# Designing Reinforcement Learning for Diffusion Models: A Unified Path-Space View

**Yixian Xu**[1*], **Yuanrui Zhang**[1*], **Shengjie Luo**[1†], **Kaiyuan Gao**[2], **Liwei Wang**[1], **Di He**[1†]
[1]State Key Laboratory of General Artificial Intelligence,
School of Intelligence Science and Technology, Peking University
[2]Huazhong University of Science and Technology
{xyx050,yuanruizhang25}@stu.pku.edu.cn, luoshengjie2408@gmail.com,
im_kai@hust.edu.cn, {wanglw,dihe}@pku.edu.cn

## Abstract

Reinforcement learning (RL) post-training provides a direct way to align diffusion models with human preferences and task-specific rewards. However, current RL algorithms for diffusion models remain fragmented: reverse-trajectory methods rely on discretized likelihood ratios, whereas forward-matching methods train on reward-labeled noising versions of the rollout samples. This paper shows that these seemingly different losses arise from a single path-space principle. Starting from the regularized diffusion-RL objective, we use importance sampling between sampling SDEs to obtain an explicit policy-gradient estimator on trajectory space. The estimator contains the stochastic Itô integral underlying Flow-GRPO-type updates; we derive an equivalent variance-reduced value-gradient form that recovers the forward-matching structure of AWM and DiffusionNFT. This identifies the empirical gap between these method families as a variance-reduction effect rather than a difference in RL principle. The derivation yields a unified design space organized by value-gradient estimation, weight functions, and sampling choices. Within this space, we propose a multi-sample KDE value-gradient estimator that reuses rollout groups, together with scale-bounded weight families that retain stable existing recipes while excluding singular ones. Experiments on SD3.5-M and Qwen-Image models validate the variance-reduction explanation and show that the resulting recipe improves over prior diffusion-RL baselines. The code is available at `https://github.com/xyx050/DesigningDiffusionRL.git`.

## 1 Introduction

Diffusion and flow models have become the backbone of modern visual generation, powering high-quality image, video, and 3D synthesis [11, 16, 23, 25, 34]. The pretrained generator is optimized for distributional alignment via score matching, but not for the downstream preferences that users ultimately care about, including human aesthetic preference, compositional correctness, text rendering, or task-specific verifiability. Reinforcement learning (RL) post-training is therefore a natural next step that treats the generator as a policy and directly optimizes rewards evaluated on generated samples, including black-box or non-differentiable rewards that are hard to express as supervised losses.

This promise has led to a rapidly growing family of diffusion-RL algorithms. Starting from the raw objective of maximizing reward under the sampling policy with a reference regularizer, the central obstacle is that diffusion models do not provide the convenient policy likelihood ratios that PPO-style RL [32] relies on. Different practical routes around this obstacle have been proposed: some discretize the reverse-sampling SDE so that each denoising step has a Gaussian transition

[*]Equal contribution.
[†]Correspondence to: Shengjie Luo <luoshengjie2408@gmail.com>, Di He<dihe@pku.edu.cn>,.

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

ratio [5, 24, 37]; others avoid reverse-time likelihoods and instead build supervised flow-matching losses from reward-labeled clean samples and their forward-noised versions [40, 45]. These routes produce loss functions that look structurally unrelated, and the community currently lacks a unified view for understanding existing recipes or designing principled diffusion-RL algorithms beyond combinations of heuristics.

Our first step is to expose the common continuous-time structure beneath these routes. Although a single diffusion path measure has no elementary likelihood, ratios between path measures generated by stochastic differential equations (SDEs) with the same diffusion coefficient have explicit form [2, 29]. Applying this path-space importance sampling to the raw diffusion-RL objective yields a tractable estimator under a proposal path measure. Its policy gradient term contains a stochastic Itô integral that corresponds to the noise exploration signal used by reverse-sampling-based GRPO methods [24]. We prove that this term has an equivalent deterministic value-gradient representation. This variance-reduced representation is the bridge from Flow-GRPO-style stochastic estimators to forward-matching methods: the large empirical gap between Flow-GRPO and DiffusionNFT/AWM is explained as a variance reduction effect rather than as a fundamentally different RL principle.

The variance-reduced form also reveals the design space. Its only unspecified component is how to estimate the value gradient $\nabla_{\mathbf{x}_t} V_t$ on sampled trajectories, together with how to weight the resulting policy gradient and quadratic regularization terms across time. A Bayes rule expression for $\nabla_{\mathbf{x}_t} V_t$ yields a deterministic one-sample estimator that recovers AWM and DiffusionNFT under their forward noising proposal, while the Brownian increment estimator recovers Flow-GRPO-type methods under reverse SDE rollouts. Lifting these choices to free parameters gives a unified template whose concrete entries specify the proposal dynamics, the value gradient estimator, the off-policy weight $\hat{w}_1$, the on-policy weight $\hat{w}_2$, and the sampler. These entries are organized into three axes: value gradient estimation, weight functions, and sampling. Under this template, previously separate algorithms become a specification of each entry in the same design space rather than separate design principles.

Finally, the framework turns this taxonomy into concrete design guidance. Along the estimator axis, we observe that existing methods all use one-sample estimators of $\nabla_{\mathbf{x}_t} V_t$; we introduce a multi-sample KDE estimator that reuses the rollout group already collected for GRPO-style advantages and reduces conditional variance without additional sampling. Along the weight axis, we identify a scale-bounded principle: the effective per-step update $\hat{w}_2(t)\widehat{\nabla V}_t$ and the quadratic regularizer $\hat{w}_1(t)\|\Delta \mathbf{v}_\theta\|^2$ should remain bounded across time and discretization scale. This principle recovers the stable weightings used by AWM, DiffusionNFT, and GRPO-Guard, rules out singular Flow-GRPO-style weights in off-policy regimes, and reduces algorithm design to a small family of interpretable weight shapes.

**Contributions.** In summary, this paper contributes:

- **A continuous-time path-space framework (Sec. 3.1, Sec. 3.2).** Starting from the raw objective Eq. (3), we use path-space importance sampling and a variance-reduction theorem (Thm. 2) to derive an explicit template Eq. (9) that unifies Flow-GRPO [24], DanceGRPO [41], TempFlow-GRPO [14],GRPO-Guard [37], AWM [40], and DiffusionNFT [45] as special cases (Table 1).
- **Principled design choices (Sec. 3.3).** We introduce a multi-sample KDE value-gradient estimator Eq. (10) and a scale-bounded weight principle that yields compact families for $\hat{w}_1$ and $\hat{w}_2$, recovering strong existing recipes while exposing the remaining tunable degrees of freedom.
- **Empirical validation (Sec. 4).** On SD3.5-M [11] and Qwen-Image [38] models across PickScore [19], OCR [7], and GenEval [13] rewards, we verify that the variance-reduction trick closes the Flow-GRPO–DiffusionNFT efficiency gap, that the proposed weight families track empirical optima, and that our final recipe exceeds the baselines in Table 1.

## 2 Backgrounds

### 2.1 Diffusion and Flow-Matching Generative Models

Diffusion and Flow-Matching generative models perform step-by-step transformations between the data distribution and a tractable Gaussian prior. The forward process noises data to a Gaussian prior,

and the reverse process generates data by denoising. Following the Flow-Matching formulation [1, 23, 25], the forward noising process is constructed as

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}, \quad t \in [0,1], \tag{1}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{x}_0 \sim p_{\text{data}}$. The Flow-Matching formulation adopts the vector-field prediction loss to train the diffusion model:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim p(t), \mathbf{x}_0 \sim p_{\text{data}}, \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\boldsymbol{\epsilon} - \mathbf{x}_0)\|^2,$$

where $\mathbf{v}_\theta(\mathbf{x}_t, t)$ is parameterized by a neural network and $p(t)$ is the distribution of the time steps. Once the model is trained, the sampling process can be performed by simulating the following SDE:

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{v}_\theta(\mathbf{x}_t, t) + \frac{\eta_t}{1-t}\big(\mathbf{x}_t + (1-t)\mathbf{v}_\theta(\mathbf{x}_t, t)\big)\right] \mathrm{d}t + \sqrt{\frac{2t\eta_t}{1-t}}\, \mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $\eta_t \geq 0$ is a non-negative function and $\mathbf{w}_t$ is the standard Brownian motion on $\mathbb{R}^d$. When $\eta_t = 1$, Eq. (2) is the time-reversal of Eq. (1); when $\eta_t = 0$, it collapses to the deterministic ODE $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \mathbf{v}_\theta(\mathbf{x}_t, t)$, i.e. the FM analogue of the probability flow ODE [34].

### 2.2 Reinforcement Learning for Diffusion Models

Let $\mathcal{C}$ be a prompt space and let $R : \mathbb{R}^d \times \mathcal{C} \to [0,1]$ be a reward function evaluated on clean samples. We denote $\mathbf{x}_{0:1}$ as the trajectory of the stochastic process $\{\mathbf{x}_t\}_{t\in[0,1]}$ in Eq. (2) and $p(\mathbf{x}_{0:1}, c)$ as its likelihood[3]. RL post-training seeks to maximize the following canonical objective:

$$\mathcal{L}(\theta) = \mathbb{E}_c\Big[\mathbb{E}_{p_\theta(\mathbf{x}_{[0:1]}, c)}[R(\mathbf{x}_0, c)] - \beta \mathrm{KL}(p_\theta \| p_{\text{ref}})\Big], \tag{3}$$

where $p_{\text{ref}}$ is a frozen reference policy.

Recently, several practical alternatives for Eq. (3) have been proposed. Flow-GRPO [24] applies an Euler–Maruyama discretization to the reverse-time sampling SDE, treating each denoising step as a tractable Gaussian transition and running clipped PPO over the discretized trajectory. Denote the current diffusion policy as $p_\theta$ and the old policy as $p_{\text{old}}$. Given a group of old-policy trajectories $\{\{\mathbf{x}^i_{t_j}\}_{j=1}^T\}_{i=1}^G$, Flow-GRPO optimizes:

$$\mathcal{L}^{\text{GRPO}}(\theta) = \mathbb{E}_{c, \{\mathbf{x}^i_0\}_{i=1}^G \sim p_{\text{old}}(\cdot|c)} \frac{1}{G}\sum_{i=1}^G \frac{1}{T}\sum_{j=1}^T \left(\min\left(r^i_j(\theta)A^i, \tilde{r}^i_j(\theta)A^i\right) - \beta \mathrm{KL}(p_\theta | p_{\text{ref}})\right).$$

Here $A^i = \frac{R(\mathbf{x}^i_0, c) - \text{mean}(\{\mathbf{x}^i_0\}_{i=1}^G)}{\text{std}(\{\mathbf{x}^i_0\}_{i=1}^G)}$ is the group-normalized reward of the $i$-th sample in the group $\{\mathbf{x}^i_0\}_{i=1}^G$, $\tilde{r}^i_j = \text{clip}(r^i_j(\theta), 1-\epsilon, 1+\epsilon)$ is the clipped likelihood ratio of the $j$-th step for the $i$-th sample where $r^i_j = \frac{p_\theta(\mathbf{x}^i_{t_{j-1}} | \mathbf{x}^i_{t_j}, c)}{p_{\text{old}}(\mathbf{x}^i_{t_{j-1}} | \mathbf{x}^i_{t_j}, c)}$ is Euler–Maruyama per-step likelihood ratio.

A different line of methods forgoes reverse-trajectory likelihood ratios entirely and constructs training signals from reward-labeled clean samples via the forward noising path. AWM [40] samples $\mathbf{x}_0$ from the old policy, evaluates its advantage $A(\mathbf{x}_0, c)$, applies the forward noising path $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$, and reuses the flow-matching target with advantage weighting[4]:

$$-\mathcal{L}^{\text{AWM}}(\theta) = \mathbb{E}_{c, \mathbf{x}_0, t, \boldsymbol{\epsilon}}\Big[A(\mathbf{x}_0, c)\ \|\mathbf{v}_\theta(\mathbf{x}_t, t, c) - (\boldsymbol{\epsilon} - \mathbf{x}_0)\|^2\Big].$$

DiffusionNFT [45] follows the same forward-noising interface but uses the reward to define positive and negative velocity mixtures around the old model, $\mathbf{v}^+ = (1-\beta)\mathbf{v}_{\text{old}} + \beta\mathbf{v}_\theta$ and $\mathbf{v}^- = (1+\beta)\mathbf{v}_{\text{old}} - \beta\mathbf{v}_\theta$:

$$-\mathcal{L}^{\text{NFT}}(\theta) = \mathbb{E}_{c, \mathbf{x}_0, t, \boldsymbol{\epsilon}}\Big[R(\mathbf{x}_0, c)\left\|\mathbf{v}^+(\mathbf{x}_t, t, c) - (\boldsymbol{\epsilon} - \mathbf{x}_0)\right\|^2 + (1 - R(\mathbf{x}_0, c))\left\|\mathbf{v}^-(\mathbf{x}_t, t, c) - (\boldsymbol{\epsilon} - \mathbf{x}_0)\right\|^2\Big].$$

At first glance, these methods appear to be driven by different principles and yield structurally unrelated loss functions. Without a unified view that connects them, the field still lacks a principled basis for understanding existing recipes or systematically designing new diffusion-RL algorithms.

[3] This is a formal notation that treats the probability space defined on continuous paths as its discretization. Actually, this is known as the path measure corresponding to a stochastic differential equation (SDE). See Appx. B for background.

[4] Note that for AWM and DiffusionNFT, we also use $\mathcal{L}$ to represent the maximum objective.

# 3 A Unified Path-Space View of Diffusion RL

## 3.1 A Path-Space View via Importance Sampling

**The structural obstacle of Diffusion RL.** Despite the variety of practical recipes, RL for diffusion models ultimately rests on the canonical objective Eq. (3). Its policy-gradient estimator $\nabla_\theta \mathbb{E}_{p_\theta}[R]$ formally requires the score $\nabla_\theta \log p_\theta(\mathbf{x}_0 \mid c)$, but $p_\theta(\mathbf{x}_0 \mid c)$, the marginal of the full generation trajectory, has no closed form. The trajectory likelihood $p_\theta(\mathbf{x}_{[0:1]}, c)$ fares no better: as the path measure of a continuous-time SDE, the path law is naturally defined as a measure on $C([0,1], \mathbb{R}^d)$, not as a tractable density with respect to a finite-dimensional reference measure [29]. The log-derivative trick that underlies PPO/GRPO [32, 33, 44] therefore does not apply directly. This is the structural obstacle that any RL algorithm on a diffusion generator must confront. Viewed through this perspective, the three methods of Sec. 2.2 are different workarounds to the same obstacle. Flow-GRPO discretises the reverse SDE into tractable Gaussian steps, while AWM and DiffusionNFT bypass reverse-time likelihoods entirely and match on the forward noising path.

**Path-Space Importance Sampling.** We try to address the obstacle directly. The key observation is that although a single diffusion path measure has no density, *pairs* of path measures generated by SDEs that share the same diffusion coefficient admit a closed-form importance sampling ratio [2, 29]. Concretely, if we sample trajectories from a tractable proposal $q$ drawn from the same SDE family (Eq. (2)) with drift $\mathbf{v}_{\text{base}}$ in place of $\mathbf{v}_\theta$ and the same noise schedule $\eta_t$, the ratio $p_\theta/q$ is explicit and differentiable in $\theta$ under certain conditions.

