathbf{w}_t$ is Brownian motion. In ideal continuous time with exact scores, this SDE preserves the flow-matching marginals for any ε_t. In practice, ε_t and the finite-step discretization of Eq. (4) both matter because training uses imperfect predictions and a small NFE budget.

Given a stochastic sampler, recent work optimizes groups of trajectories $\{\tau_i\}_{i=1}^{G}$ with a GRPO-style clipped policy objective and group-normalized rewards [Shao et al., 2024].

4 Method and Analysis

The stochastic sampler used for RL post-training has two coupled design axes: the exploration schedule ε_t and the finite-step transition rule. The schedule determines how much randomness the policy injects at each noise level, while the transition rule determines whether the resulting reverse SDE is faithfully realized at small training NFE. We first derive a schedule that balances exploration and denoising stability. We then show how existing samplers deviate from the flow-matching process, and derive PRECISE, an SDE-consistent closed-form transition under a local approximation that freezes the clean-latent posterior mean.

4.1 Balancing exploration and stability

We start with the exploration schedule ε_t. As shown in Eq. (4), ε_t controls the strength of the injected noise while also scaling the score term. We choose ε_t to balance exploration and reliability: large enough for diversity, yet stable enough to keep model error controlled along the denoising trajectory.

To formalize this tradeoff, we analyze how much denoising progress is made by the velocity term $\mathbf{u}_t \mathrm{d}t$ and the score term $-\frac{1}{2}\varepsilon_t^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)\mathrm{d}t$, and how much is canceled by the stochastic term $\varepsilon_t \, \mathrm{d}\mathbf{w}_t$. We use log signal-to-noise ratio (logSNR) to measure denoising progress, following prior SNR and noise-level parameterizations [Kingma et al., 2021, Karras et al., 2022]. For $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$, define $\mathrm{SNR}(t) \triangleq \alpha_t^2/\sigma_t^2$ and $\lambda_t \triangleq \log \mathrm{SNR}(t)$; under flow matching, $(\alpha_t, \sigma_t) = (1 - t, t)$, so $\lambda_t = \log((1 - t)^2/t^2)$.

Under this progress measure, we can analyze the first-order contribution of each term to the logSNR change independently. Consider the reverse SDE in Eq. (4). Over an infinitesimal reverse step from t to $t - \Delta t$, there are the velocity term $-\mathbf{u}_t \Delta t$, score term $\frac{1}{2}\varepsilon_t^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)\Delta t$, and stochastic term $\varepsilon_t \sqrt{\Delta t}\, \mathbf{w}$, with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The first-order logSNR contributions of the three terms are as follows.

Lemma 4.1 (First-order logSNR decomposition)**.** *For the flow-matching reverse SDE, as $\Delta t \to 0$,*

$$\Delta\lambda_{\mathrm{vel}} = \frac{2\Delta t}{t(1-t)} + o(\Delta t), \quad \Delta\lambda_{\mathrm{sco}} = \frac{\varepsilon_t^2 \Delta t}{t^2} + o(\Delta t), \quad \Delta\lambda_{\mathrm{sto}} = -\frac{\varepsilon_t^2 \Delta t}{t^2} + o(\Delta t),$$

where the three quantities denote the first-order contributions of the velocity term, score term, and stochastic term, respectively.

Lemma 4.1 shows that the net first-order progress is produced by the velocity term and is independent of ε_t, while the intermediate score and noise cancelation scales like ε_t^2/t^2. The score-term progress is nevertheless not free: before it is canceled by re-randomization, it asks the model to produce extra denoising progress beyond the normal ODE velocity progress. If this extra progress is much larger than the velocity-induced progress, model prediction errors can be amplified. We therefore control this burden by keeping the ratio $R(t) \triangleq \Delta\lambda_{\mathrm{sco}}/\Delta\lambda_{\mathrm{vel}}$ constant in time, with its magnitude set by a scalar parameter, giving the schedule

$$\varepsilon_t = \eta\sqrt{\frac{t}{1-t}}, \tag{5}$$

where $\eta > 0$ sets the magnitude of the extra model-driven progress. Under this schedule, $R(t) = (\varepsilon_t^2/t^2)/(2/(t(1-t))) = \eta^2/2$. Thus η directly controls the score-to-velocity progress ratio; in practice, values around $\sqrt{2}$ work well across tasks, where the score-induced progress is comparable to the normal velocity progress. A constant $R(t)$ keeps this burden at a fixed proportion of the velocity progress near both endpoints and leaves one scalar to tune. Flow-GRPO's schedule and the memoryless schedule of Adjoint Matching share the factor $\sqrt{t/(1-t)}$ and differ only in η: $\eta = 0.7 \approx \sqrt{2}/2$ [Liu et al., 2025] and $\eta = \sqrt{2}$ [Domingo i Enrich et al., 2025].

4.2 Existing samplers deviate from the flow-matching process

After fixing the exploration schedule, and therefore the denoising SDE, the remaining design problem is the finite-step sampler. It should follow the reverse SDE closely enough that reward optimization sees a faithful stochastic policy with minimal discretization artifact.

Euler sampler introduces excess discretization noise. Consider the standard Euler–Maruyama discretization of the reverse SDE used by Flow-GRPO [Liu et al., 2025] and Dance-GRPO [Xue et al., 2025]. One step from t to $t' < t$ is

$$\mathbf{z}_{t'}^{\mathrm{Euler}} = \mathbf{z}_t + \Big(-\mathbf{u}_t + \tfrac{1}{2}\varepsilon_t^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)\Big)\Delta t + \varepsilon_t\sqrt{\Delta t}\,\mathbf{w},$$

with $\Delta t = t - t'$. This discretization freezes the velocity and score at time t. That approximation is mostly harmless for the deterministic ODE, but problematic once new noise is injected: the noise changes both the velocity and score inside the step, while Euler transition ignores that feedback.

CPS [Wang and Yu, 2025] identified this excess noise through a one-step coefficient analysis; a point-mass toy example, for which that analysis is exact, exposes the mismatch. Assume the data distribution is a point mass at $\mathbf{z}_\star$. Solving Eq. (1) for $\boldsymbol{\epsilon}$ and plugging the resulting velocity and score into the Euler step gives

$$\mathbf{z}_{t'}^{\mathrm{Euler}} = (1-t')\mathbf{z}_\star + \left(t' - \frac{\varepsilon_t^2 \Delta t}{2t}\right)\boldsymbol{\epsilon} + \varepsilon_t\sqrt{\Delta t}\,\mathbf{w}.$$

Because $\mathbf{w}$ is independent of $\boldsymbol{\epsilon}$, the noise coefficient becomes $\sqrt{(t' - \varepsilon_t^2\Delta t/(2t))^2 + \varepsilon_t^2\Delta t}$, which is strictly larger than t' whenever $\varepsilon_t > 0$. Thus Euler adds excess noise even with a perfect model and the simplest data law.

CPS preserves coefficients while collapsing residual uncertainty. CPS [Wang and Yu, 2025] proposes transition rules of the form

$$\mathbf{z}_{t'}^{\mathrm{CPS}} = (1-t')\hat{\mathbf{z}}_0(t) + k_1\hat{\boldsymbol{\epsilon}}(t) + k_2\mathbf{w}, \qquad k_1^2 + k_2^2 = t'^2.$$

This coefficient-preserving rule is necessary for exactness under the point-mass model. For non-degenerate data distributions, coefficient preservation leaves the marginal law underdetermined

because posterior means discard residual uncertainty in $\mathbf{z}_0 \mid \mathbf{z}_t$. Specifically, the deterministic part of the CPS transition is the posterior mean of a coefficient-preserved random variable. Total covariance then shows that CPS contracts the covariance whenever that variable still has posterior uncertainty after conditioning on $\mathbf{z}_t$. For a simple local case, take $t' \approx t$ with $k_1 = 0$ and $k_2 = t'$. The transition becomes $\mathbf{z}_{t'}^{\mathrm{CPS}} = (1-t')\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t] + t'\mathbf{w}$, whereas the corresponding flow-matching marginal would use $(1-t')\mathbf{z}_0 + t'\boldsymbol{\epsilon}$. The covariance contraction, defined as the target covariance minus the CPS covariance, is exactly $(1-t')^2\mathbb{E}_{\mathbf{z}_t}[\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t)]$, so any remaining ambiguity about the clean latent is removed by the posterior mean. Appendix C gives the general derivation and a double-ring example where this contraction appears as a persistent marginal bias.

4.3 PRECISE: an SDE-consistent finite-step transition

The finite-step transition should preserve reverse-SDE dynamics while remaining usable at small NFE budgets required by RL. The previous two cases isolate two constraints: Euler freezing ignores the feedback from freshly injected noise, while coefficient preservation alone can bias the sample distribution. We therefore retain the SDE but seek a local anchor robust to injected noise.

The natural choice is the clean-latent posterior mean given the current state, $\hat{\mathbf{z}}_0(t) = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t]$. Since estimating this posterior mean is naturally a denoising task, this anchor generally moves much more slowly than the noisy latent under isotropic perturbations. Unlike CPS, we use it only to anchor the linearized reverse SDE; the residual stochastic variance is still set by the SDE transition.

Using Eqs. (2) and (3), the reverse SDE can be rewritten as

$$\mathrm{d}\mathbf{z}_t = \left(\frac{\mathbf{z}_t - \hat{\mathbf{z}}_0(t)}{t} + \frac{\varepsilon_t^2}{2t^2}\big(\mathbf{z}_t - (1-t)\hat{\mathbf{z}}_0(t)\big) \right) \mathrm{d}t + \varepsilon_t \, \mathrm{d}\mathbf{w}_t. \tag{6}$$

Assumption 4.2 (Frozen posterior mean on one step)**.** For a reverse step from t to $t' < t$, we treat the clean-latent posterior mean along the step as approximately constant, i.e., $\hat{\mathbf{z}}_0(s) \approx \hat{\mathbf{z}}_0(t)$ for all $s \in [t', t]$.

Assumption 4.2 is exact for point-mass data and serves as a local approximation over one step. Under this assumption, the SDE becomes linear and admits the following closed-form transition, which is exact for the frozen-mean SDE; Section 4.4 measures how closely it tracks the original reverse SDE.

Theorem 4.3 (Exact transition under frozen posterior mean)**.** *Assume Assumption 4.2. Then the exact transition from $\mathbf{z}_t$ to $\mathbf{z}_{t'}$ is*

$$\mathbf{z}_{t'} = (1-t')\hat{\mathbf{z}}_0(t) + \frac{t'}{t} e^{-A(t',t)/2}\big(\mathbf{z}_t - (1-t)\hat{\mathbf{z}}_0(t)\big) + t'\sqrt{1 - e^{-A(t',t)}}\,\mathbf{w},$$

where $A(t',t) \triangleq \int_{t'}^{t} \varepsilon_s^2 / s^2 \, \mathrm{d}s$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent of $\mathbf{z}_t$.

The proof is given in Appendix B.4. In practice, we replace the exact $\hat{\mathbf{z}}_0(t)$ by the model clean-latent prediction and evaluate $A(t',t)$ analytically or numerically, depending on ε_t.

For the exploration schedule in Eq. (5), $A(t',t) = \eta^2 \int_{t'}^{t} 1/(s(1-s))\,\mathrm{d}s = \eta^2 \log(t(1-t')/(t'(1-t)))$, so Eq. (2) turns Theorem 4.3 into the explicit transition rule

$$\mathbf{z}_{t'} = (1-t')\hat{\mathbf{z}}_0(t) + t'\rho(t',t)\hat{\boldsymbol{\epsilon}}(t) + t'\sqrt{1-\rho(t',t)^2}\,\mathbf{w},$$

with

$$\rho(t',t) \triangleq \left(\frac{t'(1-t)}{t(1-t')} \right)^{\eta^2/2}, \tag{7}$$

obtaining the final transition rule used by PRECISE.

4.4 Discussion

The main approximation in PRECISE is local: during one finite reverse step, we treat the clean-latent posterior mean as fixed. Appendix B.2 bounds the resulting one-step transition error by the drift of this mean within the step (Proposition B.2); the posterior covariance controls its sensitivity to the state (Lemma B.1).

We measure both quantities directly on SD3.5-M [Esser et al., 2024], against a reference that integrates the reverse SDE with 400 Euler–Maruyama substeps per step; the one-step samplers are driven by

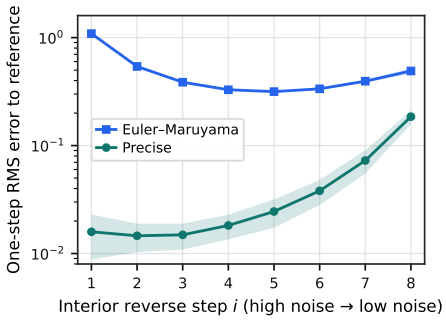

Figure 2: One-step error to a fine-grid reverse-SDE reference on SD3.5-M (10 NFE, $\eta = 1.5$). Euler–Maruyama freezes the velocity and score within a step, while PRECISE freezes the clean-latent prediction; bands show ± 1 std over eight trajectories.

the reference's fine-grid Brownian increments (Appendix B.3). Over the eight interior steps, the clean-latent prediction drifts by 0.21 of the step displacement on average over each step, so freezing it is an approximation rather than an identity. Even so, one PRECISE step lands far closer to the reference than one Euler–Maruyama step on the same SDE, which freezes the velocity and score instead (Fig. 2): the RMS error is 0.048 versus 0.485 on average, and PRECISE is closer on all 64 trajectory–step pairs, although the error ratio rises from 0.015 to 0.38 toward the data end.

The closure is less accurate where the predicted clean latent changes quickly relative to the noisy state: near ambiguous or multimodal states with large posterior covariance (Lemma B.1), where the clean-latent prediction is inaccurate, and for steps large relative to the noise level, such as the last coarse steps, where the prediction moves almost one-for-one with the noisy state.

5 Experiments

We evaluate the sampler induced by Theorem 4.3, abbreviated as PRECISE, against Dance-GRPO [Xue et al., 2025], Flow-GRPO [Liu et al., 2025], and CPS [Wang and Yu, 2025] for GRPO-based reward optimization of flow-matching models.

5.1 Experimental setup

Rewards. For rule-based optimization, we use GenEval [Ghosh et al., 2023]. For model-based rewards, we train with PickScore [Kirstain et al., 2023], CLIPScore [Hessel et al., 2021], and HPSv2.1 [Wu et al., 2023], and report ImageReward [Xu et al., 2023], UnifiedReward [Wang et al., 2025b], and Aesthetics [LAION-AI, 2022] at evaluation. These metrics cover complementary axes of reward alignment, text-image matching, and perceptual quality.

Prompt datasets. GenEval prompts are synthetic object-centric prompts from task templates over COCO categories [Ghosh et al., 2023]. PickScore, CLIPScore, and HPSv2.1 prompts come from Pick-a-Pic training captions [Kirstain et al., 2023], filtered to unique captions with at least six whitespace-separated words and split by holding out 2,048 shuffled captions for evaluation. In all cases, we use the prompt splits distributed with Flow-GRPO codebase [Liu et al., 2025].

Training and evaluation protocol. We base our experiments on the Flow-GRPO codebase [Liu et al., 2025]. Our main backbone is Stable Diffusion 3.5 Medium (SD3.5-M) [Esser et al., 2024], a pretrained text-to-image flow-matching model, at 512×512 resolution. We fine-tune with LoRA ($r = 32$, $\alpha = 64$) and use GRPO-Guard [Wang et al., 2025a]. We use no classifier-free guidance during training or evaluation and no KL loss during training. We use the same NFE for training and evaluation. Unless stated otherwise, results use the final checkpoint and each method's default scalar exploration parameter η: 0.3 for Dance-GRPO [Xue et al., 2025], 0.7 for Flow-GRPO [Liu et al., 2025], 0.7 for CPS [Wang and Yu, 2025], and 1.5 for PRECISE. We keep the rest of the Flow-GRPO recipe and dataset splits fixed, changing only the sampler and exploration setting. All runs use $8\times$ NVIDIA H20 GPUs, taking about 80GB GPU memory per H20.