This is exactly importance sampling in the *path space*: we sample entire trajectories $\{\mathbf{x}_t\}_{t\in[0,1]}$ rather than only the endpoint $\mathbf{x}_0$. Applying this technique to Eq. (3) and replacing the reward $R(\mathbf{x}_0, c)$ with the group-relative advantage $A(\mathbf{x}_0, c)$ (which leaves the policy gradient unchanged while reducing its variance), we obtain the unbiased estimator

$$\mathcal{L}(\theta, c) = \mathbb{E}_{q(\mathbf{x}_{[0:1]},c)}\left[\frac{p_\theta(\mathbf{x}_{[0:1]}, c)}{q(\mathbf{x}_{[0:1]}, c)} A(\mathbf{x}_0, c)\right] - \beta \mathbb{E}_{q(\mathbf{x}_{[0:1]},c)}\left[\frac{p_\theta(\mathbf{x}_{[0:1]}, c)}{q(\mathbf{x}_{[0:1]}, c)} \log \frac{p_\theta(\mathbf{x}_{[0:1]}, c)}{p_{\text{ref}}(\mathbf{x}_{[0:1]}, c)}\right], \quad (4)$$

where $p_\theta$, $p_{\text{ref}}$, and $q$ correspond to the sampling SDE (Eq. (2)) with velocity fields $\mathbf{v}_\theta$, $\mathbf{v}_{\text{ref}}$, and $\mathbf{v}_{\text{base}}$ respectively, and the same noise schedule $\eta_t$. In this way, the importance sampling ratio $M_1(\theta) := \log \frac{p_\theta(\mathbf{x}_{[0:1]},c)}{q(\mathbf{x}_{[0:1]},c)}$ is given by (see Appx. B):

$$M_1(\theta) = -\int_0^1 \sqrt{2\tilde{w}_{1-r}}\, \Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r) \cdot \mathrm{d}\mathbf{w}_r - \int_0^1 \tilde{w}_{1-r} \left\|\Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r)\right\|^2 \mathrm{d}r,$$

where $r = 1 - t$, $\Delta\mathbf{v}_\theta := \mathbf{v}_\theta - \mathbf{v}_{\text{base}}$ and the weight $\tilde{w}_t = \frac{(1+\eta_t)^2}{4\eta_t} \cdot \frac{1-t}{t}$ couples the training weight to the sampler noise schedule. Substituting $M_1(\theta)$ into Eq. (4), the objective becomes a single expectation under $q$:

$$\mathcal{L}(\theta, c) = \mathbb{E}_q[\exp(M_1(\theta))\, A(\mathbf{x}_0, c)] - \beta\, \mathbb{E}_q\Big[\exp(M_1(\theta))\, \big(M_1(\theta) - \log \tfrac{p_{\text{ref}}(\mathbf{x}_{[0:1]},c)}{q(\mathbf{x}_{[0:1]},c)}\big)\Big]. \quad (5)$$

**An explicit variance-reduced template.** We focus here on the policy-gradient term of Eq. (5), $\mathcal{L}^{\text{policy}}(\theta, c) := \mathbb{E}_q[\exp(M_1(\theta))A(\mathbf{x}_0, c)]$; the KL term is handled identically. Under mild assumptions that follow standard PPO practice [32, 33], the explicit form of $\mathcal{L}^{\text{policy}}(\theta, c)$ can be exactly obtained (See Appx. C.2 for proof):

**Proposition 1** (Stochastic-integral path-space estimator)**.** *Assume clipping or a small step size keeps* $\exp(M_1(\theta^-)) \approx 1$[5]*, where $\theta^-$ denotes the stop-gradient version of the current parameters. Then*

$$-\mathcal{L}^{\text{policy}}(\theta, c) =_\theta \mathbb{E}_q\left[\underbrace{A(\mathbf{x}_0, c)\int_0^1 \sqrt{2\tilde{w}_{1-r}}\, \Delta\mathbf{v}_\theta \cdot \mathrm{d}\mathbf{w}_r}_{\textit{Term (A): noise exploration}} + \underbrace{A(\mathbf{x}_0, c)\int_0^1 \tilde{w}_{1-r} \left\|\Delta\mathbf{v}_\theta\right\|^2 \mathrm{d}r}_{\textit{Term (B): regularisation}}\right], \quad (6)$$

*where $=_\theta$ denotes equality up to a $\theta$-independent constant.*

[5]Although the IS ratio $\exp(M_1(\theta^-))$ has an expectation of 1, it remains strictly below 1 with high probability in high-dimensional latent spaces. See Appx. E for details.

Intuitively, Term (B) is an advantage-weighted quadratic that anchors $\mathbf{v}_\theta$ to $\mathbf{v}_{\text{base}}$, while Term (A) follows the Brownian increments weighted by advantage — pushing $\mathbf{v}_\theta$ toward the noise realisations that yielded higher rewards. The latter is an Itô integral, however, whose variance dominates the overall variance of Eq. (6). A natural question arises: can Term (A) be rewritten as a deterministic integral with the same expectation and with smaller variance? The next theorem provides an explicit variance-reduced template (See Appx. C.3 for the proof):

**Theorem 2** (Variance-reduced path-space estimator)**.** *Let $q$ be the path measure of the sampling SDE Eq.* (2) *with base drift $\mathbf{v}_{base}$ and noise schedule $\eta_t > 0$, and let $V_t(\mathbf{x}, c) := \mathbb{E}_q[A(\mathbf{x}_0, c) \mid \mathbf{x}_t = \mathbf{x}]$ be the associated value function. Under the regularity the trust-region condition of Prop. 1, the policy gradient of Eq.* (5) *admits the (locally) variance-reduced representation*

$$-\mathcal{L}^{policy}(\theta, c) = {}_\theta\mathbb{E}_q\left[\int_0^1 \Big((1+\eta_t)\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\cdot\nabla_{\mathbf{x}_t}V_t(\mathbf{x}_t, c) + A(\mathbf{x}_0, c)\tilde{w}_t\|\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\|^2\Big)\,\mathrm{d}t\right]. \quad (7)$$

From the intractable canonical objective Eq. (3), we arrive at the explicit template Eq. (7), which is both *(i) tractable* — every term is a deterministic integral under a proposal $q$ in the same SDE family as $p_\theta$, with no reliance on path-likelihood densities of $p_\theta$ that do not exist — and *(ii) variance-reduced* — the stochastic integral in Eq. (6) is replaced by a deterministic integral depending on a $\nabla_{\mathbf{x}_t}V_t$ estimator. These two properties enable Eq. (7) to serve as a unified tool to understand different RL methods for the diffusion model.

### 3.2 A Unified Framework and Its Design Space

**The missing ingredient.** Eq. (7) leaves one quantity unspecified: the value gradient $\nabla_{\mathbf{x}_t}V_t(\mathbf{x}_t, c)$. How this quantity is evaluated on each rollout together with the weights on the two terms of Eq. (7) turns out to distinguish each diffusion RL method of Sec. 2.2 from others. We begin by deriving an analytical Bayes-rule expression for $\nabla_{\mathbf{x}_t}V_t$, from which a one-sample estimator follows.

**Proposition 3.** *The gradient of the value function admits the closed-form expression*

$$\nabla_{\mathbf{x}_t}V_t(\mathbf{x}_t, c) = \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t \mid \mathbf{x}_0)\,A(\mathbf{x}_0, c)] - V_t(\mathbf{x}_t, c)\,\mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t \mid \mathbf{x}_0)]\,.$$

*Let $\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) := (\mathbf{x}_t - \mathbf{x}_0)/t$ denote the FM conditional velocity. Under $q$ with $\mathbf{v}_{base} = \mathbb{E}[\mathbf{v} \mid \mathbf{x}_t] \approx \mathbf{v}_{old}$ and $\eta_t \equiv 1$ (i.e., $q$ is the reverse of the forward noising process), this specializes in*

$$\nabla_{\mathbf{x}_t}V_t(\mathbf{x}_t, c) = -\tfrac{1-t}{t}\,\mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}\big[A(\mathbf{x}_0, c)\,\big(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{base}(\mathbf{x}_t, t)\big)\big]\,.$$

*The corresponding one-sample estimator on a rollout trajectory $\mathbf{x}_{0:1}$ is*

$$\widehat{\nabla V}_t^{\,\mathrm{det}}(\mathbf{x}_t, c) := -\tfrac{1-t}{t}A(\mathbf{x}_0, c)\big(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{old}(\mathbf{x}_t, t)\big).$$

See Appx. C.4 for the proof. Substituting $\widehat{\nabla V}_t^{\,\mathrm{det}}$ into Thm. 2 yields the concrete training objective

$$\mathbb{E}_q\left[\int_0^1\left(\underbrace{-\tfrac{(1-t)(1+\eta_t)}{t}}_{\text{weight term 1}}\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\cdot A(\mathbf{x}_0, c)\big(\mathbf{v} - \mathbf{v}_{\text{old}}(\mathbf{x}_t, t)\big) + \underbrace{A(\mathbf{x}_0, c)\,\tilde{w}_t}_{\text{weight term 2}}\|\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\|^2\right)\mathrm{d}t\right], \quad (8)$$

where $\mathbf{v} = (\mathbf{x}_t - \mathbf{x}_0)/t$ is the single-sample conditional velocity. Eq. (8) is a fully explicit, variance-controlled diffusion-RL loss obtained by instantiating the variance-reduced template Eq. (7) with the deterministic one-sample estimator of Prop. 3 under the noise schedule $\eta_t = 1$.

**A unified framework.** Moreover, the stochastic integrand of Term (A) in Eq. (6) is itself an estimator of $\nabla_{\mathbf{x}_t}V_t$ — a *stochastic* one-sample estimator using the Brownian increment of the rollout SDE, $\widehat{\nabla V}_t^{\,\mathrm{sto}}(\mathbf{x}_t, c) := A(\mathbf{x}_0, c)\sqrt{(1-t)/(2\eta_t t\Delta t)}\,\boldsymbol{\epsilon}_t$. Substituting it through the structural identity of Thm. 2 gives a second, equally valid instantiation. The two instantiations share a common two-term form: an off-policy quadratic in $\Delta\mathbf{v}_\theta$ weighted by a function of the advantage, plus an on-policy inner product between $\Delta\mathbf{v}_\theta$ and the chosen value-gradient estimator. Viewing the estimator and the two weights as free parameters yields the following unified framework:

$$-\mathcal{L}^{\text{policy}}(\theta, c) = \int_0^1 \mathbb{E}_q\Big[\underbrace{\hat{w}_1\big(A(\mathbf{x}_0, c), t\big)\|\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\|^2}_{\text{off-policy regulariser}} + \underbrace{\hat{w}_2(t)\big\langle\Delta\mathbf{v}_\theta(\mathbf{x}_t, t), \widehat{\nabla V}_t(\mathbf{x}_t, c)\big\rangle}_{\text{on-policy policy-gradient}}\Big]\,\mathrm{d}t, \quad (9)$$

Table 1: Diffusion RL methods as rows in the five-knob design space of Eq. (9). We use the FM parameterisation throughout; $\mathbf{v}$ and $\boldsymbol{\epsilon}$ denote the FM conditional velocity and the per-step Brownian increment. See Appx. D for detailed derivations of each row.

| Method | $\mathbf{v}_{\text{base}}\ (\eta_t)$ | $\widehat{\nabla V}_t$ | $s(t)$ | $\hat{w}_1(A,t)$ | $\hat{w}_2(t)$ | Sampler |
|---|---|---|---|---|---|---|
| Flow-GRPO [24, 41] | $\mathbf{v}_{\text{old}}\ (\eta_t)$ | $s(t)\cdot A\boldsymbol{\epsilon}$ | $\sqrt{\frac{1-t}{2\eta_t t\Delta t}}$ | $A\cdot\frac{(1+\eta_t)^2}{4\eta_t}\frac{1-t}{t}$ | $1+\eta_t$ | Flow-SDE |
| GRPO-Guard [37] | $\mathbf{v}_{\text{old}}\ (\eta_t)$ | $s(t)\cdot A\boldsymbol{\epsilon}$ | $\sqrt{\frac{1-t}{2\eta_t t\Delta t}}$ | $0$ | $\frac{(1+\eta_t)}{s(t)\Delta t}$ | Flow-SDE |
| TempFlow-GRPO [14] | $\mathbf{v}_{\text{old}}\ (\eta_t)$ | $s(t)\cdot A\boldsymbol{\epsilon}$ | $\sqrt{\frac{1-t}{2\eta_t t\Delta t}}$ | $A\cdot\frac{(1+\eta_t)^2}{4}\sqrt{\frac{2(1-t)\Delta t}{\eta_t t}}$ | $(1+\eta_t)/s(t)$ | Branched-SDE |
| AWM [40] | $\mathbb{E}[\mathbf{v}\mid\mathbf{x}_t]\ (1)$ | $-s(t)\cdot A(\mathbf{v}-\mathbf{v}_{\text{old}})$ | $\frac{1-t}{t}$ | $A/\Delta t$ | $\frac{2}{s(t)\Delta t}$ | Flow-SDE / ODE |
| DiffusionNFT [45] | $\mathbb{E}[\mathbf{v}\mid\mathbf{x}_t]\ (1)$ | $-s(t)\cdot A(\mathbf{v}-\mathbf{v}_{\text{old}})$ | $\frac{1-t}{t}$ | $1/\Delta t$ | $\frac{2}{\beta s(t)\Delta t}$ | DPM-ODE [26] |
| **Ours** (Sec. 3.3) | $\mathbf{v}_{\text{old}}\ (\eta_t)$ | $\widehat{\nabla V}_t^{\text{KDE}}$(Eq. (10)) | $\frac{1-t}{t}$ | $(1-t)^{\alpha_1}/\Delta t$ | $\frac{t^{\alpha_2}}{s(t)\Delta t}$ | Flow-SDE |

where $\widehat{\nabla V}_t$ is any estimator of $\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c)$ and $\hat{w}_1, \hat{w}_2$ are arbitrary weight functions. In principle, different diffusion RL methods of Sec. 2.2 turn out to be instances of Eq. (9) under specific choices of these parameters. Table 1 enumerates these choices for various methods.

**The design space.** Reading Table 1 by columns, the concrete choices in Eq. (9) group naturally into three independent axes:

- **(i) Value-gradient estimation**: the proposal $q$ (determined by the choice of $\mathbf{v}_{\text{base}}$ and $\eta_t$) together with the estimator $\widehat{\nabla V}_t$ evaluated under $q$. A typical choice takes $q$ induced by a lagged velocity model $\mathbf{v}_{\text{old}}$ and a specific noise schedule $\eta_t$.
- **(ii) Weight functions**: the off-policy weight $\hat{w}_1(A,t)$ and the on-policy weight $\hat{w}_2(t)$. When $\mathbf{v}_{\text{old}} = \mathbf{v}_\theta$ (on-policy), the off-policy regulariser of Eq. (9) vanishes and only $\hat{w}_2$ matters; when $\mathbf{v}_{\text{old}} \neq \mathbf{v}_\theta$ (off-policy), $\hat{w}_1$ and $\hat{w}_2$ jointly determine the training signal.
- **(iii) Sampler**: the numerical scheme used to simulate $q$ and collect rollout samples.

### 3.3 Elucidating Design Choices

The design space of Sec. 3.2 has two primary axes: the value-gradient estimator $\widehat{\nabla V}_t$ and the weight pair $(\hat{w}_1, \hat{w}_2)$. We elucidate each axis in turn[6]. For the estimator, we observe that methods in Table 1 rely on a one-sample Monte Carlo estimator of $\nabla_{\mathbf{x}_t} V_t$ and we introduce a multi-sample refinement based on kernel density estimation. For the weights, we extract a scale-bounded structural principle from recent advanced methods and show that it implies the $t$-dependent shape of both $\hat{w}_1$ and $\hat{w}_2$ given the choice of value-gradient estimator.

**Estimator axis: from one-sample to multi-sample.** Every row of Table 1 uses a one-sample value gradient estimator: GRPO-type methods use the original stochastic one-sample estimator $\widehat{\nabla V}_t^{\text{sto}}$ built from a single Brownian increment, while AWM and DiffusionNFT use the deterministic one-sample estimator $\widehat{\nabla V}_t^{\text{det}}$ (defined in Prop. 3) from a single $(\mathbf{x}_0, \mathbf{x}_t)$ pair. Both carry the full conditional variance of the integrand per rollout which remains unchanged by simply enlarging the group.

The natural variance reduction is to average over $K$ samples from $q(\mathbf{x}_0|\mathbf{x}_t)$, but drawing $K$ fresh samples at each query $\mathbf{x}_t$ would cost $K$ extra rollouts. Instead, we exploit the collected rollout group $\{(\mathbf{x}_0^i, \mathbf{x}_t^i)\}_{i=1}^G$. For a noisy state $\mathbf{x}_t$, we take a kernel-weighted average over the $G$ group samples with a kernel matched to the FM forward likelihood $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0, t^2\mathbf{I})$. This gives the Nadaraya–Watson KDE estimator

$$\widehat{\nabla V}_t^{\text{KDE}}(\mathbf{x}_t, c) \;:=\; -\tfrac{1-t}{t}\cdot\frac{\sum_{i=1}^G K(\mathbf{x}_t, \mathbf{x}_0^i)\, A(\mathbf{x}_0^i, c)\,\big(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0^i) - \mathbf{v}_{\text{old}}(\mathbf{x}_t, t)\big)}{\sum_{i=1}^G K(\mathbf{x}_t, \mathbf{x}_0^i)}, \tag{10}$$

where $K(\mathbf{x}_t, \mathbf{x}_0) := \exp(-\frac{\|\mathbf{x}_t-(1-t)\mathbf{x}_0\|^2}{2h\,t^2})$ is the FM-matched Gaussian kernel and $h$ is a hyper-parameter to balance the relative weight for the samples in the group, typically fixed as a constant

[6]We treat sampler choices as specific designs within value-gradient estimation (e.g., trajectory branching in MixGRPO and TempFlow-GRPO).

depends on the dimension. The KDE estimator is also inspired by Bertrand et al. [4], which shows that including one anchor posterior sample can make the self-normalized estimator unbiased. We summarize the properties of different value-gradient estimators as follows.

**Proposition 4** (Properties of the value-gradient estimators)**.** *Let $A(\mathbf{x}_0, c)$ be square-integrable. Assume $\mathbf{v}_{base} = \mathbf{v}_{old}$, then the following arguments hold:*

*(i). The discretized stochastic one-sample estimator is asymptotically unbiased in the continuous-time Itô limit of Thm. 2:* $\lim_{\Delta t \to 0} \mathbb{E}\Big[\widehat{\nabla V}_{t,\Delta t}^{\,\mathrm{sto}} \mid \mathbf{x}_t\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c)$*;*

*(ii). Assume $\eta_t \equiv 1$, $\mathbf{v}_{old}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) \mid \mathbf{x}_t]$, then the deterministic one-sample estimator is unbiased:* $\mathbb{E}\Big[\widehat{\nabla V}_t^{\,\mathrm{det}} \mid \mathbf{x}_t\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c)$*;*

*(iii). Under the assumption of (ii) and $h = 1$, the KDE estimator is unbiased and has smaller variance than the deterministic one-sample estimator:*

$$\mathbb{E}\Big[\widehat{\nabla V}_t^{\,\mathrm{KDE}}(\mathbf{x}_t^j, c) \mid \mathbf{x}_t^j\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t^j, c), \quad \mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{KDE}} \mid \mathbf{x}_t^j\Big) \le \mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{det}} \mid \mathbf{x}_t^j\Big).$$

See Appx. C.5 for the proof. Prop. 4 shows that the KDE estimator is unbiased and performs variance reduction for the deterministic one-sample estimator when $h = 1$. With the same rollout construction but $h \neq 1$, the anchor is drawn from the true posterior while the normalized weights correspond to a different kernel, so the estimator is generally biased. Thus, $h$ explicitly trades exact posterior matching for smoother finite-sample estimates; we quantify this trade-off in Sec. 4.1.

**Weight axis: a scale-bounded principle.** From Eq. (9), the on-policy gradient contribution at time $t$ has magnitude proportional to $\hat{w}_2(t)\,\widehat{\nabla V}_t(\mathbf{x}_t, c)$: this product is what scales the per-step update before the network-gradient factor $\partial_\theta \Delta \mathbf{v}_\theta$ enters. For stable training, its magnitude should be controlled across $t$: neither blowing up at particular timesteps (which would amplify gradient variance and skew the update) nor vanishing where gradient signal is needed. We therefore inspect $\hat{w}_2(t)\,\widehat{\nabla V}_t$ across the rows of Table 1:

| | | |
|---|---|---|
| Flow-GRPO: | $\hat{w}_2(t)\,\widehat{\nabla V}_t \Delta t \;=\; (1+\eta_t)\,A\,\sqrt{(1-t)\Delta t/(2\eta_t\, t)}\,\boldsymbol{\epsilon}_t$ | *(blows up as , $t \to 0$)* |
| GRPO-Guard: | $\hat{w}_2(t)\,\widehat{\nabla V}_t \Delta t \;=\; (1+\eta_t)\,A\,\boldsymbol{\epsilon}_t$ | *(bounded)* |
| AWM: | $\hat{w}_2(t)\,\widehat{\nabla V}_t \Delta t \;=\; -2\,A\,(\mathbf{v} - \mathbf{v}_{\text{old}})$ | *(bounded)* |
| DiffusionNFT: | $\hat{w}_2(t)\,\widehat{\nabla V}_t \Delta t \;=\; -\frac{2}{\beta}\,A\,(\mathbf{v} - \mathbf{v}_{\text{old}})$ | *(bounded).* |

The three methods that empirically outperform Flow-GRPO each pair their estimator with a weight whose product $\hat{w}_2\,\widehat{\nabla V}_t \Delta t$ stays bounded uniformly in $t$; Flow-GRPO alone violates this property. We conclude the pattern to an explicit design principle:

> **Scale-bounded principle.** Choose $\hat{w}_2(t)$ so that $\hat{w}_2(t)\,\widehat{\nabla V}_t \Delta t$ stays bounded uniformly in $t$ ; analogously, choose $\hat{w}_1(A, t)$ so that $\hat{w}_1(A, t)\|\Delta \mathbf{v}_\theta\|^2 \Delta t$ stays bounded uniformly in $t$.

The principle applies asymmetrically to the two weights, reflecting the asymmetric $t$-dependence of the two factors it multiplies. For $\hat{w}_2(t)$, the deterministic estimator $\widehat{\nabla V}_t^{\,\mathrm{det}}$ and our KDE estimator $\widehat{\nabla V}_t^{\,\mathrm{KDE}}$ both inherit the $(1-t)/t$ prefactor of Prop. 3; the principle then *pins down* the shape of $\hat{w}_2(t)$ up to a single free exponent. Let $s(t)$ be the scale factor of the estimator $\widehat{\nabla V}_t$ (summarized in Table 1), $w_2(t) := \hat{w}_2(t) s(t) \Delta t$. The principle requires $w_2(t)$ to stay bounded uniformly in $t$. We adopt the minimal single-exponent family $w_2(t) \;=\; t^\alpha, \alpha \ge 0$ which saturates the principle and concentrates mass more sharply near $t = 1$ for larger $\alpha$ (any multiplicative constant is omitted).

For $\hat{w}_1$, the situation is different: because $\|\Delta \mathbf{v}_\theta\|^2$ carries no $t$-singularity, the principle reduces to "$\hat{w}_1 \Delta t$ is bounded in $t$". Flow-GRPO's $\hat{w}_1 \Delta t = A\,\tilde{w}_t \Delta t \propto A(1-t)\Delta t/t$ diverges as $t \to 0$ and is ruled out; AWM's $\hat{w}_1 \Delta t = A$ and DiffusionNFT's $\hat{w}_1 \Delta t = 1$ are admissible, but the principle alone does not pick the monotonicity direction in $t$ or the presence of advantage coupling. Let $w_1(A, t) = \hat{w}_1(A, t) \Delta t$. Four minimal single-exponent candidates remain:

$$w_1(A, t) \;\in\; \big\{\, t^\alpha,\; (1-t)^\alpha,\; A(\mathbf{x}_0, c)\, t^\alpha,\; A(\mathbf{x}_0, c)\,(1-t)^\alpha \,\big\}, \quad \alpha \ge 0,$$

distinguishing **(a)** whether the trust-region tightens toward the clean-data end ($(1-t)^\alpha$) or the noise end ($t^\alpha$), and **(b)** whether its strength couples to the advantage. Which of the four shapes is optimal is an empirical question the principle cannot resolve; we ablate across the family in Sec. 4.2.

# 4 Experiments

In this section, we show the experimental results on the design of each component in our framework. In Sec. 4.1, we introduce the settings of our experiments and give an example to show how the variance reduction trick in Sec. 3.1 works. In Sec. 4.2, we show the principles of weight design for different estimators, which gives a mapping between the value-gradient estimator and its optimal weight. In Sec. 4.3, we compare our final recipe derived by following the design-space map to its optimum, and show the comparison with previous baseline models across different reward settings.

## 4.1 Settings

**Setup.** We conduct our experiments on the SD-3.5 Medium model [11] and Qwen-Image [38] at a resolution of $512 \times 512$. Following Flow-GRPO [24] and DiffusionNFT [45], we set group size $G = 24$, $T = 10$ denoising steps for training and $T = 40$ for evaluation. In addition, we choose to fine-tune the pretrained model using LoRA [17] and disable classifier-free guidance (CFG) [15] in the training and evaluation stages for fair comparison. We evaluate our methods on various rewards including PickScore [19], visual-text OCR accuracy [7], and GenEval [13]. See Appx. G for details.

**How the variance-reduction trick works.** In Sec. 3.1 and Sec. 3.2, we show the correctness of the one-sample estimator and view it as the estimator for the variance-reduced form of Term A in Eq. (6). Although the methods exhibit strong theoretical properties, it's unclear whether it helps to improve the training convergence of RL algorithms. Previous works AWM [40] and DiffusionNFT [45] show that this type of estimator accelerates the training convergence of RL training under specific weight design (Table 1), it's unclear either the estimator or the specific weight design leads to the empirical performance. To find out this, we directly compare the training performance of three training objectives: (1) the original objective Eq. (6) with $\eta_t = 1$ (corresponding to the reverse of the forward process) ; (2) the one-sample estimator with inherited weighting Eq. (8); (3) A reweighted version of Eq. (8) (same as AWM [40]).

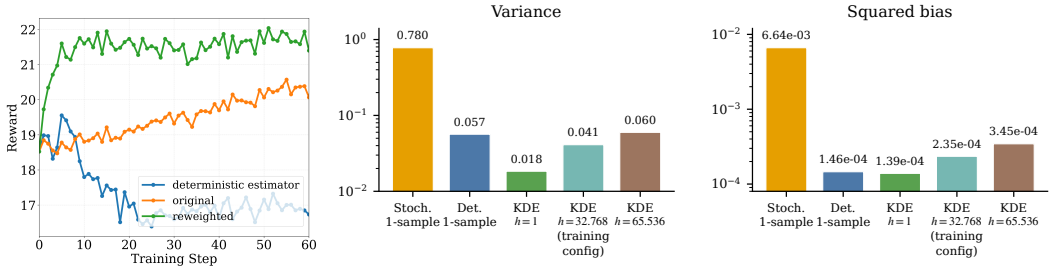
The results are shown in Fig. 1(a). The one-sample estimator with inherited weighting suffers instability and ultimately fails, indicating that the variance reduction alone can not accelerate the training; the reweighting tricks are also indispensable. With the reweighted form, the one-sample estimator achieves remarkably faster convergence speed. This observation inspires us to investigate the weight design for each value-gradient estimator in the next subsection.

**Direct bias–variance measurement.** We directly measure the estimation error in the setting that matches the conditions of Prop. 4. We set $\eta_t = 1$ and evaluate SD3.5-Medium with PickScore on four prompts, using 128 trajectories per prompt and 40 denoising steps. As shown in Fig. 1(a), the stochastic estimator has $13.8\times$ the variance of the deterministic one-sample estimator, directly supporting the variance-reduction mechanism in Thm. 2. The matched KDE estimator ($h = 1$) further reduces variance by $67.5\%$ relative to the deterministic estimator. Squared bias contributes less than $2\%$ of the MSE in every setting, so the observed error is dominated by variance.

**Bias–smoothness trade-off.** The bandwidth controls a practical bias–smoothness trade-off. Although $h = 1$ yields the lowest variance (Prop. 4 and Fig. 1(b)), its kernel weights can change abruptly across adjacent denoising steps in the high-dimensional latent space. Increasing $h$ smooths this temporal transition and improves optimization stability, at the cost of weakening posterior matching and increasing both variance and bias. We therefore use the intermediate $h$ that retains a meaningful variance reduction while producing smoother value-gradient targets along the trajectory.

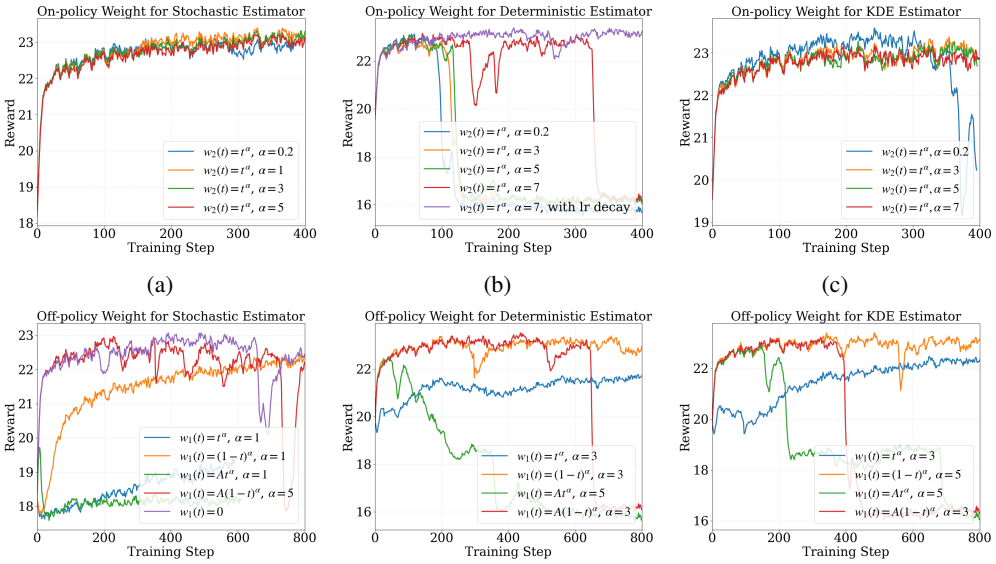
## 4.2 A Mapping between the Value-Gradient Estimator and Its Optimal Weight

**On-policy weighting $w_2$.** We study three types of estimators: stochastic one-sample estimator $\widehat{\nabla V_t}^{\mathrm{sto}}$, deterministic one-sample estimator $\widehat{\nabla V_t}^{\mathrm{det}}$ and our proposed KDE estimator $\widehat{\nabla V_t}^{\mathrm{KDE}}$. As introduced in Sec. 3.3, we investigate the weight: $w_2(t) = t^\alpha$. We find that different sampler noise




(a) (b) (c)

Figure 1: **(a)** Comparison between different objectives on Pickscore reward. Original objective Eq. (6) with $\eta_t = 1$ (Orange line); one-sample estimator with weighting in Eq. (8) (Blue line); reweighted version of Eq. (8) (Green line). **(b,c)** Empirical variance and squared bias of the value-gradient estimators. We set $\eta_t = 1$ and evaluate SD3.5-Medium with PickScore reward on four prompts, using 128 trajectories per prompt and 40 denoising steps. At every intermediate state, the reference value gradient is approximated from all 128 trajectories; each estimator is computed with $G = 24$ then compared with this common reference. The KDE training configuration uses $h = 32.768$.




(a) (b) (c)

(d) (e) (f)

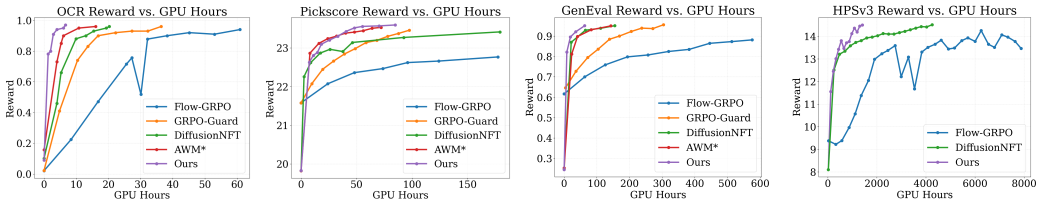
Figure 2: On-policy weight (row 1) and off-policy (row 2) weight ablations for the stochastic, deterministic, and our KDE estimator. The results for the same estimator are in the same column.

schedules affect the behavior (Appx. G.1), and we choose the best sampler noise schedule for weight ablation. The results are shown in Fig. 2 (first row). By adjusting the weights parameter $\alpha$ (see Fig. 7 for visualization), we find that a different estimator exhibits a different behavior for weight preference. The stochastic estimators are not so sensitive w.r.t. the weight adjustment, $w_2 = t$ performs slightly better (Fig. 2 (a)). Weight with larger $\alpha$ helps stabilize the training of the deterministic estimator, but can not fully resolve it without learning rate decay (Fig. 2 (b)). Our KDE estimator exhibits better training stability under weight with larger $\alpha$ without any extra design (Fig. 2 (c)). Larger $\alpha$ improves stability also implies that more training effort should be allocated to high-noise regions, which affects the sampling process at early stages.

**Off-policy weighting $w_1$.** Based on the optimal on-policy weight, we adjust the off-policy weight $w_1$. As introduced in Sec. 3.3, we adopt the following weight design: $w_1 = \{0, t^\alpha, At^\alpha, (1-t)^\alpha, A(1-t)^\alpha\}$. For stochastic estimator, we find that $w_1 = 0$ works well (design of GRPO-Guard in Table 1); the weight in $\{t^\alpha, (1-t)^\alpha\}$ overemphasize the regularization term and leads to slow convergence, while weights $\{At^\alpha, A(1-t)^\alpha\}$ leads to training instability (Fig. 2 (d)). Similarly, for the deterministic estimator and our KDE estimator, $t^\alpha$ degrades the training convergence speed and

$\{At^\alpha, A(1-t)^\alpha\}$ causes training instability (Fig. 2 (e,f)). We find that $w_1(t) = (1-t)^\alpha$ balances the weight between the off-policy regularization and the policy gradient in Eq. (9), which implies that regularization in low-noise regions helps to improve the training stability.