5.2 Main results

We train all four samplers on SD3.5-M under two protocols: 10-NFE/3000-iteration and 30-NFE/1000-iteration, both taking about 3 days. We report final-checkpoint training rewards and held-out metrics,[5] using mean and standard deviation over three evaluation seeds where available, and plot training curves for the three in-domain rewards. We also include DiffusionNFT [Zheng et al., 2026a] as a reference, using their default 25-NFE training setting, training for 3 days, and evaluating at 30 NFE.[6]

Table 1: Main SD3.5-M results under the 10-NFE/3000-iteration and 30-NFE/1000-iteration protocols. Multi-seed entries are mean±standard deviation over three evaluation seeds. Higher is better for every metric.

	Training rewards			Held-out metrics		
Method	PickScore ↑	CLIPScore ↑	HPSv2.1 ↑	ImageReward ↑	UnifiedReward v2 ↑	Aesthetics ↑
10-NFE protocol						
Dance-GRPO [Xue et al., 2025]	22.428±0.007	0.942±0.000	0.340±0.000	1.350±0.010	0.647±0.000	6.620±0.007
Flow-GRPO [Liu et al., 2025]	23.242±0.003	1.018±0.001	0.377±0.000	**1.627±0.004**	0.652±0.000	6.530±0.003
CPS [Wang and Yu, 2025]	23.670±0.002	1.026±0.001	0.389±0.000	1.607±0.005	0.650±0.000	**6.667±0.003**
PRECISE	**23.745±0.004**	**1.038±0.001**	**0.391±0.000**	1.615±0.004	**0.654±0.000**	6.652±0.003
30-NFE protocol						
Dance-GRPO [Xue et al., 2025]	22.237±0.004	0.868±0.001	0.330±0.000	1.082±0.008	0.652±0.001	6.568±0.005
Flow-GRPO [Liu et al., 2025]	23.227±0.006	0.967±0.001	0.374±0.000	1.528±0.002	0.656±0.000	6.736±0.004
CPS [Wang and Yu, 2025]	23.288±0.007	0.994±0.001	**0.379±0.000**	1.542±0.001	0.652±0.000	6.693±0.002
PRECISE	**23.421±0.006**	**1.002±0.001**	**0.379±0.000**	**1.548±0.004**	**0.658±0.000**	**6.766±0.003**
DiffusionNFT [Zheng et al., 2026a]	23.349	1.010	0.363	1.573	0.660	6.494

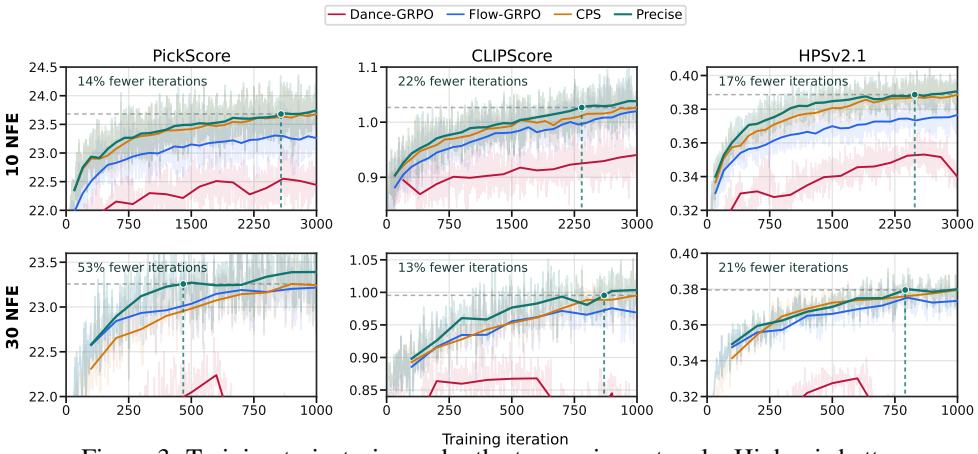

Figure 3: Training trajectories under the two main protocols. Higher is better.

As shown in Table 1 and Figure 3, PRECISE ranks first on all three training rewards and remains competitive on held-out metrics, and the same ordering holds through most of optimization. Using the best value among the prior samplers on each metric as the target, PRECISE reaches the PickScore, CLIPScore, and HPSv2.1 targets with 14.2%–22.1% fewer iterations in the 10-NFE protocol, and 13.1%–53.2% fewer iterations in the 30-NFE protocol, directly translating to wall-clock savings since sampler overhead is negligible.

The gaps among stochastic samplers highlight both sampler-design components: exploration-stability balance and SDE consistency. Dance-GRPO uses a constant noise schedule $\varepsilon_t = \eta$, too conservative in early denoising and too aggressive in late steps, according to our analysis in Section 4. It yields little layout variation and corrupts fine details, causing training collapse. A similar, smaller gap exists between PRECISE and CPS. In contrast, the gap between PRECISE and Flow-GRPO mainly stems from removing excess discretization noise, which boosts sample quality and reward scores.

Since Dance-GRPO underperforms, we exclude it from later experiments for efficiency.

[5]For Dance-GRPO in the 30-NFE protocol, we report iteration 600, the highest checkpoint before training becomes unstable.

[6]We tried to accelerate DiffusionNFT with 10-NFE training, but it is less stable and diverges after two days. The best pre-divergence checkpoint is worse than the default setting.

5.3 Task-level validation

As a task-level check, we isolate one rule-based reward and one model-based reward in the 10-NFE/3000-iteration protocol. Table 2 shows that PRECISE is best on both GenEval and PickScore.

Table 2: Head-to-head results under the 10-NFE protocol. Higher is better.

Method	GenEval ↑	PickScore ↑
Flow-GRPO [Liu et al., 2025]	0.969	24.283
CPS [Wang and Yu, 2025]	<u>0.972</u>	<u>24.594</u>
PRECISE	**0.980**	**24.668**

5.4 Results on FLUX.2

We further evaluate on FLUX.2 Klein 4B Base [Black Forest Labs, 2025, 2026] at 512×512 resolution. We fine-tune with LoRA ($r = 16$, $\alpha = 16$) under a 20-NFE/1000-iteration protocol, and train for about 1 day on $8\times$ NVIDIA H20 GPUs. Table 3 and Figure 4 report final rewards and training curves.

Table 3: FLUX.2 Klein 20-NFE results at 1000 iterations. Entries are mean±standard deviation over three evaluation seeds. Higher is better for every metric.

	Training rewards			Held-out metrics		
Method	PickScore ↑	CLIPScore ↑	HPSv2.1 ↑	ImageReward ↑	UnifiedReward v2 ↑	Aesthetics ↑
Flow-GRPO [Liu et al., 2025]	<u>22.591±0.006</u>	<u>0.905±0.001</u>	0.355±0.000	1.266±0.001	**0.654±0.000**	<u>6.676±0.005</u>
CPS [Wang and Yu, 2025]	22.486±0.005	0.898±0.001	<u>0.362±0.000</u>	<u>1.334±0.010</u>	0.649±0.001	6.513±0.000
PRECISE	**22.791±0.010**	**0.930±0.001**	**0.366±0.000**	**1.410±0.006**	<u>0.653±0.000</u>	**6.677±0.007**

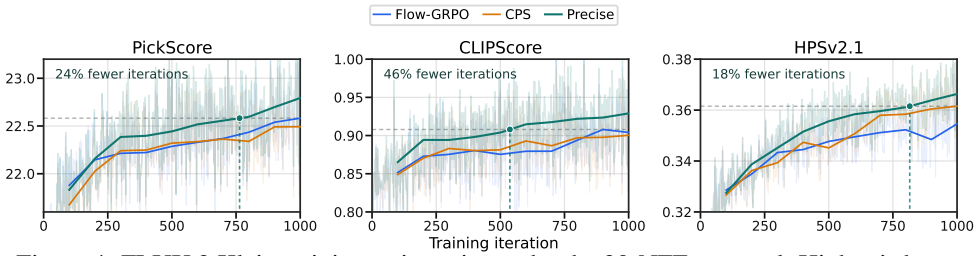

Figure 4: FLUX.2 Klein training trajectories under the 20-NFE protocol. Higher is better.

Table 3 shows that gains transfer from SD3.5-M to the newer FLUX.2 backbone. At the final checkpoint, PRECISE ranks first on five of six metrics and remains within 0.001 of the best sampler on UnifiedReward v2. Figure 4 shows the same trend: PRECISE rises faster and finishes higher on the three training rewards. Using the best value among the prior samplers as the target, PRECISE reaches the PickScore, CLIPScore, and HPSv2.1 targets with 18.4%–46.3% fewer iterations.