### 4.3 Comparison with Prior Works




(a) (b) (c) (d)

Figure 3: **(a,b,c).** Training time comparison between our method and several baselines. **(d).** Comparison between Flow-GRPO, DiffusionNFT, and our method on Qwen-Image, HPSv3 reward. * means using the official implementation with different training configs compared to other baselines.

Based on the estimator–optimal weight mapping constructed in Sec. 4.2, we perform a training-time comparison between our best setting and previous strong baselines across several rewards. We use our KDE estimator with its optimal weighting $w_2 = t^\alpha, w_1 = (1-t)^\alpha$. The results are summarized in Fig. 3 (a,b,c). On OCR reward, our method significantly accelerates training convergence compared to previous SOTA methods: $2\times$ faster than AWM, $3\times$ faster than DiffusionNFT. On Pickscore and GenEval rewards, our method achieves comparable performance with AWM and DiffusionNFT without using other designs (e.g., specific EMA-KL loss in training, elaborate sampling design, training time-step selection), and outperforms all other baselines. We also conduct experiments on the large-scale model Qwen-Image [38] with HPSv3 reward [27], and the result in Fig. 3(d) shows that our method achieves $4\times$ faster convergence speed than DiffusionNFT under the same configurations.

**Ablation study.** To isolate the source of the gain, we start the ablation study from DiffusionNFT without its EMA-KL stabilization. The results are summarized in Fig. 4. The without EMA-KL baseline is unstable and eventually fails after 60 GPU hours. Replacing only its deterministic one-sample estimator with our KDE estimator improves its stability but hurts the performance; variance reduction alone is therefore insufficient. Using our on-policy weight helps stabilize and improve training performance. Finally, combining KDE with both our on-policy and off-policy weights reaches 23.52 after 60 hours. For comparison, the original DiffusionNFT (on-policy, with EMA-KL) reaches 23.382 after 160 hours, so our recipe is roughly $3\times$ faster.

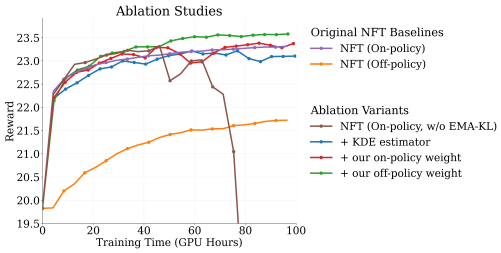



Figure 4: Ablation studies on the components of our final recipe.

## 5 Conclusion

In this work, we develop a unified path-space framework that places previous diffusion-RL algorithms on a common theoretical foundation. Starting from the original RL objective, we derive a general training objective based on the variance-reduced form of the original objective. Previous reverse-trajectory methods and forward-matching methods can all be viewed as instances of this general objective. The derivation also yields a unified design space, including the value-gradient estimation, weight functions, and sampling choices. We propose a multi-sample KDE value gradient estimator and find a mapping between different estimators and their optimal weight design. Experiments on SD3.5-M and Qwen-Image models validate our theoretical explanation and show the empirical improvement of our result recipe.

## Acknowledgments and Disclosure of Funding

DH is supported by National Science and Technology Major Project (2022ZD0114902), National Science Foundation of China (NSFC62376007), National Science Foundation of China (under Key Project No. 92570203), Beijing Natural Science Foundation (Z250001) and Beijing Major Science and Technology Project under Contract no. Z251100008425004. This work is supported in part by the State Key Laboratory of General Artificial Intelligence.

## References

[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[2] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[3] Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling rl post-training of diffusion and flow-matching models. *arXiv preprint arXiv:2605.10759*, 2026.

[4] Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the closed-form of flow matching: Generalization does not arise from target stochasticity. *Advances in neural information processing systems*, 38:8522–8549, 2026.

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

[6] George Casella and Roger Berger. *Statistical inference*. CRC press, 2024.

[7] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36: 9353–9387, 2023.

[8] 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.

[9] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

[10] Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pages 53791–53846, 2025.

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

[12] 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.

[13] 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.

[14] Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

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

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

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

[18] Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*. springer, 2014.

[19] 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.

[20] Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models. *arXiv preprint arXiv:2604.17415*, 2026.

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

[22] 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*, 2025.

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

[24] 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.

[25] 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.

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

[27] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15086–15095. IEEE, 2025.

[28] David McAllister, Miika Aittala, Tero Karras, Janne Hellsten, Angjoo Kanazawa, Timo Aila, and Samuli Laine. Finite difference flow optimization for rl post-training of text-to-image models. *arXiv preprint arXiv:2603.12893*, 2026.

[29] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[30] Peter Potaptchik, Cheuk-Kit Lee, and Michael S Albergo. Tilt matching for scalable sampling and fine-tuning. *arXiv preprint arXiv:2512.21829*, 2025.

[31] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. 2023.

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

[33] 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.

[34] 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*, 2020.

[35] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

[36] 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.

[37] 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*, 2025.

[38] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[39] 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.

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

[41] 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.

[42] 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.

[43] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37: 73366–73398, 2024.

[44] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[45] 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. *arXiv preprint arXiv:2509.16117*, 2025.

# A Organization of Appendix

The organization of the appendix is as follows. Appx. B provides background in stochastic calculus. Appx. C gives all the proofs of the theorems in the main text. Appx. D gives the derivations in Table 1. Appx. F discusses the related works. Appx. G provides experimental details for Sec. 4 and shows some additional results. Finally, qualitative examples are shown in Appx. H.

# B Backgrounds for Stochastic Analysis

In this section, we recall some basic results from stochastic calculus. Please see Oksendal [29] or Karatzas and Shreve [18] for details. Assume the probability space is $(\Omega, \mathcal{F}, \mathbb{P})$. A $d$-dimensional stochastic process is a parametrized collection of random vectors $\{\mathbf{x}_t\}_{t\in T}$, where $T$ is the time interval. For each $\omega \in \Omega$ we can consider the function $t \to \mathbf{x}_t(\omega),\ t \in T$, which is called the path of $\mathbf{x}_t$. Assume $\mathbf{x}$ is jointly measurable in $(t, \omega)$, then the stochastic process is a probability measure $\mathbb{P}$ on the measurable space $((\mathbb{R}^d)^T, \mathcal{B})$, where $(\mathbb{R}^d)^T$ contains all functions from $T$ to $\mathbb{R}^d$. In this paper, we only consider the stochastic process with continuous sample paths; then the stochastic process is a probability measure on the measurable space $(C(T, \mathbb{R}^d), \mathcal{B})$, where $C(T, \mathbb{R}^d)$ contains all continuous functions from $T$ to $\mathbb{R}^d$. In this paper, we focus on the diffusion process, which can be defined by its generator:

**Definition 5.** The *infinitesimal generator* of a stochastic process $(\mathbf{x}_t)_t$ for a function $\phi(\mathbf{x})$ is

$$\mathcal{L}_t\phi(\mathbf{x}) = \lim_{s\to 0^+} \frac{\mathbb{E}[\phi(\mathbf{x}_{t+s})|\mathbf{x}_t = \mathbf{x}] - \phi(\mathbf{x})}{s},$$

where $\phi$ is a smooth function. For an Itô process defined as the solution to the SDE $\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + g_t\mathrm{d}\mathbf{w}_t$, the generator is

$$\mathcal{L}_t = \sum_{i=1}^{d} \mathbf{f}^i(\mathbf{x}, t)\partial_i + \frac{g_t^2}{2}\sum_{i=1}^{d}\partial_i^2, \quad \text{where} \quad \partial_i = \frac{\partial}{\partial x_i}.$$

In stochastic calculus, Itô's lemma is a useful tool for calculating derivatives as the chain rule in calculus.

**Lemma 6** (Itô's lemma; [29], Thm. 4.2.1)**.** *Assume $\mathbf{x}_t$ is the solution of SDE $\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{w}_t$. Then for any function $f \in C^2(\mathbb{R}^d \times \mathbb{R})$, we have*

$$\mathrm{d}f(\mathbf{x}_t, t) = \left(\partial_t f(\mathbf{x}_t, t) + \nabla_{\mathbf{x}} f(\mathbf{x}_t, t)\cdot \mathbf{f}(\mathbf{x}_t, t) + \frac{g_t^2}{2}\Delta f(\mathbf{x}_t, t)\right)\mathrm{d}t + g_t\nabla_{\mathbf{x}} f(\mathbf{x}_t, t)\cdot\,\mathrm{d}\mathbf{w}_t.$$

For the calculation of the value function, the following Kolmogorov's backward equation is useful.

**Proposition 7** (Kolmogorov's Backward Equation; [29], Thm. 8.1.1)**.** *Let $V_t(\mathbf{x}) = \mathbb{E}[R(\mathbf{x}_T, c)|\mathbf{x}_t = \mathbf{x}]$ be the value function. Then $V$ is the solution of the following PDE*

$$\frac{\partial V}{\partial t} + \mathcal{A}_t V = 0, \quad V_T(\mathbf{x}) = R(\mathbf{x}, c),$$

*where $\mathbf{x}_t$ satisfies the SDE $\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{w}_t$ and $\mathcal{A}_t$ is its generator:*

$$\mathcal{A}f(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t)\cdot\nabla f(\mathbf{x}, t) + \frac{g_t^2}{2}\Delta f(\mathbf{x}, t), \quad \forall f \in C^2(\mathbb{R}^d \times \mathbb{R}).$$

For the calculation of the importance sampling ratio between two path measures of diffusion processes, Girsanov's theorem is useful:

**Theorem 8** (Girsanov; [29], Thm. 8.6.6)**.** *Let $\mathbf{x}_t, \mathbf{y}_t$ solve the coupled SDEs on $\mathbb{R}^d$*

$$\mathrm{d}\mathbf{x}_t = \mathbf{u}(\mathbf{x}_t, t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{w}_t, \quad \mathrm{d}\mathbf{y}_t = \mathbf{v}(\mathbf{y}_t, t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{w}_t,$$

*with the same initial condition and diffusion coefficient, and let $\mathbb{P}, \mathbb{Q}$ be their path measures. Assume $\mathbf{u}, \mathbf{v}$ are sufficiently smooth and satisfy Novikov's condition $\mathbb{E}[\exp(\frac{1}{2}\int_0^T \|(\mathbf{u}-\mathbf{v})/g_t\|^2\,\mathrm{d}t)] < \infty$. Then*

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}} = \exp\left(\int_0^T \tfrac{1}{g_t}(\mathbf{u}-\mathbf{v})(\mathbf{y}_t, t)\cdot\,\mathrm{d}\mathbf{w}_t - \int_0^T \tfrac{1}{2g_t^2}\|(\mathbf{u}-\mathbf{v})(\mathbf{y}_t, t)\|^2\,\mathrm{d}t\right).$$

# C Proofs

## C.1 Setup: path-space importance sampling

Let $\mathbf{v}_{\text{base}}, \mathbf{v}_\theta, \mathbf{v}_{\text{ref}} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ be three smooth velocity fields, and write $\mathbb{Q}, \mathbb{P}_\theta, \mathbb{P}_{\text{ref}}$ for the corresponding path measures on $C([0,1], \mathbb{R}^d)$:

$$\mathbb{P}_\theta: \ \mathrm{d}\mathbf{x}_t = \left[\mathbf{v}_\theta(\mathbf{x}_t, t) + \frac{\eta_t}{1-t}\big(\mathbf{x}_t + (1-t)\mathbf{v}_\theta(\mathbf{x}_t, t)\big)\right] \mathrm{d}t + \sqrt{\frac{2t\eta_t}{1-t}}\,\mathrm{d}\mathbf{w}_t,$$

$$\mathbb{Q}: \ \mathrm{d}\mathbf{x}_t^{\text{base}} = \left[\mathbf{v}_{\text{base}}(\mathbf{x}_t^{\text{base}}, t) + \frac{\eta_t}{1-t}\big(\mathbf{x}_t^{\text{base}} + (1-t)\mathbf{v}_{\text{base}}(\mathbf{x}_t^{\text{base}}, t)\big)\right] \mathrm{d}t + \sqrt{\frac{2t\eta_t}{1-t}}\,\mathrm{d}\mathbf{w}_t,$$

$$\mathbb{P}_{\text{ref}}: \ \mathrm{d}\mathbf{x}_t^{\text{ref}} = \left[\mathbf{v}_{\text{ref}}(\mathbf{x}_t^{\text{ref}}, t) + \frac{\eta_t}{1-t}\big(\mathbf{x}_t^{\text{ref}} + (1-t)\mathbf{v}_{\text{ref}}(\mathbf{x}_t^{\text{ref}}, t)\big)\right] \mathrm{d}t + \sqrt{\frac{2t\eta_t}{1-t}}\,\mathrm{d}\mathbf{w}_t,$$

where $\mathbf{x}_1, \mathbf{x}_1^{\text{base}}, \mathbf{x}_1^{\text{ref}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Then the original objective Eq. (3) becomes $\max \mathbb{E}_c \mathcal{L}(\theta, c)$, where

$$\mathcal{L}(\theta, c) = \mathbb{E}_{\mathbb{P}_\theta}[R(\mathbf{x}_0, c)] - \beta \mathrm{KL}(\mathbb{P}_\theta \| \mathbb{P}_{\text{ref}}).$$

If we want to use the importance sampling technique, the objective becomes

$$\mathcal{L}(\theta, c) = \mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{P}_\theta}{\mathrm{d}\mathbb{Q}} R(\mathbf{x}_0, c)\right] - \beta \mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{P}_\theta}{\mathrm{d}\mathbb{Q}} \log\left(\frac{\mathrm{d}\mathbb{P}_\theta}{\mathrm{d}\mathbb{P}_{\text{ref}}}\right)\right],$$

where $\frac{\mathrm{d}\mathbb{P}_\theta}{\mathrm{d}\mathbb{Q}}$ is the Radon-Nikodym derivative of $\mathbb{P}_\theta$ with respect to $\mathbb{Q}$. By Thm. 8,

$$M_1(\theta) = -\int_0^1 \sqrt{2\tilde{w}_{1-r}}\, \Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r) \cdot \mathrm{d}\mathbf{w}_r - \int_0^1 \tilde{w}_{1-r} \left\|\Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r)\right\|^2 \mathrm{d}r, \quad (11)$$

where $r = 1-t$, $\Delta\mathbf{v}_\theta := \mathbf{v}_\theta - \mathbf{v}_{\text{base}}$ and $\tilde{w}_t = \frac{(1+\eta_t)^2}{4\eta_t} \cdot \frac{1-t}{t}$. The results recover Eq. (5).

## C.2 Proof of Prop. 1

**Proposition 9** (Stochastic-integral path-space estimator)**.** *Assume that the clipping or a small step size keeps* $|\exp(M_1(\theta^-)) - 1| < \epsilon$ *and* $\mathbb{E}_{\mathbb{Q}}[\|A\nabla_\theta M_1(\theta)\|] < M$ *bounded, where* $\theta^-$ *denotes the stop-gradient version of the current parameters. Then* $-\nabla_\theta \mathcal{L}^{\text{policy}}(\theta, c) \to$

$$\nabla_\theta \mathbb{E}_{\mathbb{Q}}\left[A(\mathbf{x}_0, c) \int_0^1 \sqrt{2\tilde{w}_{1-r}}\, \Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r) \cdot \mathrm{d}\mathbf{w}_r + A(\mathbf{x}_0, c) \int_0^1 \tilde{w}_{1-r} \left\|\Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, r)\right\|^2 \mathrm{d}r\right],$$

*as* $\epsilon \to 0$.

***Proof.*** Consider the policy-gradient term $\mathcal{L}^{\text{policy}}(\theta, c) = \mathbb{E}_{\mathbb{Q}}[\exp(M_1(\theta))A(\mathbf{x}_0, c)]$ of Eq. (5). The gradient of the policy-gradient term is given by

$$\nabla_\theta \mathcal{L}^{\text{policy}}(\theta, c) = \mathbb{E}_{\mathbb{Q}}[A(\mathbf{x}_0, c) \exp(M_1(\theta^-)) \nabla_\theta M_1(\theta)].$$

Under the assumption $|\exp(M_1(\theta^-)) - 1| < \epsilon$,

$$\begin{aligned}
&\|\nabla_\theta \mathcal{L}^{\text{policy}}(\theta, c) - \mathbb{E}_{\mathbb{Q}}[A(\mathbf{x}_0, c)\nabla_\theta M_1(\theta)]\| \\
&= \left\|\mathbb{E}_{\mathbb{Q}}\big[(\exp(M_1(\theta^-)) - 1) A \nabla_\theta M_1(\theta)\big]\right\| \le \mathbb{E}_{\mathbb{Q}}\big[|\exp(M_1(\theta^-)) - 1| \|A\nabla_\theta M_1(\theta)\|\big] \\
&\le \epsilon M.
\end{aligned}$$

By Eq. (11),

$$M_1(\theta) = -\int_0^1 \sqrt{2\tilde{w}_{1-r}}\, \Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r) \cdot \mathrm{d}\mathbf{w}_r - \int_0^1 \tilde{w}_{1-r} \left\|\Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, r)\right\|^2 \mathrm{d}r.$$

Then we conclude that $-\nabla_\theta \mathcal{L}^{\text{policy}}(\theta, c) \to -\mathbb{E}_{\mathbb{Q}}[A(\mathbf{x}_0, c)\nabla_\theta M_1(\theta)] =$

$$\nabla_\theta \mathbb{E}_{\mathbb{Q}}\left[A(\mathbf{x}_0, c) \int_0^1 \sqrt{2\tilde{w}_{1-r}}\, \Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, 1-r) \cdot \mathrm{d}\mathbf{w}_r + A(\mathbf{x}_0, c) \int_0^1 \tilde{w}_{1-r} \left\|\Delta\mathbf{v}_\theta(\mathbf{x}_{1-r}, r)\right\|^2 \mathrm{d}r\right],$$

as $\epsilon \to 0$. □

### C.3 Proof of Thm. 2

**Convention on time direction.** Before starting the proof, we fix the time convention. The FM-time $t \in [0,1]$ used in the main text has $t = 0$ at the clean sample and $t = 1$ at pure noise; under this convention, the sampling process runs backward from $t = 1$ to $t = 0$. For the Itô calculus below, we use a sampling time $r := 1 - t \in [0,1]$ and denote $\mathbf{y}_r := \mathbf{x}_{1-r} = \mathbf{x}_t$ that runs forward during sampling, with $r = 0$ at the initial noise $\mathbf{y}_0 = \mathbf{x}_1$ and $r = 1$ at the final sample $\mathbf{y}_1 = \mathbf{x}_0$.

**Theorem 10** (Variance-reduced path-space estimator)**.** *Let $q$ be the path measure of the sampling SDE Eq.* (2) *with base drift $\mathbf{v}_{base}$ and noise schedule $\eta_t > 0$, and let $V_t(\mathbf{x}, c) := \mathbb{E}_q[A(\mathbf{x}_0, c) \mid \mathbf{x}_t = \mathbf{x}]$ be the associated value function. Additionally, assume that the SDE coefficients and $V_t$ are smooth. Under the trust-region condition of Prop. 1, the policy gradient of Eq.* (5) *admits the (locally) variance-reduced representation*

$$-\mathcal{L}^{policy}(\theta, c) = {}_\theta\mathbb{E}_q\left[\int_0^1 \Big((1+\eta_t)\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\cdot\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c) + A(\mathbf{x}_0, c)\tilde{w}_t\|\Delta\mathbf{v}_\theta(\mathbf{x}_t, t)\|^2\Big)\,\mathrm{d}t\right]. \quad (12)$$

*Let $\widehat{\nabla V}^{\mathrm{sto}}_{t,\Delta t} := A(\mathbf{x}_0, c)\sqrt{(1-t)/(2\eta_t t\Delta t)}\,\boldsymbol{\epsilon}_t$ be the local discretization of Term (A) in Eq.* (6)*. Assume $\widehat{\nabla V}^{\mathrm{sto}}_{t,\Delta t}$ is square-integrable for any interior time $t \in (0,1)$. When $\Delta t \downarrow 0$, we have*

$$\mathrm{Var}_{\mathbb{Q}}\Big(\widehat{\nabla V}^{\mathrm{sto}}_{t,\Delta t}\Big) \geq \mathrm{Var}_{\mathbb{Q}}(\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c))\,. \quad (13)$$

***Proof.*** We prove the identity for the policy-gradient term; the quadratic Term (B) in Eq. (6) is carried through unchanged. To avoid mixing the two time directions, we use the forward sampling time process $\mathbf{y}_r := \mathbf{x}_{1-r}$ for $r \in [0,1]$. Then the sampling SDE (Eq. (2)) becomes

$$\mathbb{Q}: \quad \mathrm{d}\mathbf{y}_r = \underbrace{-\Big[\mathbf{v}_{\text{base}}(\mathbf{x}_r, 1-r) + \frac{\eta_{1-r}}{r}\big(\mathbf{y}_r + r\mathbf{v}_{\text{base}}(\mathbf{y}_r, 1-r)\big)\Big]}_{:=\boldsymbol{\mu}_r(\mathbf{y}_r)}\,\mathrm{d}r + \underbrace{\sqrt{\frac{2(1-r)\eta_{1-r}}{r}}}_{:=g_r}\,\mathrm{d}\mathbf{w}_r,$$

where $\mathbf{y}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Under $\mathbb{Q}$, we simply denote the sampling process as

$$\mathrm{d}\mathbf{y}_r = \boldsymbol{\mu}_r(\mathbf{y}_r)\,\mathrm{d}r + g_r\,\mathrm{d}\mathbf{w}_r. \quad (14)$$

All calculations can first be performed on $[\varepsilon, 1-\varepsilon]$ and then extended to $[0,1]$ by square-integrability and a limiting argument. Let

$$\begin{aligned} H_r &:= \sqrt{2\tilde{w}_{1-r}}\,\Delta\mathbf{v}_\theta(\mathbf{y}_r, 1-r), \quad (15)\\ I_r &:= \int_0^r H_s\cdot\,\mathrm{d}\mathbf{w}_s,\\ \bar{V}_r(\mathbf{x}, c) &:= \mathbb{E}_{\mathbb{Q}}[A(\mathbf{y}_1, c) \mid \mathbf{y}_r = \mathbf{x}], \qquad V_r := \bar{V}_r(\mathbf{y}_r, c). \end{aligned}$$

By square-integrability, $I_r$ is an Itô martingale. The process $V_r = \mathbb{E}_{\mathbb{Q}}[A(\mathbf{y}_1, c) \mid \mathcal{F}_r]$ is also a martingale by the tower property, with $V_1 = A(\mathbf{y}_1, c) = A(\mathbf{x}_0, c)$. The Kolmogorov backward equation for $\bar{V}_r$ under Eq. (14) is

$$\partial_r\bar{V}_r + \boldsymbol{\mu}_r\cdot\nabla_{\mathbf{x}}\bar{V}_r + \frac{g_r^2}{2}\Delta_{\mathbf{x}}\bar{V}_r = 0. \quad (16)$$

Applying Itô's formula and using Eq. (16) eliminates the drift term:

$$\begin{aligned} \mathrm{d}V_r &= \partial_r\bar{V}_r(\mathbf{y}_r, c)\,\mathrm{d}r + \nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r, c)\cdot\,\mathrm{d}\mathbf{y}_r + \frac{1}{2}\Delta\bar{V}_r(\mathbf{y}_r, c)\,\mathrm{d}[\mathbf{y}_r, \mathbf{y}_r],\\ &= \partial_r\bar{V}_r(\mathbf{y}_r, c)\,\mathrm{d}r + \nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r, c)\cdot(\boldsymbol{\mu}_r(\mathbf{y}_r)\,\mathrm{d}r + g_r\,\mathrm{d}\mathbf{w}_r) + \frac{g_r^2}{2}\Delta\bar{V}_r(\mathbf{y}_r, c)\,\mathrm{d}r,\\ &= \underbrace{\Big(\partial_r\bar{V}_r(\mathbf{y}_r, c) + \nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r, c)\cdot\boldsymbol{\mu}_r(\mathbf{y}_r) + \frac{g_r^2}{2}\Delta\bar{V}_r(\mathbf{y}_r, c)\Big)}_{=0,\text{ by Eq. (16)}}\,\mathrm{d}r + g_r\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r, c)\cdot\,\mathrm{d}\mathbf{w}_r\\ &= g_r\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r, c)\cdot\,\mathrm{d}\mathbf{w}_r, \quad (17) \end{aligned}$$

where $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$ is the Laplace operator and $[\cdot,\cdot]$ is the quadratic variation of the diffusion process. It remains to identify the cross-variation of the two martingales. Eq. (15) and Eq. (17) give

$$\begin{aligned}\mathrm{d}[V,I]_r &= g_r\sqrt{2\tilde{w}_{1-r}}\,\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\;\mathrm{d}r\\ &= (1+\eta_{1-r})\,\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\;\mathrm{d}r,\end{aligned}$$

where the second equality follows from

$$g_r\sqrt{2\tilde{w}_{1-r}} = \sqrt{\frac{2t\eta_t}{1-t}}\sqrt{\frac{(1+\eta_t)^2(1-t)}{2\eta_t t}} = 1+\eta_t, \qquad t = 1-r.$$

Itô's product rule now yields

$$\begin{aligned}\mathrm{d}(V_rI_r) &= V_r\,\mathrm{d}I_r + I_r\,\mathrm{d}V_r + \mathrm{d}[V,I]_r\\ &= \big(V_rH_r + I_rg_r\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\big)\cdot\mathrm{d}\mathbf{w}_r + (1+\eta_{1-r})\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\;\mathrm{d}r.\end{aligned}$$

After integration,

$$\begin{aligned}V_1I_1 = V_0I_0 &+ \int_0^1 \big[V_rH_r + g_rI_r\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\big]\cdot\mathrm{d}\mathbf{w}_r\\ &+ \int_0^1 (1+\eta_{1-r})\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\;\mathrm{d}r.\end{aligned}$$

Take expectation on both sides and by $I_0 = 0$, we get

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}[V_1I_1] &= \mathbb{E}_{\mathbb{Q}}\left[A(\mathbf{x}_0,c)\int_0^1\sqrt{2\tilde{w}_{1-r}}\,\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\mathrm{d}\mathbf{w}_r\right]\\ &= \mathbb{E}_{\mathbb{Q}}\left[\int_0^1(1+\eta_{1-r})\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\nabla\bar{V}_r(\mathbf{y}_r,c)\;\mathrm{d}r\right]\\ &\quad+\underbrace{\mathbb{E}_{\mathbb{Q}}\left[\int_0^1\big(V_rH_r+g_rI_r\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\big)\cdot\mathrm{d}\mathbf{w}_r\right]}_{=0,\quad\text{by the martingale property of Itô integral}}\\ &= \mathbb{E}_{\mathbb{Q}}\left[\int_0^1(1+\eta_{1-r})\Delta\mathbf{v}_\theta(\mathbf{y}_r,1-r)\cdot\nabla\bar{V}_r(\mathbf{y}_r,c)\;\mathrm{d}r\right].\end{aligned}$$

Changing variables back to FM-time $t = 1-r$ turns the right-hand side into the first term of Eq. (12). Restoring the unchanged quadratic Term (B) gives

$$-\mathcal{L}^{\text{policy}}(\theta,c) = {}_\theta\mathbb{E}_q\left[\int_0^1\Big((1+\eta_t)\Delta\mathbf{v}_\theta(\mathbf{x}_t,t)\cdot\nabla_{\mathbf{x}_t}V_t(\mathbf{x}_t,c) + A(\mathbf{x}_0,c)\tilde{w}_t\|\Delta\mathbf{v}_\theta(\mathbf{x}_t,t)\|^2\Big)\;\mathrm{d}t\right].$$

which is exactly Eq. (12).

We now prove Eq. (13). For a positive sampling-time step $\Delta r$,

$$\widehat{\nabla V}^{\,\text{sto}}_{t,\Delta r} = \frac{A(\mathbf{y}_1,c)}{g_r\Delta r}\big(\mathbf{w}_{r+\Delta r}-\mathbf{w}_r\big), \qquad t = 1-r.$$

By the tower property, the martingale identity $V_{r+\Delta r} = \mathbb{E}_{\mathbb{Q}}[A(\mathbf{y}_1,c)\mid\mathcal{F}_{r+\Delta r}]$, and Eq. (17),

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}\Big[\widehat{\nabla V}^{\,\text{sto}}_{t,\Delta r}\mid\mathcal{F}_r\Big] &= \frac{1}{g_r\Delta r}\mathbb{E}_{\mathbb{Q}}[(V_{r+\Delta r}-V_r)(\mathbf{w}_{r+\Delta r}-\mathbf{w}_r)\mid\mathcal{F}_r]\\ &= \frac{1}{g_r\Delta r}\mathbb{E}_{\mathbb{Q}}\left[\int_r^{r+\Delta r} g_s\nabla_{\mathbf{x}}\bar{V}_s(\mathbf{y}_s,c)\;\mathrm{d}s\,\middle|\,\mathcal{F}_r\right].\end{aligned}$$

By Jensen's inequality and Fubini's theorem,

$$\begin{aligned}\Big\|\mathbb{E}_{\mathbb{Q}}\Big[\widehat{\nabla V}^{\,\text{sto}}_{t,\Delta r}\mid\mathcal{F}_r\Big]-\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\Big\|_{L^2(\mathbb{Q})} &\le \frac{1}{g_r\Delta r}\int_r^{r+\Delta r}\big\|g_s\nabla_{\mathbf{x}}\bar{V}_s(\mathbf{y}_s,c)-g_r\nabla_{\mathbf{x}}\bar{V}_r(\mathbf{y}_r,c)\big\|_{L^2(\mathbb{Q})}\;\mathrm{d}s\\ &\xrightarrow[\Delta r\downarrow 0]{} 0.\end{aligned}$$

By the Markov property, $\mathbb{E}_{\mathbb{Q}}[\widehat{\nabla V}^{\text{sto}}_{t,\Delta t} \mid \mathcal{F}_r] = \mathbb{E}_q[\widehat{\nabla V}^{\text{sto}}_{t,\Delta t} \mid \mathbf{x}_t]$. By the property of conditional variance,

$$\begin{aligned}\operatorname{Var}_{\mathbb{Q}}\Big(\widehat{\nabla V}^{\text{sto}}_{t,\Delta t}\Big) &= \operatorname{Var}_{\mathbb{Q}}\Big(\mathbb{E}_{\mathbb{Q}}[\widehat{\nabla V}^{\text{sto}}_{t,\Delta t} \mid \mathbf{x}_t]\Big) + \mathbb{E}_q\Big[\operatorname{Var}_q\Big(\widehat{\nabla V}^{\text{sto}}_{t,\Delta t} \mid \mathbf{x}_t\Big)\Big] \\ &\geq \operatorname{Var}_{\mathbb{Q}}\Big(\mathbb{E}_{\mathbb{Q}}[\widehat{\nabla V}^{\text{sto}}_{t,\Delta t} \mid \mathbf{x}_t]\Big) \xrightarrow[\Delta t \downarrow 0]{} \operatorname{Var}_{\mathbb{Q}}(\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c))\,.\end{aligned}$$

Identifying the positive step magnitudes $\Delta r = \Delta t$ proves Eq. (13). □

### C.4 Proof of Prop. 3

**Proposition 11.** *The gradient of the value function admits the closed-form expression*

$$\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c) = \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0)\, A(\mathbf{x}_0, c)] - V_t(\mathbf{x}_t, c)\, \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0)]\,.$$

*Let $\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) := (\mathbf{x}_t - \mathbf{x}_0)/t$ denote the FM conditional velocity. Under $q$ with $\mathbf{v}_{base} = \mathbb{E}[\mathbf{v} \mid \mathbf{x}_t] \approx \mathbf{v}_{old}$ and $\eta_t \equiv 1$ (i.e., $q$ is the reverse of the forward noising process), this specialises to*

$$\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c) = -\tfrac{1-t}{t}\, \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}\big[A(\mathbf{x}_0, c)\,\big(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{base}(\mathbf{x}_t, t)\big)\big]\,.$$

*The corresponding one-sample estimator on a rollout trajectory $\mathbf{x}_{0:1}$ is*

$$\widehat{\nabla V}^{\text{det}}_t(\mathbf{x}_t, c) := -\tfrac{1-t}{t} A(\mathbf{x}_0, c)\big(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{old}(\mathbf{x}_t, t)\big).$$

***Proof.*** Under $\mathbb{Q}$ with $\eta_t \equiv 1$, the sampling SDE reduces to the reverse of the forward noising process (Sec. 2.1), and the forward transition kernel is $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0, t^2\mathbf{I})$. Differentiating the value function, we get

$$\begin{aligned}\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \frac{\mathbb{E}_{\mathbf{x}_0}[R(\mathbf{x}_0, c) q(\mathbf{x}_t|\mathbf{x}_0)]}{\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0)]} \\ &= \frac{\mathbb{E}_{\mathbf{x}_0}[\nabla_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_0) R(\mathbf{x}_0, c)]}{\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0)]} - \frac{\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0) R(\mathbf{x}_0, c)]\, \mathbb{E}_{\mathbf{x}_0}[\nabla_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_0)]}{\left(\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0)]\right)^2} \\ &= \frac{\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) R(\mathbf{x}_0, c)]}{\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0)]} \\ &\quad - \frac{\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0) R(\mathbf{x}_0, c)]\, \mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0)]}{\left(\mathbb{E}_{\mathbf{x}_0}[q(\mathbf{x}_t|\mathbf{x}_0)]\right)^2} \\ &= \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}\left[\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) R(\mathbf{x}_0, c)\right] - V_t(\mathbf{x}_t) \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0)]. \qquad (18)\end{aligned}$$