5.5 Ablations

Exploration schedule and discretization. We isolate the two sampler-design components with a progression from Dance-GRPO [Xue et al., 2025] to PRECISE. First, we replace the Euler-style transition in Dance-GRPO with our SDE-consistent finite-step transition while keeping the constant exploration schedule. Second, we replace the constant schedule with the logSNR-derived schedule in Eq. (5), giving PRECISE. We set $\eta = 0.7$ for the intermediate sampler, about a $2\times$ increase over default Dance-GRPO, enabled by the improved stability from our discretization.

Table 4: Ablation on exploration schedule and discretization under the SD3.5-M 10-NFE protocol at 3000 iterations. The first row reports Dance-GRPO raw scores; later rows report absolute changes over the previous row. Higher is better for every metric.

	Training rewards			Held-out metrics		
Ablation	PickScore ↑	CLIPScore ↑	HPSv2.1 ↑	ImageReward ↑	UnifiedReward v2 ↑	Aesthetics ↑
Dance-GRPO [Xue et al., 2025]	22.428	0.942	0.340	1.350	0.647	6.620
+ SDE-consistent discretization	+0.826	+0.043	+0.037	+0.151	+0.003	+0.008
+ logSNR exploration schedule	+0.491	+0.053	+0.014	+0.114	+0.004	+0.024

Table 4 shows that replacing Dance-GRPO's Euler transition with the SDE-consistent finite-step transition improves all reported metrics, and replacing the constant schedule with the logSNR-derived schedule gives another consistent gain. Thus, both the finite-step transition and exploration schedule contribute to reward optimization.

NFE and exploration strength. We additionally test whether PRECISE is robust to NFE and exploration strength.

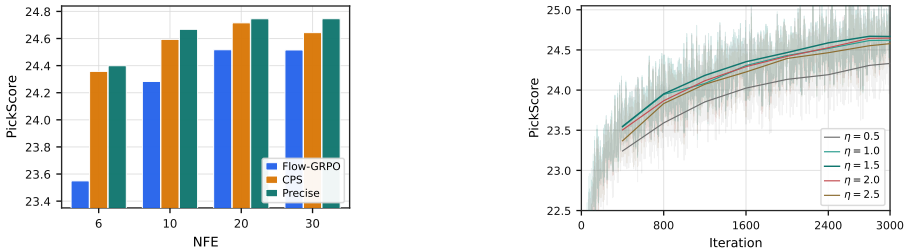

Figure 5: Ablations on NFE and exploration strength. Left: NFE ablation on PickScore. Right: η ablation on PickScore. Higher is better.

The NFE ablation trains each sampler for 3000 iterations at $N \in \{6, 10, 20, 30\}$ on PickScore. In the left panel of Figure 5, PRECISE leads at every tested NFE, with the clearest margin in the aggressive 6-NFE regime. Flow-GRPO [Liu et al., 2025] degrades sharply at low NFE because of excess noise.

To test robustness to the exploration strength η, we sweep $\eta \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ for PRECISE in the 10-NFE PickScore setting. As shown in the right panel of Figure 5, performance is stable across $\eta \in \{1.0, 1.5, 2.0\}$ and degrades gracefully at the more extreme values, due to lack of exploration or denoising instability. Thus, PRECISE is robust across a practical range of NFE budgets and exploration strengths.

6 Conclusion

We presented PRECISE, a stochastic sampler for online RL post-training of flow-matching models. PRECISE treats the sampler as part of the policy and addresses its two design choices: how to balance exploration and stability, and how to discretize the resulting SDE at a small NFE budget. It combines a logSNR-based exploration schedule with an SDE-consistent finite-step transition derived under a new frozen posterior-mean approximation. Extensive experiments demonstrate that PRECISE consistently improves reward optimization over prior samplers, both in terms of final reward and training time.

Limitations. The frozen posterior-mean approximation is local and less accurate where the clean-latent prediction changes quickly within a step. Each training configuration is run with a single seed, and our evaluation relies on automatic reward models rather than a human preference study.

Acknowledgments and Disclosure of Funding

This work was supported by Tencent Hunyuan, which also provided the computing resources. Weijie Kong, Yue Wu, Qi Tian, Jiangfeng Xiong, Jianwei Zhang, Liefeng Bo, and Zhao Zhong are employees of Tencent; Bo Peng, Tao Huang, and Junzhe Li carried out this work as interns at Tencent Hunyuan. The authors declare no other competing interests.

References

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Black Forest Labs. FLUX.2: Analyzing and enhancing the latent space of FLUX – representation comparison. `https://bfl.ai/research/representation-comparison`, 2025. Technical report.

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. `https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence`, 2026. Release article.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. HunyuanImage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Jaemoo Choi, Yuchen Zhu, Wei Guo, Petr Molodyk, Bo Yuan, Jinbin Bai, Yi Xin, Molei Tao, and Yongxin Chen. Rethinking the design space of reinforcement learning for diffusion models: On the importance of likelihood estimation beyond loss design. *arXiv preprint arXiv:2602.04663*, 2026.

Zheng Ding and Weirui Ye. TreeGRPO: Tree-advantage GRPO for online RL post-training of diffusion models. *arXiv preprint arXiv:2512.08153*, 2025.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025.

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 *International Conference on Machine Learning*, 2024.

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 *International Conference on Learning Representations*, 2025.

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.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

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.

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*, pages 7514–7528, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in Neural Information Processing Systems*, 34:21696–21707, 2021.

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.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

LAION-AI. LAION-Aesthetics Predictor V1. `https://github.com/LAION-AI/aesthetic-predictor`, 2022.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE. *arXiv preprint arXiv:2507.21802*, 2025a.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and efficient GRPO with structured branching in diffusion models. *arXiv preprint arXiv:2509.06040*, 2025b.

Yexiong Lin, Jia Shi, Shanshan Ye, Wanyu Wang, Yu Yao, and Tongliang Liu. SubFlow: Sub-mode conditioned flow matching for diverse one-step generation. *arXiv preprint arXiv:2604.12273*, 2026a.

Yexiong Lin, Yu Yao, Yang Zhou, and Tongliang Liu. Beyond optimal transport: Model-aligned coupling for flow matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pages 3955–3964, 2026b.

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. *arXiv preprint arXiv:2505.05470*, 2025.

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

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*, 2023.

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*, pages 8748–8763. PMLR, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 2023.

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*, pages 8228–8238, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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

Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. GRPO-Guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint arXiv:2510.22319*, 2025a.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025b.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. HunyuanVideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

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

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

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *International Conference on Machine Learning*, 2026.

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*, 2025.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024.

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

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

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, 2026a.

Mingzhe Zheng, Weijie Kong, Yue Wu, Dengyang Jiang, Yue Ma, Xuanhua He, Bin Lin, Kaixiong Gong, Zhao Zhong, Liefeng Bo, Qifeng Chen, and Harry Yang. Manifold-aware exploration for reinforcement learning in video generation. *arXiv preprint arXiv:2603.21872*, 2026b.

A Additional Qualitative Comparisons

Figures 6 and 7 qualitatively compare images sampled by Flow-GRPO, CPS, and PRECISE under the two SD3.5-M protocols in Section 5.2. In the 10-NFE setting, PRECISE more reliably preserves global composition, local texture, and prompt-specific attributes than Flow-GRPO and CPS, while Flow-GRPO samples still exhibit noise artifacts after training. Under 30-NFE training and evaluation, similar qualitative differences remain visible on identity, attribute binding, and scene structure.