The Gaussian score of $q(\mathbf{x}_t|\mathbf{x}_0)$ is

$$\nabla_{\mathbf{x}} \log q(\mathbf{x}_t \mid \mathbf{x}_0) = -\frac{\mathbf{x}_t - (1-t)\mathbf{x}_0}{t^2} = -\frac{1-t}{t} \cdot \mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \frac{1}{t}\mathbf{x}_t,$$

where $\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) := (\mathbf{x}_t - \mathbf{x}_0)/t$ is the FM conditional velocity. Substituting into Eq. (18) and replacing $R$ with the group-relative advantage $A$ (it can be viewed as a normalized reward function),

$$\nabla_{\mathbf{x}} V_t(\mathbf{x}_t, c) = -\tfrac{1-t}{t}\, \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[A(\mathbf{x}_0, c)(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbb{E}[\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0)|\mathbf{x}_t]]\,.$$

Any good approximation $\mathbf{v}_{\text{old}}(\mathbf{x}_t, t) \approx \mathbb{E}[\mathbf{v}|\mathbf{x}_t]$ (which holds whenever $\mathbf{v}_{\text{old}}$ is a well-trained diffusion model) gives

$$\nabla_{\mathbf{x}} V_t(\mathbf{x}_t, c) = -\tfrac{1-t}{t}\, \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)}[A(\mathbf{x}_0, c)\,(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{\text{old}}(\mathbf{x}_t, t))]\,,$$

whose one-sample Monte Carlo estimator is $\widehat{\nabla V}_t = -\frac{1-t}{t} A(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{\text{old}}(\mathbf{x}_t, t))$. □

### C.5 Proof of Prop. 4

**Proposition 12** (Properties of the value-gradient estimators)**.** *Let $A(\mathbf{x}_0, c)$ be square-integrable. Assume $\mathbf{v}_{base} = \mathbf{v}_{old}$, then the following arguments hold:*

*(i). The discretized stochastic one-sample estimator is asymptotically unbiased in the continuous-time Itô limit of Thm. 2: $\lim_{\Delta t \to 0} \mathbb{E}\Big[\widehat{\nabla V}^{\text{sto}}_{t,\Delta t} \mid \mathbf{x}_t\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c)$;*

*(ii). Assume $\eta_t \equiv 1$, $\mathbf{v}_{old}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) \mid \mathbf{x}_t]$, then the deterministic one-sample estimator is unbiased: $\mathbb{E}\Big[\widehat{\nabla V}_t^{\,\mathrm{det}} \mid \mathbf{x}_t\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c)$;*

*(iii). Under the assumption of (ii) and $h = 1$, the KDE estimator is unbiased and has smaller variance than the deterministic one-sample estimator:*

$$\mathbb{E}\Big[\widehat{\nabla V}_t^{\,\mathrm{KDE}}(\mathbf{x}_t^j, c) \mid \mathbf{x}_t^j\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t^j, c), \quad \mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{KDE}} \mid \mathbf{x}_t^j\Big) \le \mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{det}} \mid \mathbf{x}_t^j\Big).$$

***Proof***. Again, we use the forward sampling time $r = 1 - t$ from the proof of Thm. 2.

**Proof of (i).** Let $\mathcal{F}_r$ be the filtration generated by the proposal process. For a step $\Delta r > 0$, define

$$\boldsymbol{\epsilon}_{r,\Delta r} := \frac{\mathbf{w}_{r+\Delta r} - \mathbf{w}_r}{\sqrt{\Delta r}}.$$

Recall from Eq. (17) that the value martingale $V_r = \mathbb{E}_{\mathbb{Q}}[A(\mathbf{y}_1, c) \mid \mathcal{F}_r]$ satisfies

$$\mathrm{d}V_r = g_r \nabla_{\mathbf{x}} \bar{V}_r(\mathbf{y}_r, c) \cdot \mathrm{d}\mathbf{w}_r, \qquad g_r = \sqrt{\frac{2t\eta_t}{1-t}}, \quad t = 1 - r.$$

By the tower property and the fact that $\mathbf{w}_{r+\Delta r} - \mathbf{w}_r$ has zero conditional mean,

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[A(\mathbf{y}_1, c)(\mathbf{w}_{r+\Delta r} - \mathbf{w}_r) \mid \mathcal{F}_r] &= \mathbb{E}_{\mathbb{Q}}[\mathbb{E}[A(\mathbf{y}_1, c)(\mathbf{w}_{r+\Delta r} - \mathbf{w}_r) \mid \mathcal{F}_{r+\Delta r}] \mid \mathcal{F}_r] \\
&= \mathbb{E}_{\mathbb{Q}}[\mathbb{E}[A(\mathbf{y}_1, c) \mid \mathcal{F}_{r+\Delta r}](\mathbf{w}_{r+\Delta r} - \mathbf{w}_r) \mid \mathcal{F}_r] \\
&= \mathbb{E}_{\mathbb{Q}}[V_{r+\Delta r}(\mathbf{w}_{r+\Delta r} - \mathbf{w}_r) \mid \mathcal{F}_r] \\
&= \mathbb{E}_{\mathbb{Q}}[(V_{r+\Delta r} - V_r)(\mathbf{w}_{r+\Delta r} - \mathbf{w}_r) \mid \mathcal{F}_r] \\
&= g_r \nabla_{\mathbf{x}} \bar{V}_r(\mathbf{y}_r, c)\, \Delta r + o(\Delta r),
\end{aligned}$$

where the last equality follows from the Itô isometry and the continuity of the martingale integrand. Consequently,

$$\mathbb{E}_{\mathbb{Q}}\left[A(\mathbf{y}_1, c)\frac{\boldsymbol{\epsilon}_{r,\Delta r}}{\sqrt{\Delta r}} \,\middle|\, \mathcal{F}_r\right] = g_r \nabla_{\mathbf{x}} \bar{V}_r(\mathbf{y}_r, c) + o(1).$$

The stochastic estimator in Sec. 3.2 can be written as

$$\widehat{\nabla V}_{t,\Delta r}^{\,\mathrm{sto}} = \frac{1}{g_r} A(\mathbf{y}_1, c)\frac{\boldsymbol{\epsilon}_{r,\Delta r}}{\sqrt{\Delta r}}.$$

Taking $\Delta r \to 0$ and using the Markov property, so that conditioning on $\mathcal{F}_r$ reduces to conditioning on $\mathbf{y}_r = \mathbf{x}_t$, yields

$$\lim_{\Delta r \to 0} \mathbb{E}_{\mathbb{Q}}\Big[\widehat{\nabla V}_{t,\Delta r}^{\,\mathrm{sto}} \mid \mathbf{x}_t\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c).$$

Identifying the positive sampling-time step $\Delta r$ with the discretization magnitude $\Delta t$ in the proposition proves (i). Thus the stochastic one-sample estimator is asymptotically unbiased.

**Proof of (ii).** When $\eta_t \equiv 1$, the transition kernel is $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0, t^2\mathbf{I})$. Under the assumption $\mathbf{v}_{\mathrm{old}}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) \mid \mathbf{x}_t]$, by Prop. 3,

$$\nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c) = -\frac{1-t}{t}\,\mathbb{E}\big[A(\mathbf{x}_0, c)\big(\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{\mathrm{old}}(\mathbf{x}_t, t)\big) \mid \mathbf{x}_t\big].$$

The random vector inside this conditional expectation is precisely $\widehat{\nabla V}_t^{\,\mathrm{det}}$. Therefore

$$\mathbb{E}\Big[\widehat{\nabla V}_t^{\,\mathrm{det}} \mid \mathbf{x}_t\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t, c),$$

which proves the unbiasedness of the deterministic one-sample estimator.

**Proof of (iii).** The results can be directly deduced from Prop. 2 of Bertrand et al. [4] (see also Exercise 7.32 of Casella and Berger [6]), we give the detailed proof here for completeness. Fix a query sample $(\mathbf{x}_t^j, c)$ from the rollout group and write $q_0$ for the endpoint marginal under the proposal. At $h = 1$, the kernel in Eq. (10) is proportional to the likelihood $q(\mathbf{x}_t^j \mid \mathbf{x}_0)$; hence Bayes' rule gives

$$q(\mathbf{x}_0 \mid \mathbf{x}_t^j) = \frac{K(\mathbf{x}_t^j, \mathbf{x}_0) q_0(\mathbf{x}_0)}{\mathbb{E}_{\mathbf{z}\sim q_0}[K(\mathbf{x}_t^j, \mathbf{z})]}.$$

Conditioned on $\mathbf{x}_t^j$, the anchor endpoint $\mathbf{x}_0^j$ is therefore distributed according to $q(\mathbf{x}_0 \mid \mathbf{x}_t^j)$, while the remaining $G-1$ endpoints are independent samples from $q_0$. To expose the resulting Rao–Blackwell structure, introduce an auxiliary index $J \sim \mathrm{Unif}\{1, \dots, G\}$, sample $\mathbf{X}_J$ from $q(\mathbf{x}_0 \mid \mathbf{x}_t^j)$, and sample $\{\mathbf{X}_i\}_{i\neq J}$ independently from $q_0$. Because the KDE estimator is symmetric in the group elements, this random relabeling does not change its distribution.

Define

$$F(\mathbf{X}_i) := -\frac{1-t}{t} A(\mathbf{X}_i, c)\big(\mathbf{v}(\mathbf{x}_t^j, \mathbf{X}_i) - \mathbf{v}_{\text{old}}(\mathbf{x}_t^j, t)\big), \qquad K_i := K(\mathbf{x}_t^j, \mathbf{X}_i).$$

Then by Bayes' rule,

$$\mathbb{P}(J = i \mid \mathbf{X}_1, \dots, \mathbf{X}_G) = \frac{K_i}{\sum_{\ell=1}^G K_\ell}.$$

Since the KDE estimator is the conditional expectation of the deterministic anchor estimator:

$$\mathbb{E}[F(\mathbf{X}_J) \mid \mathbf{X}_1, \dots, \mathbf{X}_G] = \frac{\sum_{i=1}^G K_i F(\mathbf{X}_i)}{\sum_{i=1}^G K_i} = \widehat{\nabla V}_t^{\,\mathrm{KDE}}(\mathbf{x}_t^j, c).$$

Applying the tower property and the result of (ii),

$$\mathbb{E}\Big[\widehat{\nabla V}_t^{\,\mathrm{KDE}}(\mathbf{x}_t^j, c) \mid \mathbf{x}_t^j\Big] = \mathbb{E}\Big[F(\mathbf{X}_J) \mid \mathbf{x}_t^j\Big] = \nabla_{\mathbf{x}_t} V_t(\mathbf{x}_t^j, c),$$

which means the KDE estimator is unbiased under the condition. Finally, applying the property of conditional variance, we get

$$\begin{aligned}\mathrm{Var}\Big(F(\mathbf{X}_J) \mid \mathbf{x}_t^j\Big) &= \mathbb{E}\Big[\mathrm{Var}\Big(F(\mathbf{X}_J) \mid \mathbf{X}_{1:G}, \mathbf{x}_t^j\Big)\Big|\mathbf{x}_t^j\Big] + \mathrm{Var}\Big(\mathbb{E}[F(\mathbf{X}_J) \mid \mathbf{X}_{1:G}, \mathbf{x}_t^j]\Big|\mathbf{x}_t^j\Big) \\ &\geq \mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{KDE}}(\mathbf{x}_t^j, c)\Big|\mathbf{x}_t^j\Big).\end{aligned}$$

Since $F(\mathbf{X}_J)$ has the same conditional distribution as $\widehat{\nabla V}_t^{\,\mathrm{det}}$, then

$$\mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{KDE}} \mid \mathbf{x}_t^j\Big) \leq \mathrm{Var}\Big(\widehat{\nabla V}_t^{\,\mathrm{det}} \mid \mathbf{x}_t^j\Big).$$

□

# D Results in Table 1

In this section, we provide detailed derivations of results in Table 1.

## D.1 Flow-GRPO, GRPO-Guard and TempFlow-GRPO

**Flow-GRPO [24].** The stochastic-integral path-space estimator of the Flow-GRPO objective is Eq. (6). Applying an Euler–Maruyama discretization to Eq. (6) gives

$$\begin{aligned}-\mathcal{L}^{\mathrm{GRPO}}(\theta, c) &=_\theta \mathbb{E}_q\left[A(\mathbf{x}_0, c)\int_0^1 \sqrt{2\tilde{w}_{1-r}}\,\Delta\mathbf{v}_\theta \cdot \mathrm{d}\mathbf{w}_r + A(\mathbf{x}_0, c)\int_0^1 \tilde{w}_{1-r}\,\|\Delta\mathbf{v}_\theta\|^2\,\mathrm{d}r\right] \\ &\approx \mathbb{E}_q\left[\sum_{i=0}^{N-1}\Big(A(\mathbf{x}_0, c)\sqrt{2\tilde{w}_{1-r_i}}\,\Delta\mathbf{v}_\theta \cdot \sqrt{\Delta r_i}\boldsymbol{\epsilon}_i + A(\mathbf{x}_0, c)\tilde{w}_{1-r_i}\,\|\Delta\mathbf{v}_\theta\|^2\,\Delta r_i\Big)\right] \\ &= \mathbb{E}_q\left[\sum_{i=0}^{N-1}\Big(A(\mathbf{x}_0, c)\sqrt{2\tilde{w}_{1-r_i}}\,\Delta\mathbf{v}_\theta \cdot \frac{1}{\sqrt{\Delta r_i}}\boldsymbol{\epsilon}_i + A(\mathbf{x}_0, c)\tilde{w}_{1-r_i}\,\|\Delta\mathbf{v}_\theta\|^2\Big)\Delta r_i\right]\end{aligned}$$

With the change of variable $t = 1 - r$, let $\Delta r_i = r_{i+1} - r_i = t_i - t_{i+1} = \Delta t_i > 0$. The sum can then be written entirely in terms of $t$ as

$$-\mathcal{L}^{\text{GRPO}}(\theta, c) \approx \mathbb{E}_q \left[ \sum_{i=0}^{N-1} \left( A(\mathbf{x}_0, c) \sqrt{2\tilde{w}_{t_i}}\, \Delta \mathbf{v}_\theta \cdot \frac{1}{\sqrt{\Delta t_i}} \boldsymbol{\epsilon}_i + A(\mathbf{x}_0, c) \tilde{w}_{t_i} \left\| \Delta \mathbf{v}_\theta \right\|^2 \right) \Delta t_i \right]. \quad (19)$$

Substituting $\tilde{w}_t = \frac{(1+\eta_t)^2}{4\eta_t} \frac{1-t}{t}$ into Eq. (19) directly gives the Flow-GRPO expressions in Table 1:

$$\widehat{\nabla V}_t^{\,\text{sto}} = s(t) A(\mathbf{x}_0, c) \boldsymbol{\epsilon}, \quad s(t) = \sqrt{\frac{1-t}{2\eta_t t \Delta t}},$$
$$\hat{w}_1(A, t) = A \frac{(1+\eta_t)^2}{4\eta_t} \frac{1-t}{t}, \quad \hat{w}_2(t) = 1 + \eta_t.$$

**GRPO-Guard [37].** The difference between Flow-GRPO and GRPO-Guard is that GRPO-Guard drops the regularization Term (B) and normalizes Term (A) in Eq. (6), corresponding to changes in weight $\hat{w}_1(A, t)$ and $\hat{w}_2(t)$:

$$\hat{w}_1(A, t) = 0, \quad \hat{w}_2(t) = (1 + \eta_t) \sqrt{\frac{2\eta_t t}{(1-t)\Delta t}}.$$

**TempFlow-GRPO [14].** TempFlow-GRPO first rolls out an ODE trajectory. At each point $\mathbf{x}_t$ along this trajectory, it takes a single SDE step to generate a group of branch states $\mathbf{x}_{t-\Delta t}$, each of which is then rolled out to the endpoint using an ODE solver. The resulting trajectories form a branching structure, which is the reason we call it branched-SDE sampler. TempFlow-GRPO also reweights the policy loss by $g_t \sqrt{\Delta t}$, where $g_t = \sqrt{2\eta_t t/(1-t)}$, thus its weights are

$$\hat{w}_1(A, t) = A \frac{(1+\eta_t)^2}{4} \sqrt{\frac{2(1-t)\Delta t}{\eta_t t}}, \quad \hat{w}_2(t) = (1 + \eta_t) \sqrt{\frac{2\eta_t t \Delta t}{1-t}}.$$

### D.2 AWM and DiffusionNFT

**Advantage Weighted Matching (AWM) [40].** Recall from Sec. 2.2 that AWM first draws a rollout sample $\mathbf{x}_0 \sim p_{\text{old}}(\cdot \mid c)$ (with Flow-SDE or Flow-ODE, based on different rewards) and then constructs an intermediate state by forward noising:

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

The training objective is [7]

$$-\mathcal{L}^{\text{AWM}}(\theta) = \mathbb{E}_{c, \mathbf{x}_0 \sim p_{\text{old}},\, t \sim p(t),\, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[ A(\mathbf{x}_0, c) \left\| \mathbf{v}_\theta(\mathbf{x}_t, t, c) - \mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) \right\|^2 \right]. \quad (20)$$

In our design space, we consider the reverse sampling process Eq. (2) to obtain $\mathbf{x}_{[0:1]}$. To close the gap between the forward and reverse process, we construct the reverse sampling process with

$$\mathbf{v}_{\text{base}}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) \mid \mathbf{x}_t] \;\text{ with }\; \eta_t = 1, \quad \text{where } \mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) := \frac{\mathbf{x}_t - \mathbf{x}_0}{t}.$$

Under such choices, the reverse sampling process Eq. (2) is exactly the time reversal of the forward noising path above, which means that $(\mathbf{x}_0, \mathbf{x}_t)$ constructed by the reverse sampling process and forward noising process share the same joint distributions. Thus, the AWM is equivalent to using $\mathbf{v}_{\text{base}}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) \mid \mathbf{x}_t]$ and $\eta_t \equiv 1$ in our design space. Under this condition, Prop. 3 gives

$$\widehat{\nabla V}_t^{\,\text{det}}(\mathbf{x}_t, c) = -s(t) A(\mathbf{x}_0, c) \big( \mathbf{v}(\mathbf{x}_t, \mathbf{x}_0) - \mathbf{v}_{\text{old}}(\mathbf{x}_t, t) \big), \quad s(t) := \frac{1-t}{t}. \quad (21)$$

[7] The original AWM objective additionally includes an importance-sampling ratio between the current and rollout policies. This ratio is exactly one in the on-policy setting used by the practical algorithm and remains close to one under the mildly off-policy updates considered in practice. We therefore omit it from the present derivation.