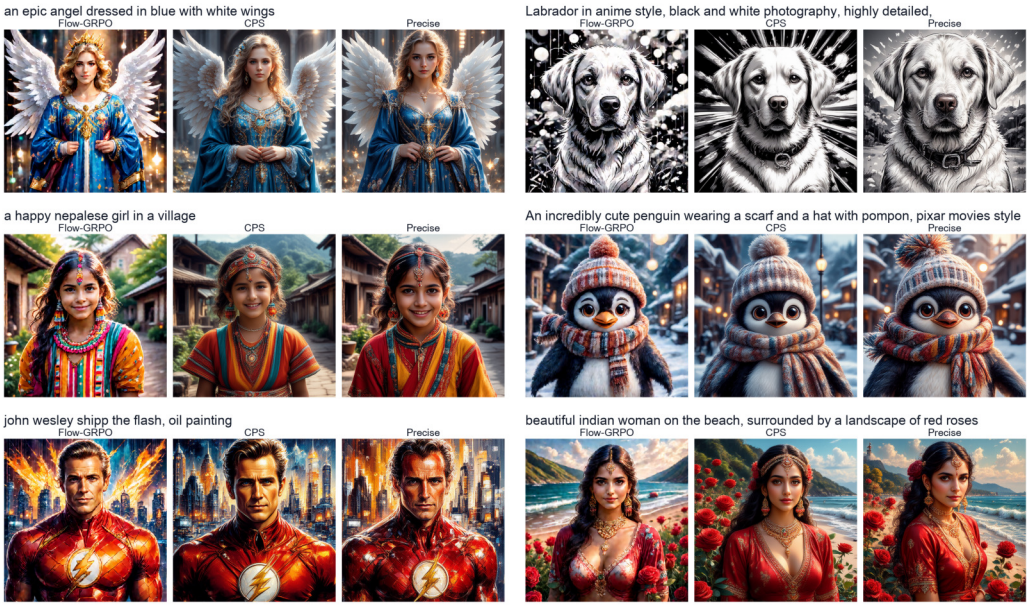

Figure 6: Qualitative examples under the 10-NFE protocol. PRECISE is the most consistent on global structure and attribute binding.

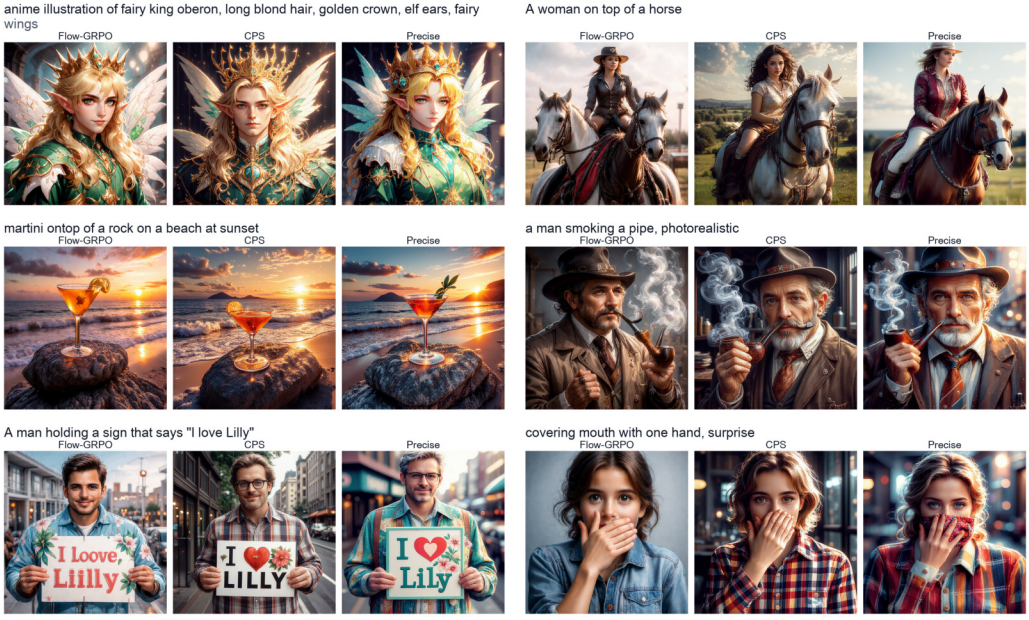

Figure 7: Qualitative examples under the 30-NFE protocol. PRECISE remains the most stable on identity and scene layout.

B Proofs and Extensions for the Sampler Analysis

This appendix contains the formal derivations behind the sampler analysis in the main text. The key point is local: posterior covariance controls how much the clean-latent posterior mean moves, and once this mean is frozen, the reverse SDE has an exact Gaussian transition. Lemma 4.1 gives the first-order logSNR balance among the velocity term, score term, and stochastic term. Lemma B.1 and Proposition B.2 bound the local error introduced by freezing the posterior mean, Appendix B.3 measures that error directly on SD3.5-M, and Theorem 4.3 gives the exact finite-step transition of the frozen-posterior-mean SDE.

B.1 Proof of Lemma 4.1

Proof. Over a reverse step from t to $t - \Delta t$, write the one-step change as the sum of three parts:

$$\begin{aligned} \text{(velocity term)} \quad & -\mathbf{u}_t \Delta t, \\ \text{(score term)} \quad & \frac{1}{2}\varepsilon_t^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t) \Delta t, \\ \text{(stochastic term)} \quad & \varepsilon_t \sqrt{\Delta t}\, \mathbf{w}, \qquad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned}$$

The velocity term takes

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\boldsymbol{\epsilon}$$

to

$$\mathbf{z}_{t-\Delta t} = (1-(t-\Delta t))\mathbf{z}_0 + (t-\Delta t)\boldsymbol{\epsilon},$$

so its logSNR increment is

$$\begin{aligned} \Delta\lambda_{\text{vel}} &= \log \frac{(1-t+\Delta t)^2}{(t-\Delta t)^2} - \log \frac{(1-t)^2}{t^2} \\ &= \frac{2}{t(1-t)} \Delta t + o(\Delta t). \end{aligned}$$

Next isolate the stochastic term while keeping the time index fixed at t. This replaces the noise coefficient t by

$$\sqrt{t^2 + \varepsilon_t^2 \Delta t},$$

hence

$$\begin{aligned} \Delta\lambda_{\text{sto}} &= \log \frac{t^2}{t^2 + \varepsilon_t^2 \Delta t} \\ &= -\frac{\varepsilon_t^2}{t^2} \Delta t + o(\Delta t). \end{aligned}$$

To isolate the score contribution, we freeze time at t and consider the score drift together with its paired stochastic diffusion. With respect to the frozen marginal p_t, this pair forms a Langevin operator whose Fokker–Planck adjoint annihilates p_t. Hence the combined score-noise substep has no first-order effect on the marginal. Consequently, the first-order score contribution must cancel the isolated stochastic perturbation:

$$\Delta\lambda_{\text{sco}} = -\Delta\lambda_{\text{sto}} = \frac{\varepsilon_t^2}{t^2} \Delta t + o(\Delta t).$$

Together with the identity for the velocity term above, this proves the three first-order contributions claimed in Lemma 4.1. □

B.2 Local stability of the frozen posterior-mean approximation

The frozen posterior-mean approximation is controlled when the clean-latent posterior mean changes slowly under local state and time perturbations. Let $\mathbf{m}_t(\mathbf{z}) \triangleq \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t = \mathbf{z}]$ denote the clean-latent posterior mean as a function of the current latent.

Lemma B.1 (Posterior-covariance control of the clean-latent posterior mean)**.** *For the flow-matching path $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, assume regularity conditions that justify differentiating the posterior mean with respect to the conditioning variable, such as a regular conditional density, local domination, and finite second moments. Then*

$$\nabla_{\mathbf{z}} \mathbf{m}_t(\mathbf{z}) = \frac{1-t}{t^2} \operatorname{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t = \mathbf{z}). \tag{8}$$

Consequently, on any convex local region where

$$\lambda_{\max}(\operatorname{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t = \mathbf{z})) \le L_z \frac{t^2}{1-t},$$

this mean map is locally L_z-Lipschitz in state.

Lemma B.1 recovers the point-mass case as the zero-covariance limit, where this mean is exactly constant. When the posterior covariance is small but nonzero, it gives a local sensitivity bound that can be converted into a one-step transition-law bound.