We specify the weights

$$\hat{w}_1(A,t) = \frac{A}{\Delta t}, \quad \hat{w}_2(t) = \frac{2}{s(t)\Delta t}.$$

Substituting these weights and the estimator Eq. (21) into Eq. (9) gives

$$\begin{aligned} -\mathcal{L}^{\text{policy}}(\theta) &= \int_0^1 \frac{1}{\Delta t}\mathbb{E}_q\Big[A\|\Delta\mathbf{v}_\theta\|^2 - 2A\,\langle\Delta\mathbf{v}_\theta, \mathbf{v}-\mathbf{v}_{\text{old}}\rangle\Big]\,\mathrm{d}t \\ &=_\theta \int_0^1 \frac{1}{\Delta t}\mathbb{E}_q\Big[A(\mathbf{x}_0,c)\|\mathbf{v}_\theta(\mathbf{x}_t,t,c)-\mathbf{v}(\mathbf{x}_t,\mathbf{x}_0)\|^2\Big]\,\mathrm{d}t, \end{aligned} \tag{22}$$

.

For the $\Delta t$ term, partition $[0,1]$ into $n$ timestep bins $I_i$ of widths $\Delta t_i$, and write $\Delta t(t) = \Delta t_i$ for $t \in I_i$. We define the density by

$$p(t) := \frac{1}{n\Delta t_i}, \quad t \in I_i.$$

Using $1/\Delta t(t) = np(t)$, Eq. (22) becomes

$$-\mathcal{L}^{\text{policy}}(\theta) =_\theta n\,\mathbb{E}_{c,\mathbf{x}_0\sim p_{\text{old}},\,t\sim p(t),\,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\Big[A(\mathbf{x}_0,c)\|\mathbf{v}_\theta(\mathbf{x}_t,t,c)-\mathbf{v}(\mathbf{x}_t,\mathbf{x}_0)\|^2\Big].$$

This is the same as the AWM loss Eq. (20) up to the positive constant factor $n$, which does not change the optimization process. The timesteps sampled by the original AWM objective from $p(t)$ are precisely the points used to discretize the continuous-time integral in our formulation.

**DiffusionNFT [45].** Similar to AWM, DiffusionNFT first rolls out with the DPM-Solver [26] to obtain $\mathbf{x}_0$ and then samples $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$. It minimizes the objective

$$-\mathcal{L}^{\text{NFT}}(\theta) = \mathbb{E}_{c,\,\mathbf{x}_0\sim p_{\text{old}},\,\mathbf{x}_t\sim p(\cdot|\mathbf{x}_0),\,t}\Big[\hat{R}(\mathbf{x}_0,c)\big\|\mathbf{v}^+-\mathbf{v}\big\|^2 + \big(1-\hat{R}(\mathbf{x}_0,c)\big)\big\|\mathbf{v}^- - \mathbf{v}\big\|^2\Big],$$

where

$$\mathbf{v}^+ = (1-\beta)\mathbf{v}_{\text{old}}(\mathbf{x}_t,t,c) + \beta\mathbf{v}_\theta(\mathbf{x}_t,t,c), \quad \mathbf{v}^- = (1+\beta)\mathbf{v}_{\text{old}}(\mathbf{x}_t,t,c) - \beta\mathbf{v}_\theta(\mathbf{x}_t,t,c),$$
$$\hat{R}(\mathbf{x}_0,c) = \frac{A(\mathbf{x}_0,c)+1}{2}, \quad \mathbf{v} = (\mathbf{x}_t-\mathbf{x}_0)/t.$$

Define

$$\Delta\mathbf{v}_\theta := \mathbf{v}_\theta - \mathbf{v}_{\text{old}}, \quad \mathbf{d} := \mathbf{v} - \mathbf{v}_{\text{old}}.$$

Then $\mathbf{v}^+ - \mathbf{v} = \beta\Delta\mathbf{v}_\theta - \mathbf{d}$ and $\mathbf{v}^- - \mathbf{v} = -\beta\Delta\mathbf{v}_\theta - \mathbf{d}$. Substituting these identities and expanding the two squares gives

$$\begin{aligned} -\mathcal{L}^{\text{NFT}}(\theta) &= \mathbb{E}\Big[\hat{R}\|\beta\Delta\mathbf{v}_\theta - \mathbf{d}\|^2 + (1-\hat{R})\|-\beta\Delta\mathbf{v}_\theta - \mathbf{d}\|^2\Big] \\ &= \mathbb{E}\Big[\beta^2\|\Delta\mathbf{v}_\theta\|^2 - 2\beta(2\hat{R}-1)\Delta\mathbf{v}_\theta\cdot\mathbf{d} + \|\mathbf{d}\|^2\Big] \\ &=_\theta \beta^2\mathbb{E}_{c,\,\mathbf{x}_0\sim p_{\text{old}},\,\mathbf{x}_t\sim p(\cdot|\mathbf{x}_0),\,t}\left[\|\Delta\mathbf{v}_\theta\|^2 - \Delta\mathbf{v}_\theta\cdot\frac{2A}{\beta}(\mathbf{v}-\mathbf{v}_{\text{old}})\right], \end{aligned}$$

The scaling factor $\beta^2$ does not change the optimization process; thus, we have

$$-\mathcal{L}^{\text{NFT}}(\theta) = \mathbb{E}_{c,\,\mathbf{x}_0\sim p_{\text{old}},\,\mathbf{x}_t\sim p(\cdot|\mathbf{x}_0),\,t}\left[\|\Delta\mathbf{v}_\theta\|^2 - \Delta\mathbf{v}_\theta\cdot\frac{2A}{\beta}(\mathbf{v}-\mathbf{v}_{\text{old}})\right].$$

Therefore, similarly to AWM, DiffusionNFT is recovered in our design space by choosing $\mathbf{v}_{\text{base}} = \mathbb{E}[\mathbf{v}\mid\mathbf{x}_t]$, $\eta_t = 1$, and the deterministic one-sample estimator Eq. (21). The weights functions are specified by $\hat{w}_1(t) = 1/\Delta t$ and $\hat{w}_2(t) = 2/(\beta s(t)\Delta t)$.

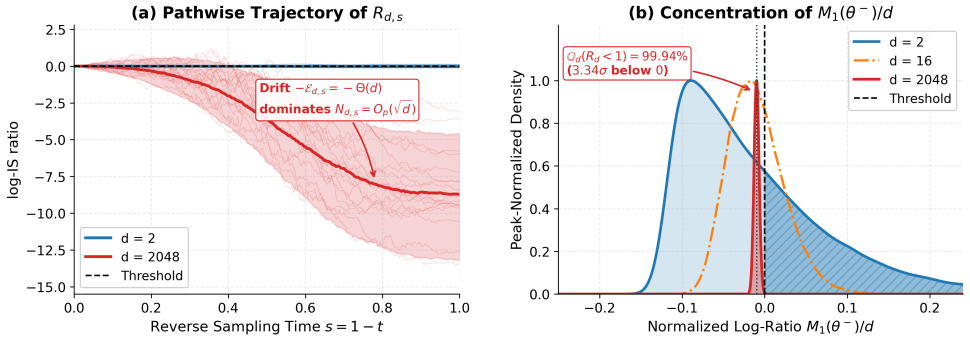



Figure 5: High-dimensional concentration of the path-space IS ratio $R_d = \exp(M_1(\theta^-))$ under the Flow-Matching SDE. **(a)** Along the reverse sampling time $s = 1 - t \in [0, 1]$, the log-IS ratio for $d = 2$ fluctuates around 0 ($R_{2,s} = 1$), whereas for $d = 2048$, the negative linear drift dominates the martingale term and drives the trajectories exponentially below $R_{d,s} = 1$. **(b)** Peak-normalized density of the normalized log-ratio $M_1(\theta^-)/d$, which concentrates sharply around $-\bar{\kappa} = -0.010 < 0$ as the dimension $d$ grows from 2 to 2048.

## E High-dimensional Behavior of the Path-space IS Ratio

We highlight an interesting phenomenon regarding the importance sampling (IS) ratio in Prop. 1 under the proposal path measure. Although the IS ratio $\exp(M_1(\theta^-))$ has an expectation of 1, it remains strictly below 1 with high probability in high-dimensional latent spaces. Fig. 5 illustrates this phenomenon concretely.

Fix a prompt $c$ and a parameter value $\theta^-$. All probabilities below are conditional on this fixed prompt and policy. For each dimension $d$, let $\mathbb{Q}_d$ be the proposal path measure in Appx. C.1, and let $\mathbf{w}_r$ be its $d$-dimensional Brownian motion in sampling time $r = 1 - t$. Denote

$$\Delta \mathbf{v}_d(\mathbf{x}, t) := \mathbf{v}_{\theta^-}(\mathbf{x}, t, c) - \mathbf{v}_{\text{base}}(\mathbf{x}, t, c), \qquad \widetilde{w}_t := \frac{(1+\eta_t)^2}{4\eta_t} \frac{1-t}{t},$$

$$\mathbf{h}_{d,r} := \sqrt{2\widetilde{w}_{1-r}}\, \Delta \mathbf{v}_d(\mathbf{x}_{1-r}, 1-r), \qquad N_{d,s} := -\int_0^s \mathbf{h}_{d,r} \cdot \mathrm{d}\mathbf{w}_r,$$

$$\mathcal{E}_{d,s} := \frac{1}{2} \int_0^s \|\mathbf{h}_{d,r}\|^2 \, \mathrm{d}r, \qquad \mathcal{E}_d := \mathcal{E}_{d,1}.$$

Assume $\eta_t > 0$ for $t \in (0, 1)$, and $\mathcal{E}_d < \infty$ $\mathbb{Q}_d$-almost surely. Then the IS ratio is given by

$$R_d := \exp\big(M_1(\theta^-)\big) = \frac{\mathrm{d}\mathbb{P}_{\theta^-,d}}{\mathrm{d}\mathbb{Q}_d}, \qquad M_1(\theta^-) = N_{d,1} - \mathcal{E}_d, \qquad \langle N_d \rangle_s = 2\mathcal{E}_{d,s}. \tag{23}$$

Let $R_{d,s} := \exp(N_{d,s} - \mathcal{E}_{d,s})$, so that $R_{d,1} = R_d$. The proposition below shows the concentration of the IS ratio:

**Proposition 13.** *In the setup above, assume the Novikov condition $\mathbb{E}_{\mathbb{Q}_d}[e^{\mathcal{E}_d}] < \infty$. Then for every $\lambda \in [0, 1]$, the process*

$$Z_{d,s}^{(\lambda)} := \exp\big(\lambda N_{d,s} - \lambda^2 \mathcal{E}_{d,s}\big), \qquad 0 \le s \le 1,$$

*is a uniformly integrable $\mathbb{Q}_d$-martingale on $[0, 1]$, with $\mathbb{E}_{\mathbb{Q}_d}[Z_{d,s}^{(\lambda)}] = 1$ for every $s \in [0, 1]$. Furthermore, for every $a > 0$ and $\gamma \in [0, 1)$,*

$$\mathbb{Q}_d\big(R_d \ge e^{-\gamma a},\ \mathcal{E}_d \ge a\big) \le \exp\left(-\frac{(1-\gamma)^2 a}{4}\right).$$

*Consequently,*

$$\mathbb{Q}_d(R_d \ge 1) \le \mathbb{Q}_d(\mathcal{E}_d < a) + e^{-a/4}. \tag{24}$$

***Proof.*** Fix $\lambda \in [0,1]$. The quadratic variation of $\lambda N_d$ is $\langle \lambda N_d \rangle_s = \lambda^2 \langle N_d \rangle_s$. Consequently,

$$\mathbb{E}_{\mathbb{Q}_d}\left[\exp\left(\frac{1}{2}\langle \lambda N_d \rangle_1\right)\right] = \mathbb{E}_{\mathbb{Q}_d}[e^{\lambda^2 \mathcal{E}_d}] \le \mathbb{E}_{\mathbb{Q}_d}[e^{\mathcal{E}_d}] < \infty.$$

Thus $Z_d^{(\lambda)}$ is a uniformly integrable martingale on $[0,1]$. Since $Z_{d,0}^{(\lambda)} = 1$, its expectation equals one at every time. Its relation to the original importance ratio is

$$Z_{d,s}^{(\lambda)} = R_{d,s}^{\lambda} \exp(\lambda(1-\lambda)\mathcal{E}_{d,s}).$$

On $F := \{R_d \ge e^{-\gamma a},\, \mathcal{E}_d \ge a\}$, Eq. (23) gives

$$\begin{aligned} N_{d,1} &\ge \mathcal{E}_d - \gamma a \ge (1-\gamma)\mathcal{E}_d, \\ \log Z_{d,1}^{(\lambda)} &\ge \big(\lambda(1-\gamma) - \lambda^2\big)\mathcal{E}_d \ge \big(\lambda(1-\gamma) - \lambda^2\big)a, \quad \forall \lambda \in [0,1]. \end{aligned}$$

Therefore

$$\mathbb{Q}_d(F) \le e^{-\left(\lambda(1-\gamma)-\lambda^2\right)a}\mathbb{E}_{\mathbb{Q}_d}\left[Z_{d,1}^{(\lambda)}\mathbf{1}_F\right] \le e^{-\left(\lambda(1-\gamma)-\lambda^2\right)a}\mathbb{E}_{\mathbb{Q}_d}[Z_{d,1}^{(\lambda)}] = e^{-\left(\lambda(1-\gamma)-\lambda^2\right)a}.$$

Take $\lambda = (1-\gamma)/2$ maximizes $\lambda(1-\gamma) - \lambda^2$ over $[0,1]$, we have

$$\mathbb{Q}_d(F) \le \exp\left(-\frac{(1-\gamma)^2 a}{4}\right).$$

Taking $\gamma = 0$ and splitting according to whether $\mathcal{E}_d < a$ proves Eq. (24).