Proposition B.2 (One-step transition error under local regularity)**.** *Consider one reverse step from t to $t' = t - \Delta$ on a compact time interval away from the endpoints. Let $\mathcal{K}_{t,t'}(\cdot \mid \mathbf{z}_t)$ denote the conditional transition law of the original reverse SDE in Eq.* (6)*, and let $\widetilde{\mathcal{K}}_{t,t'}(\cdot \mid \mathbf{z}_t)$ denote the conditional transition law of the frozen posterior-mean SDE obtained by replacing $\mathbf{m}_s(\mathbf{z}_s)$ with $\mathbf{m}_t(\mathbf{z}_t)$ throughout the step. Suppose the two processes are synchronously coupled and remain in a local tube where the drift is C_z-Lipschitz in the state and C_m-Lipschitz in the clean-latent posterior mean. If*

$$\mathbb{E}\,\|\mathbf{m}_s(\mathbf{Z}_s) - \mathbf{m}_t(\mathbf{z}_t)\| \le \omega(s), \qquad s \in [t', t],$$

then the transition laws satisfy

$$W_1\Big(\mathcal{K}_{t,t'}(\cdot \mid \mathbf{z}_t), \widetilde{\mathcal{K}}_{t,t'}(\cdot \mid \mathbf{z}_t)\Big) \le C_m e^{C_z \Delta} \int_{t'}^{t} \omega(s)\,\mathrm{d}s. \tag{9}$$

In particular, if the posterior mean is L_s-Lipschitz in time and L_z-Lipschitz in state along the step, and if $\mathbb{E}\|\mathbf{Z}_s - \mathbf{z}_t\| \le B(t-s) + \bar{\varepsilon}\sqrt{t-s}$, then the right-hand side is at most

$$C_m e^{C_z \Delta} \left(\frac{L_s + L_z B}{2} \Delta^2 + \frac{2 L_z \bar{\varepsilon}}{3} \Delta^{3/2} \right).$$

Thus the approximation is accurate on steps where the posterior mean is locally stable in time and state.

Proof of Lemma B.1. Write $a_t = 1 - t$. For a fixed $t \in (0, 1)$, the posterior density of $\mathbf{z}_0$ given $\mathbf{z}_t = \mathbf{z}$ is proportional to

$$p_0(\mathbf{x}) \exp\left(-\frac{\|\mathbf{z} - a_t \mathbf{x}\|^2}{2t^2} \right).$$

Let $q_t(\mathbf{x} \mid \mathbf{z})$ denote this posterior density. Its score with respect to the conditioning variable is

$$\nabla_{\mathbf{z}} \log q_t(\mathbf{x} \mid \mathbf{z}) = \frac{a_t \mathbf{x} - \mathbf{z}}{t^2} - \nabla_{\mathbf{z}} \log p_t(\mathbf{z}).$$

Differentiating the posterior mean under the integral and using the standard identity $\nabla_{\mathbf{z}} \mathbb{E}_{q_t}[f] = \operatorname{Cov}_{q_t}(f, \nabla_{\mathbf{z}} \log q_t)$ gives

$$\begin{aligned} \nabla_{\mathbf{z}} \mathbf{m}_t(\mathbf{z}) &= \operatorname{Cov}\left(\mathbf{z}_0, \frac{a_t \mathbf{z}_0 - \mathbf{z}}{t^2} - \nabla_{\mathbf{z}} \log p_t(\mathbf{z}) \;\middle|\; \mathbf{z}_t = \mathbf{z} \right) \\ &= \frac{a_t}{t^2} \operatorname{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t = \mathbf{z}), \end{aligned}$$

because the terms depending only on $\mathbf{z}$ have zero posterior covariance with $\mathbf{z}_0$. Since $a_t = 1 - t$, this proves Eq. (8). The Lipschitz statement follows by bounding the operator norm of the Jacobian and applying the mean-value theorem on the region. □

Proof of Proposition B.2. Let $\widetilde{\mathbf{Z}}_s$ denote the frozen posterior-mean process coupled to $\mathbf{Z}_s$ with the same Brownian path and the same initial condition at time t. The Brownian terms cancel in the synchronous coupling. By the local Lipschitz assumptions, the coupled difference $\boldsymbol{\delta}_s \triangleq \mathbf{Z}_s - \widetilde{\mathbf{Z}}_s$ satisfies the integral inequality

$$\mathbb{E}\|\boldsymbol{\delta}_r\| \le \int_r^t C_z \mathbb{E}\|\boldsymbol{\delta}_s\| \,\mathrm{d}s + C_m \int_r^t \mathbb{E}\,\|\mathbf{m}_s(\mathbf{Z}_s) - \mathbf{m}_t(\mathbf{z}_t)\| \,\mathrm{d}s, \qquad r \in [t', t],$$

Gronwall's inequality and the definition of ω give

$$\mathbb{E}\|\boldsymbol{\delta}_{t'}\| \le C_m e^{C_z \Delta} \int_{t'}^t \omega(s) \,\mathrm{d}s.$$

This endpoint coupling upper-bounds W_1, proving Eq. (9).

For the explicit Lipschitz corollary, decompose for $s \in [t', t]$

$$\begin{aligned}\|\mathbf{m}_s(\mathbf{Z}_s) - \mathbf{m}_t(\mathbf{z}_t)\| &\le \|\mathbf{m}_s(\mathbf{Z}_s) - \mathbf{m}_t(\mathbf{Z}_s)\| + \|\mathbf{m}_t(\mathbf{Z}_s) - \mathbf{m}_t(\mathbf{z}_t)\| \\ &\le L_s(t-s) + L_z\|\mathbf{Z}_s - \mathbf{z}_t\|.\end{aligned}$$

Combining this with the assumed path-increment bound and integrating over the step gives the stated explicit rate. □

B.3 Direct diagnostics of the frozen posterior-mean closure

Protocol. We use SD3.5-M [Esser et al., 2024] at the 10-NFE anchors of the training scheduler with $\eta = 1.5$ and no classifier-free guidance, on eight held-out PickScore prompts with one noise realization each. The reference path $\mathbf{Z}^{\mathrm{ref}}$ integrates the reverse SDE in Eq. (4) with 400 Euler–Maruyama substeps per coarse step, re-evaluating the model at every substep. At each coarse anchor t_i, the one-step branches start from the reference state $\mathbf{Z}^{\mathrm{ref}}(t_i)$ and are driven by the reference's fine-grid Gaussian innovations over the step: one-step Euler–Maruyama, on the same SDE with $\eta = 1.5$, freezes the velocity, score, and ε_{t_i} and uses the sum of the increments; one-step PRECISE freezes the anchor clean-latent prediction and uses the weighted sum of the increments implied by its exact transition. This is a shared discrete-innovation coupling rather than a continuous Brownian-path coupling. Because Euler–Maruyama is evaluated at $\eta = 1.5$, its error here is not that of Flow-GRPO at its default η. All distances are per-coordinate RMS. Normalized quantities are formed within each trajectory and reported as mean $\pm$ sample standard deviation over trajectories. We summarize the eight interior steps $i = 1, \ldots, 8$, excluding the entrance step at $t = 1$, where ε_t diverges, and the terminal step. In the last row of Table 5, we first average the eight steps within each trajectory and then aggregate across trajectories.

Within-step drift. Let $\hat{\mathbf{z}}_0(\mathbf{z}, s)$ denote the model clean-latent prediction and $\Delta_i = t_i - t_{i+1}$. Along reference trajectory n, define

$$d_{n,i}(s) = \mathrm{RMS}\big[\hat{\mathbf{z}}_0(\mathbf{Z}_n^{\mathrm{ref}}(s), s) - \hat{\mathbf{z}}_0(\mathbf{Z}_n^{\mathrm{ref}}(t_i), t_i)\big], \qquad A_{n,i} = \frac{1}{\Delta_i}\int_{t_{i+1}}^{t_i} d_{n,i}(s) \,\mathrm{d}s,$$

an empirical counterpart of the drift $\omega(s)$ in Proposition B.2. We estimate $A_{n,i}$ by trapezoidal integration over the substeps and divide it by the reference coarse-step displacement $R_{n,i} = \mathrm{RMS}[\mathbf{Z}_n^{\mathrm{ref}}(t_i) - \mathbf{Z}_n^{\mathrm{ref}}(t_{i+1})]$.

One-step error. For branch b, let $E_{b,i} = \mathrm{RMS}[\mathbf{Z}_b(t_{i+1}) - \mathbf{Z}^{\mathrm{ref}}(t_{i+1})]$. A PRECISE/Euler error ratio below one favors PRECISE.

Results. Table 5 reports both diagnostics. Averaged over a step, the clean-latent prediction drifts by 16–33% of the coarse-step displacement, and the drift at the end of a step is larger, so Assumption 4.2 is an approximation at 10 NFE rather than an identity. Nevertheless, PRECISE is closer to the reference than Euler–Maruyama on all 64 trajectory–step pairs, and the error ratio is below one for every trajectory, with a trajectory-level mean of 0.11. Both the drift and the error ratio are largest at the last interior step, which is large relative to its noise level.