□

**Corollary 14** (High-dimensional suppression of the importance ratio)**.** *Under the assumptions of Prop. 13, suppose there exist a constant $\kappa > 0$ independent of $d$ and a sequence $\delta_d \to 0$ such that*

$$\mathbb{Q}_d\left(\int_0^1 \widetilde{w}_t \|\Delta \mathbf{v}_d(\mathbf{x}_t, t)\|^2 \,\mathrm{d}t \ge \kappa d\right) = \mathbb{Q}_d(\mathcal{E}_d \ge \kappa d) \ge 1 - \delta_d. \tag{25}$$

*Then*

$$\mathbb{Q}_d\big(\exp(M_1(\theta^-)) < 1\big) \ge 1 - \delta_d - e^{-\kappa d/4} \longrightarrow 1.$$

***Proof.*** Apply Prop. 13 with $a = \kappa d$, and use Eq. (25) to bound the event $\{\mathcal{E}_d < \kappa d\}$. □

# F Related Work

**Reverse-sampling-based RL.** DDPO [5] and DPOK [12] pioneered policy-gradient fine-tuning for DDPM-style models by treating reverse denoising as a finite-horizon MDP. Flow-GRPO [24] and Dance-GRPO [41] extend this paradigm to flow-matching models by converting the sampling ODE into a marginally equivalent SDE and applying clipped GRPO-style updates. Subsequent methods refine this recipe along different axes: GRPO-Guard [37] regulates and rescales per-step likelihood ratios, TempFlow-GRPO [14] branches trajectories at selected timesteps, and MixGRPO [22] confines stochastic exploration to a sliding time window. FDFO [28] follows a different route and estimates improvement directions through finite differences rather than analytic policy gradients. Our framework disentangles the estimator, weighting, and sampling choices underlying these methods and explains their efficiency differences through the resulting value-gradient variance and scale.

Complementary analyses also seek to organize this rapidly growing design space. Choi et al. [8] separate the policy-gradient objective, likelihood estimator, and sampler, and identify likelihood estimation as the dominant empirical factor. We refine the objective axis further by decomposing it into value-gradient estimation and the associated on- and off-policy weights. The concurrent work Reward Score Matching [20] develops a score-space view that connects several reward-based fine-tuning objectives; in contrast, our path-space formulation explicitly recovers the training structures of AWM and DiffusionNFT and relates them to reverse-sampling policy gradients.

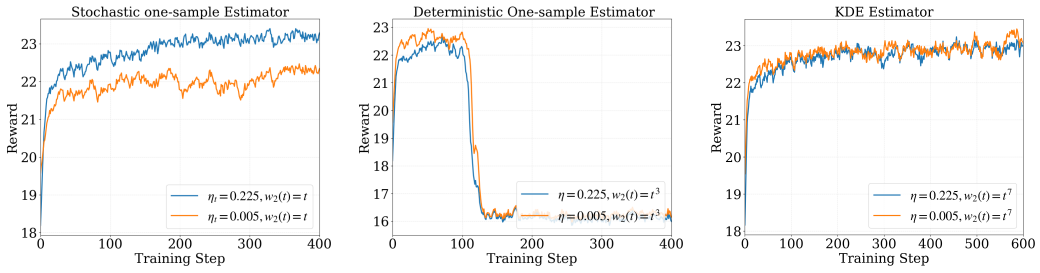
**Forward-matching RL methods.** An orthogonal line retains the analytic forward-noising process and injects reward information into a pretraining-style regression loss. Reward-Weighted Regression (RWR) [21] weights the diffusion training objective by endpoint rewards. AWM [40] and DiffusionNFT [45] similarly avoid the reverse-process MDP and construct reward- or advantage-dependent flow-matching targets from sampled clean endpoints and their analytically noised states. These methods report substantial improvements in training efficiency, but their relationship to reverse-sampling policy gradients and the role of their specialized timestep weights were previously unclear. Our path-space derivation shows that they instantiate the same policy-gradient structure with a deterministic value-gradient estimator and different weight choices.

**Stochastic-control and adjoint-based methods.** A complementary line formulates KL-regularized reward fine-tuning as entropy-regularized stochastic optimal control. ELEGANT [35] develops this viewpoint for continuous-time diffusion models, while Adjoint Matching [10] introduces a memoryless noise schedule and turns the Bellman optimality condition into an on-policy regression of the value gradient; its pathwise estimator, however, requires reward gradients and a backward adjoint sweep along an SDE trajectory. Tilt Matching [30] instead relates the original and reward-tilted velocity fields through a regression objective and avoids both reward gradients and trajectory backpropagation by annealing through intermediate tilted models. Most closely related to our work, the concurrent Reinforce Adjoint Matching (RAM) method [3] takes a stochastic optimal control viewpoint, identifies the value gradient as the optimal adjoint, and combines the resulting fixed point with a REINFORCE identity to derive a forward-matching-type loss. In contrast, we derive the value-gradient form from path-space policy-gradient variance reduction, yielding a unified framework for both GRPO-style reverse-sampling algorithms and forward-based methods such as AWM and DiffusionNFT. Our framework further establishes an estimator-specific optimal weight mapping and introduces a new multi-sample KDE value-gradient estimator in the final recipe.

**Reward-gradient and offline preference methods.** When the reward is differentiable, it can be optimized directly by backpropagating through all or part of the sampling trajectory [9, 31, 39]. This approach provides a direct training signal but requires reward gradients and incurs substantial memory and computation costs when differentiation spans many denoising steps. A separate offline line adapts direct preference optimization to diffusion models, learning from paired human-preference data without online reward evaluation [36, 42, 43]. These methods operate under a different feedback regime from the black-box online rewards considered in our framework.

# G Experimental Details

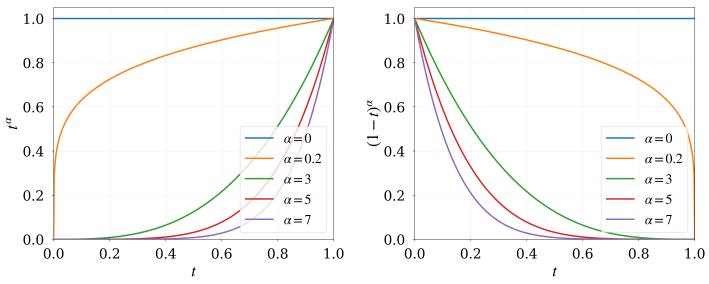
## G.1 Additional Results




(a) (b) (c)

Figure 6: Sampling noise schedule ablations for the stochastic one-sample estimator, deterministic one-sample estimator, and our KDE estimator. For each estimator, we run experiments for sampler noise schedule $\eta_t \in \{0.005, 0.225\}$ on the Pickscore reward.

In Fig. 6, we show the sampling noise schedule ablations for each estimator. We find that the stochastic estimator in GRPO-based methods performs better in the larger $\eta_t$ setting (Fig. 6 (a)), while the deterministic estimator suffers from training instability in both settings, and the small noise scheduler performs slightly better (Fig. 6 (b)). Our estimator is not sensitive to different sampling noise scales and is stable across all these settings.




(a) (b)

Figure 7: Visualization of different weights

In Fig. 7, we provide a visualization of different weight functions.

### G.2 Hyperparameters

We summarize the hyperparameters of our experiments in Table 2. In all the experiments, we disable CFG and use the batch-level advantage normalization.

Table 2: Hyperparameters used throughout Sec. 4.

| Hyperparameter | Value |
|---|---|
| Group size $G$ | 24 for SD 3.5, 16 for Qwen-Image |
| Train denoising steps $T$ | 10 |
| Inference denoising steps | 40 for SD 3.5, 50 for Qwen-Image |
| Rollout SDE noise schedule | 0.225 for stochastic estimator, 0.005 for deterministic and KDE estimator |
| Optimizer | AdamW, $\mathrm{lr} = 3\times10^{-4}$, $\beta_1=0.9$, $\beta_2=0.999$ |
| KL coefficient $\beta$ | $10^{-4}$ |
| Batch size | 9 for SD 3.5, 4 for Qwen-Image |
| Number of batches per epoch | 16 for SD 3.5, 4 for Qwen-Image |
| Train CFG | Disable CFG in training |
| $h$ in Eq. (10) | $d/2000$, where $d$ is the latent dimension |

### G.3 Robustness across random seeds

We repeat the SD3.5-Medium experiment with PickScore reward using three random seeds while keeping all other configurations fixed. Table 3 reports the mean and standard deviation at four matched training checkpoints.

Table 3: PickScore across three random seeds (mean $\pm$ standard deviation). Our method achieves a higher mean at every checkpoint with small run-to-run variation.

| Training time (GPU hours) | 18.58 | 36.44 | 54.28 | 72.12 |
|---|---|---|---|---|
| DiffusionNFT | $22.85 \pm 0.02886$ | $23.06 \pm 0.01618$ | $23.16 \pm 0.02153$ | $23.22 \pm 0.01274$ |
| Ours | $22.95 \pm 0.05240$ | $23.28 \pm 0.02239$ | $23.48 \pm 0.005547$ | $23.56 \pm 0.01461$ |

At the final checkpoint, our method obtains $23.56 \pm 0.01461$, corresponding to a relative standard deviation of $0.062\%$; DiffusionNFT obtains $23.22 \pm 0.01274$ with a relative standard deviation of $0.055\%$. The consistent gap across checkpoints and seeds indicates that the improvement is not driven by a favorable random seed.

### G.4 Runtime comparison

The KDE estimator computes pairwise distances in $O(G^2d)$ time. With $G = 24$ and latent dimension $d = 65{,}536$, this is on the order of $10^7$ arithmetic operations per timestep and is efficiently parallelized on GPUs. It also reuses cached rollout latents, velocities, and rewards, so it requires no additional

denoiser evaluation. Under matched settings, our update uses a trainable-policy forward and, when KL regularization is enabled, a frozen-reference forward. DiffusionNFT additionally evaluates the old policy at the renoised latent, requiring three rather than two denoiser forwards per optimized timestep. Since one denoiser forward costs approximately $10^{12}$ FLOPs, this saved network evaluation dominates the pairwise-distance overhead.

Table 4: Measured wall-clock training time per epoch under matched SD3.5-Medium settings.

| Method | Training time per epoch (seconds) |
|---|---|
| DiffusionNFT | 115 |
| KDE estimator (ours) | 97 |

As shown in Table 4, our method saves 18 seconds per epoch, a $15.7\%$ wall-clock reduction. Thus, the $O(G^2 d)$ KDE computation has little effect on end-to-end runtime, while eliminating one denoiser forward pass provides a pronounced speedup.

## H Qualitative Examples

The visualization grids (Fig. 8, Fig. 9,Fig. 10) contain prompts from the test set of Pickscore reward training. For each prompt, images are generated with the same inference configuration from SD3.5-Medium after the RL training with GRPO-Guard, DiffusionNFT, and our method. The rows are organized by prompt split and the columns by method, enabling direct inspection of prompt alignment, perceptual quality, excessive saturation, and repetitive reward-hacking patterns without selecting examples solely by reward.

As a complementary quantitative check, a panel of eight evaluators assesses randomly generated images drawn from a pool of 30 prompts spanning the same three splits. Evaluators are blinded to method identity and judge overall visual quality, prompt alignment, and aesthetic preference. Aggregating the evaluation decisions according to this protocol yields method-level preference shares of $51.4\%$ for our method, $29.4\%$ for GRPO-Guard, and $19.2\%$ for DiffusionNFT. These results suggest that the reward gains do not merely reflect numerical over-optimization.

## I Limitations

Our experiments are conducted on SD3.5-M and Qwen-Image with a fixed group size $G = 24$ and a limited set of reward functions (PickScore, OCR, and GenEval). While these cover a representative range of reward types, the generalizability of our findings to other model architectures (e.g., video diffusion models), reward functions (*e.g.*, human-preference models or verifiable task rewards) has not been systematically evaluated.

## J Social Impact

This work proposes a unified theoretical framework for reinforcement learning post-training of diffusion generative models, which can facilitate the alignment of generative models with human preferences and safety constraints, thereby contributing to the responsible development of generative AI systems. However, this work mainly focuses on theory and algorithms, so the work does not have a broader social impact.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We summarize our contribution in the introduction with clear support in the specific section.

   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: We have added a section on limitations in the appendix.

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: We provide a detailed proof in Appx. C

   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: We show the experimental details in Appx. G

   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: We will open-source the code after the acception of the paper.

   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) necessary to understand the results?

   Answer: [Yes]

   Justification: We show the experimental details in Appx. G.

   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: We show the experiments on various rewards and base models.

   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: We mention it in Appx. G.

   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 research conducted in the paper conforms with 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: We discuss the social impact in Appx. J.

    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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    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: [N/A]

    Justification: The paper does not use existing assets.

    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: [N/A]

    Justification: the paper does not release new assets.

    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 paper does not involve crowdsourcing nor research with human subjects.

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 nor research with human subjects.

    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: We only use LLM to polish writing.

    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.

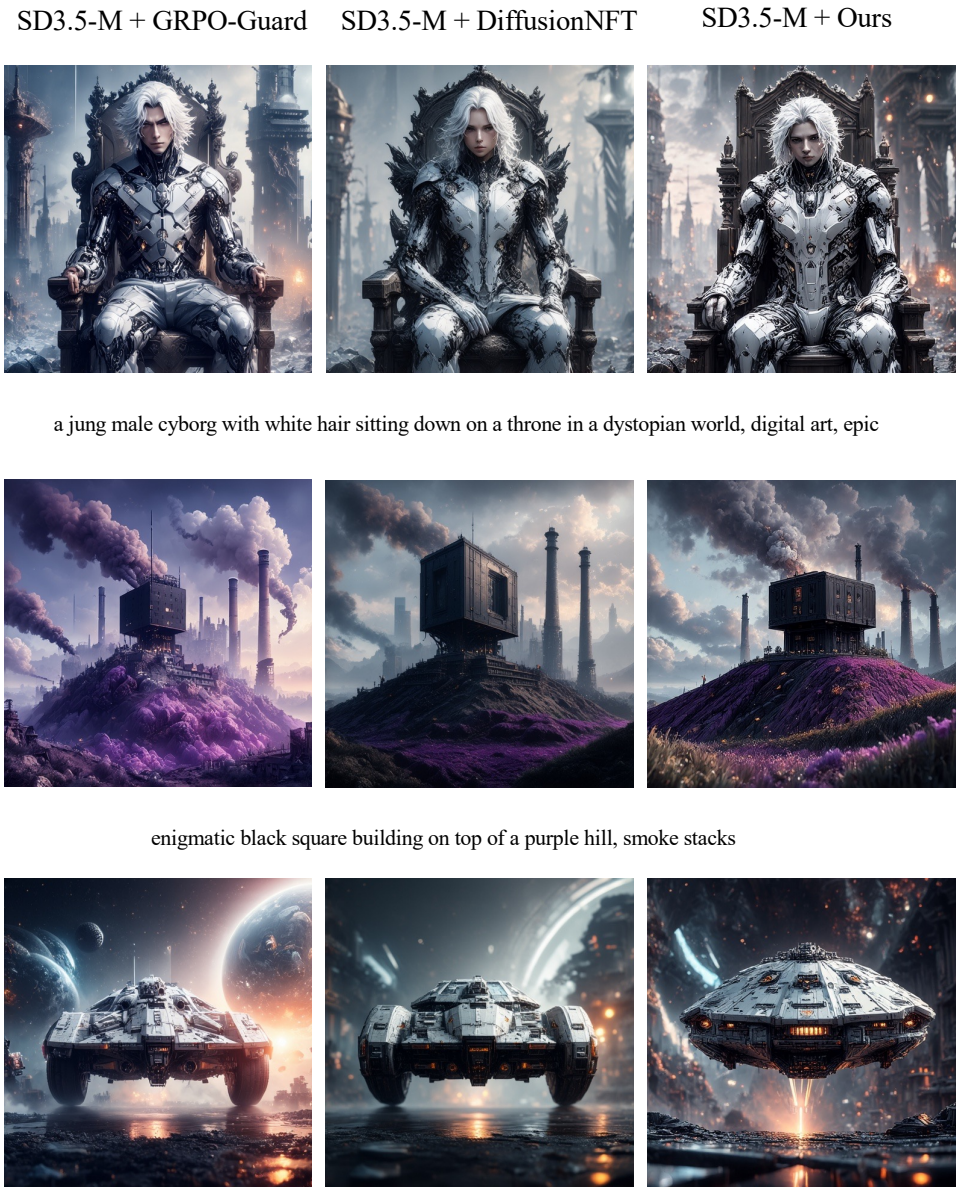



Figure 8: Qualitative comparison across GRPO-Guard, DiffusionNFT and our method.

SD3.5-M + GRPO-Guard SD3.5-M + DiffusionNFT SD3.5-M + Ours



photo of Ford Focus RS, night time, city, city roads, Miami streets



Labrador in anime style, black and white photography, highly detailed,



Portrait frame Penthouse Magazine, expressive and deep symmetrical eyes, stunning face, naive and innocent facial expression, stunningly beautiful slender girl, shaggy haircut, white hair color, in a sci-fi tight-fitting jumpsuit, the girl looks up into the camera, against the background of a white sci-fi room, stunning quality, ultra-high resolution, photorealistic, 8k, hyperrealism, rembrandt lighting

Figure 9: Qualitative comparison across GRPO-Guard, DiffusionNFT and our method.




Figure 10: Qualitative comparison across GRPO-Guard, DiffusionNFT and our method.