Table 5: Direct diagnostics of the frozen posterior-mean closure on SD3.5-M (10-NFE anchors, $\eta = 1.5$, eight trajectories). Entries are mean±standard deviation over trajectories.

Step i	Drift A_i/R_i	PRECISE error E_i	Euler error E_i	PRECISE/Euler ratio
1	0.2240±0.0932	0.0159±0.0071	1.0895±0.0038	0.0146±0.0065
2	0.1801±0.0518	0.0146±0.0043	0.5407±0.0013	0.0270±0.0080
3	0.1595±0.0415	0.0149±0.0040	0.3864±0.0015	0.0386±0.0103
4	0.1603±0.0416	0.0182±0.0047	0.3290±0.0019	0.0554±0.0144
5	0.1677±0.0490	0.0245±0.0072	0.3161±0.0027	0.0775±0.0233
6	0.1901±0.0514	0.0381±0.0100	0.3355±0.0059	0.1140±0.0316
7	0.2387±0.0603	0.0727±0.0177	0.3947±0.0122	0.1856±0.0505
8	0.3259±0.0659	0.1852±0.0272	0.4909±0.0275	0.3809±0.0765
Mean over i	0.2058±0.0430	0.0480±0.0082	0.4854±0.0065	0.1117±0.0240

B.4 Proof of Theorem 4.3

Proof. We prove the exact transition of the frozen-posterior-mean SDE. Fix a step from t to $t' < t$ and write

$$\mathbf{m} \triangleq \mathbf{m}_t(\mathbf{z}_t).$$

Under Assumption 4.2, the state-dependent value $\mathbf{m}_s(\mathbf{z}_s)$ is replaced by this fixed value throughout the interval $[t', t]$. Equation (6) then reduces to the linear SDE

$$\mathrm{d}\mathbf{z}_s = \left(\frac{\mathbf{z}_s - \mathbf{m}}{s} + \frac{\varepsilon_s^2}{2s^2}\big(\mathbf{z}_s - (1-s)\mathbf{m}\big) \right) \mathrm{d}s + \varepsilon_s \, \mathrm{d}\mathbf{w}_s. \tag{10}$$

Introduce the descending-time parameter

$$\tau \in [0, t - t'], \qquad s(\tau) \triangleq t - \tau,$$

and define the normalized residual

$$\mathbf{r}_\tau \triangleq \frac{\mathbf{z}_{s(\tau)} - (1 - s(\tau))\mathbf{m}}{s(\tau)}.$$

Because $s(\tau)$ decreases with τ, we have $\mathrm{d}s = -\mathrm{d}\tau$. Using

$$\mathbf{z}_{s(\tau)} = (1 - s(\tau))\mathbf{m} + s(\tau)\mathbf{r}_\tau$$

and substituting into Eq. (10), a direct calculation gives

$$\mathrm{d}\mathbf{r}_\tau = -\frac{\varepsilon_{s(\tau)}^2}{2s(\tau)^2}\mathbf{r}_\tau \, \mathrm{d}\tau + \frac{\varepsilon_{s(\tau)}}{s(\tau)} \, \mathrm{d}\mathbf{B}_\tau, \tag{11}$$

where $\mathbf{B}_\tau$ denotes the Brownian driver in descending time.

Equation (11) is a linear SDE. Its solution is

$$\mathbf{r}_\tau = \exp\left(-\frac{1}{2} \int_0^\tau \frac{\varepsilon_{s(u)}^2}{s(u)^2} \, \mathrm{d}u \right) \mathbf{r}_0 + \int_0^\tau \exp\left(-\frac{1}{2} \int_u^\tau \frac{\varepsilon_{s(v)}^2}{s(v)^2} \, \mathrm{d}v \right) \frac{\varepsilon_{s(u)}}{s(u)} \, \mathrm{d}\mathbf{B}_u.$$

The stochastic integral is Gaussian with mean zero. By Itô isometry, its covariance is

$$\int_0^\tau \exp\left(-\int_u^\tau \frac{\varepsilon_{s(v)}^2}{s(v)^2} \, \mathrm{d}v \right) \frac{\varepsilon_{s(u)}^2}{s(u)^2} \, \mathrm{d}u \mathbf{I} = \left(1 - \exp\left(-\int_0^\tau \frac{\varepsilon_{s(u)}^2}{s(u)^2} \, \mathrm{d}u \right) \right) \mathbf{I}.$$

Hence

$$\mathbf{r}_\tau = e^{-A(s(\tau),t)/2}\mathbf{r}_0 + \sqrt{1 - e^{-A(s(\tau),t)}} \, \mathbf{w},$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent of $\mathbf{r}_0$, and

$$A(s(\tau), t) = \int_{s(\tau)}^t \frac{\varepsilon_u^2}{u^2} \, \mathrm{d}u.$$

Now

$$\mathbf{r}_0 = \frac{\mathbf{z}_t - (1-t)\mathbf{m}}{t}.$$

Evaluating the solution at $\tau = t - t'$ so that $s(\tau) = t'$ and then transforming back to $\mathbf{z}_{t'} = (1-t')\mathbf{m} + t'\mathbf{r}_{t-t'}$ gives

$$\mathbf{z}_{t'} = (1-t')\mathbf{m} + \frac{t'}{t} e^{-A(t',t)/2}\big(\mathbf{z}_t - (1-t)\mathbf{m}\big) + t'\sqrt{1 - e^{-A(t',t)}}\,\mathbf{w},$$

which is exactly the claimed formula. □

C Why CPS does not guarantee consistency

This appendix expands the CPS covariance argument in Section 4. With perfect prediction, Flow-GRPO [Liu et al., 2025] is Euler-Maruyama applied to the oracle reverse SDE, and PRECISE has the same first-order expansion, so both share the correct continuous-time limit as the step size goes to zero. CPS [Wang and Yu, 2025] addresses the point-mass failure mode of Euler-style transitions by preserving the nominal signal and noise coefficients, but the same rule can collapse residual posterior uncertainty for non-degenerate data distributions.

Posterior means discard residual uncertainty. Consider the flow-matching forward process from Eq. (1),

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\boldsymbol{\epsilon},$$

and let

$$\hat{\mathbf{z}}_0(t) \triangleq \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t].$$

The posterior mean of the noise is

$$\hat{\boldsymbol{\epsilon}}(t) = \mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{z}_t] = \frac{\mathbf{z}_t - (1-t)\hat{\mathbf{z}}_0(t)}{t}.$$

CPS [Wang and Yu, 2025] applies a coefficient-preserving transition

$$\mathbf{z}_{t'}^{\mathrm{CPS}} = (1-t')\hat{\mathbf{z}}_0(t) + k_1\hat{\boldsymbol{\epsilon}}(t) + k_2\mathbf{w}, \qquad k_1^2 + k_2^2 = t'^2,$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent of $(\mathbf{z}_0, \boldsymbol{\epsilon})$. Define

$$\mathbf{L} \triangleq (1-t')\mathbf{z}_0 + k_1\boldsymbol{\epsilon}.$$

Then the deterministic component of the CPS transition is the posterior mean of the coefficient-preserved random variable $\mathbf{L}$:

$$(1-t')\hat{\mathbf{z}}_0(t) + k_1\hat{\boldsymbol{\epsilon}}(t) = \mathbb{E}[\mathbf{L} \mid \mathbf{z}_t].$$

By the law of total covariance,

$$\mathrm{Cov}(\mathbf{z}_{t'}^{\mathrm{CPS}}) = \mathrm{Cov}(\mathbb{E}[\mathbf{L} \mid \mathbf{z}_t]) + k_2^2\mathbf{I} = \mathrm{Cov}(\mathbf{L}) - \mathbb{E}_{\mathbf{z}_t}\left[\mathrm{Cov}(\mathbf{L} \mid \mathbf{z}_t)\right] + k_2^2\mathbf{I}.$$

Since $\mathbf{z}_0$ and $\boldsymbol{\epsilon}$ are independent,

$$\mathrm{Cov}(\mathbf{L}) + k_2^2\mathbf{I} = (1-t')^2\,\mathrm{Cov}(\mathbf{z}_0) + (k_1^2 + k_2^2)\mathbf{I} = (1-t')^2\,\mathrm{Cov}(\mathbf{z}_0) + t'^2\mathbf{I}.$$

Thus CPS has covariance

$$\mathrm{Cov}(\mathbf{z}_{t'}^{\mathrm{CPS}}) \preceq (1-t')^2\,\mathrm{Cov}(\mathbf{z}_0) + t'^2\mathbf{I},$$

with strict trace contraction whenever $\mathbb{E}_{\mathbf{z}_t}\left[\mathrm{tr}\,\mathrm{Cov}(\mathbf{L} \mid \mathbf{z}_t)\right] > 0$. Thus coefficient preservation fixes the point-mass case, but it does not by itself preserve the full marginal law. Whenever $\mathbf{z}_t$ does not determine the coefficient-preserved random variable $\mathbf{L}$, replacing $\mathbf{L}$ by its posterior mean removes the conditional covariance that should remain in the next-step marginal.

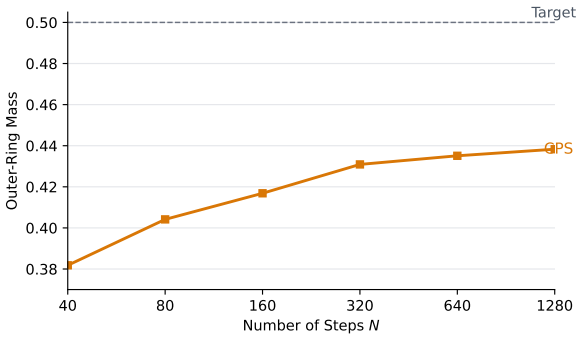

Figure 8: CPS outer-ring mass on the equal-mass double-ring distribution. The target value is 0.5, but CPS remains biased toward the inner ring as NFE increases.

Double-ring example. The double-ring example in Figure 1 visualizes this contraction as an inward bias. We use an equal-mass 2D distribution supported on rings of radii 0.5 and 1.0, and approximate the oracle clean-latent posterior mean $\hat{\mathbf{z}}_0(t) = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_t]$ with a dense discrete support on both rings. We use $\eta = 0.7$ for Flow-GRPO [Liu et al., 2025] and CPS [Wang and Yu, 2025], and $\eta = 1.5$ for PRECISE, matching the main experimental protocol. As shown in Figure 1, CPS samples remain biased toward the inner ring even at $N = 80$.

Figure 8 isolates the large-N regime by tracking the CPS outer-ring mass as the NFE increases to $N = 1280$. The target outer-ring mass is 0.5, but the curve does not approach that value as N increases, demonstrating that the coefficient-preserving rule does not guarantee convergence to the correct distribution.

D External Assets, Licenses, and Terms of Use

We use existing code, prompt datasets, pretrained backbones, reward models, and evaluation metrics. Table 6 lists the original sources and licenses. We use these assets for research evaluation and do not redistribute third-party model weights or datasets beyond their stated terms.

Following the Flow-GRPO codebase [Liu et al., 2025], reported CLIPScore values are the raw image-text similarity multiplied by $10/3$. This scale does not affect training dynamics since rewards are normalized to compute advantages.

Table 6: External assets used in this work.

Asset	Source	License
Flow-GRPO codebase	Official Flow-GRPO repository [Liu et al., 2025]	MIT
Stable Diffusion 3.5 Medium	Stability AI SD3.5-M checkpoint [Esser et al., 2024]	Stability AI Community License
FLUX.2 Klein 4B Base	Black Forest Labs FLUX.2 Klein 4B Base [Black Forest Labs, 2025, 2026]	Apache-2.0
GenEval	GenEval benchmark [Ghosh et al., 2023]	MIT
PickScore	PickScore benchmark and implementation [Kirstain et al., 2023]	MIT
CLIPScore / CLIP	CLIPScore with OpenAI CLIP (`clip-vit-large-patch14`) [Hessel et al., 2021, Radford et al., 2021]	MIT
HPSv2.1	HPSv2.1 checkpoint and code [Wu et al., 2023]	Apache-2.0
ImageReward	ImageReward package and checkpoint [Xu et al., 2023]	Apache-2.0
UnifiedReward v2	UnifiedReward checkpoint [Wang et al., 2025b]	MIT
Aesthetic predictor	LAION-Aesthetics Predictor V1 [LAION-AI, 2022]	MIT

E Broader Societal Impacts

This work can make visual generation systems more controllable and efficient to adapt, supporting creative, accessibility, and educational uses. However, stronger post-training can also increase misuse risks, amplify reward-model bias, reduce diversity, or weaken safeguards, so deployment should include audits, safety filters, provenance mechanisms where appropriate, and evaluations for bias, diversity, memorization, and misuse-sensitive content.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Introduction state the sampler-design problem, the PRECISE transition, and the empirical claims. These claims are supported by the method analysis in Section 4, the main SD3.5-M results in Table 1 and Figure 3, and the FLUX.2 transfer results in Section 5.4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the key approximation behind the method, namely the frozen clean-latent posterior mean assumption in Assumption 4.2, and analyzes its local stability in Lemma B.1, Proposition B.2, and Figure 2. It also reports robustness over NFE and exploration strength in Figure 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The main sampler assumptions and theoretical statements are stated in Section 4, including Assumption 4.2, Lemma 4.1, and Theorem 4.3. Appendix B.2 states and proves Lemma B.1 and Proposition B.2; complete derivations are provided in Appendix B.1, Appendix B.2, and Appendix B.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 specifies the reward models, prompt sources and splits, backbone models, LoRA settings, NFE, training iterations, evaluation protocol, and sampler exploration strengths, with the main protocols reported in Table 1. The PRECISE transition rule is derived in Theorem 4.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is publicly available at `https://github.com/Tencent-Hunyuan/Precise`, including the PRECISE sampler implementation and reproduction scripts. The experimental setup is described in Section 5, and the sampler transition rule is specified in Theorem 4.3.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 reports the prompt datasets and held-out split, reward models, SD3.5-M backbone, resolution, LoRA hyperparameters, use of GRPO-Guard, CFG/KL settings, NFE, iteration counts, and sampler exploration strengths. Section 5.4 specifies the corresponding FLUX.2 Klein settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For the main SD3.5-M and FLUX.2 comparisons in Table 1 and Table 3, we report means and standard deviations over three evaluation seeds when available. Due to compute constraints, each training configuration is run once.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 states that training runs use 8$\times$ NVIDIA H20 GPUs with approximately 80GB GPU memory per H20. The same section reports the main SD3.5-M 10-NFE and 30-NFE protocols, their iteration counts, and their approximate 3-day execution time, and reports that the FLUX.2 Klein 20-NFE protocol takes approximately 1 day on the same 8-GPU worker. The task-level and exploration-strength experiments use the same worker type and the stated 10-NFE/3000-iteration protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The experiments use existing pretrained models, benchmarks, reward models, and prompt datasets described in Section 5, and do not involve new human-subject data collection. We are not aware of any aspect of the study that violates the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix E discusses positive impacts from more controllable and efficient visual generation, negative dual-use and bias risks from improved text-to-image post-training, and mitigation directions such as reward-model audits, safety filters, provenance mechanisms, and expanded evaluation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The manuscript does not introduce or release a new pretrained model, image generator, or scraped dataset; it evaluates a sampler using existing backbones and benchmarks in Section 5. Therefore, release safeguards for new high-risk assets are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Section 5 credits the existing codebase, backbones, prompt datasets, benchmarks, and reward models through citations. Appendix D summarizes their sources and licenses, and states that we do not redistribute third-party model weights or datasets beyond their stated terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new sampler, released at `https://github.com/Tencent-Hunyuan/Precise`. The sampler transition rule is documented in Section 4, and the repository includes implementation and reproduction scripts.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The study does not conduct crowdsourcing or new research with human subjects. Section 5 uses existing benchmarks, prompt datasets, and reward models.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or new human-subject studies, as the experiments in Section 5 rely on existing datasets, benchmarks, and reward models. IRB approval is therefore not applicable to this work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method is a stochastic sampler and SDE-consistent transition for flow-matching models, developed in Section 4. The paper does not use LLMs as an important, original, or non-standard component of the research methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.