STEEPEST GUIDANCE: A PRACTICAL AND PRINCIPLED APPROACH TO INFERENCE-TIME ALIGNMENT OF FLOW AND DIFFUSION-BASED MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time alignment of flow and diffusion-based models is critical for achieving flexible generative modeling. Theoretically, Doob's h-transform provides an elegant solution to this problem, and most existing methods are based on this principle. However, in practice, estimating the optimal guidance derived from Doob's h-transform at inference time is challenging. To deal with this issue, we regard inference-time alignment as a sequential optimization problem in the space of probability measures and propose a novel framework called *Steepest Guidance*, based on the principle of maximizing local improvement in the objective. We provide a theoretical analysis of the proposed method and demonstrate its effectiveness through extensive experiments.

1 INTRODUCTION

Flow and diffusion-based generative models have emerged as powerful tools for modeling complex data distributions across various domains, including image generation (Ho et al., 2020; Lipman et al., 2022), natural language processing (Li et al., 2022), and molecular design (Hoogeboom et al., 2022). In many applications, we often aim to optimize some criteria over the generated samples. For instance, in image generation, we may want to generate images that not only look realistic but also satisfy certain aesthetic criteria or exhibit diversity. This can be formalized as an optimization problem on a reward functional $R[\mu]$ over the distribution of generated samples μ. Typically, the reward functional is defined as the expected value of a reward function $r : \mathbb{R}^d \to \mathbb{R}$ over the distribution μ, i.e., $R[\mu] = \int r(y)\mu(dy)$. However, in some applications, we may want to consider more complex rewards such as diversity-promoting objectives (Corso et al., 2023; Vinograd et al., 2026; Santi et al., 2025) or risk-sensitive objectives (Zhang et al., 2020; Santi et al., 2025; Wang et al., 2026), which cannot be expressed as the expected value of a reward function since they are non-linear in μ.

Reward-guided generation is often formulated as an optimal control problem (Domingo-Enrich et al., 2024), where the generative model is treated as a stochastic process, and the reward serves as a control objective. Theoretically, the optimal solution to this problem is given by Doob's h-transform (Rogers & Williams, 2000), which provides a principled way to modify the generative process to maximize the expected reward, while ensuring that the generated samples remain close to the original data distribution. Several methods (Domingo-Enrich et al., 2024; Uehara et al., 2024) have been proposed to address this problem by learning a guidance model that steers the generative process towards high-reward regions of the sample space. In addition, Marion et al. (2024); Kawata et al. (2025); Santi et al. (2025) have developed a general framework for fine-tuning diffusion and flow models to maximize arbitrary reward functionals including non-linear ones.

On the other hand, a line of work has developed training-free approaches, where the guidance is estimated during the inference phase without additional training. The plug-in estimator (Dandapanthula & Boffi, 2026) computes the optimal guidance via backpropagation through the generative process, which requires the reward and sampling process to be differentiable. Recently, to deal with non-differentiable rewards, Zhu et al. (2026) have proposed REINFORCE (Williams, 1992) estimators for the optimal guidance.

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

In spite of the intensive research on training-free inference-time alignment of flow and diffusion models, there are still several challenges that remain to be addressed. First, most existing methods can only handle linear reward functionals, and cannot be applied to non-linear reward functionals. Second, even if the reward functional is linear, the optimal guidance is difficult to estimate in practice. Recently, Dandapanthula & Boffi (2026) have revealed that the plug-in estimator with finite Monte Carlo samples is biased, leading to suboptimal performance in reward-guided generation. In addition, as we show later, REINFORCE estimators also suffer from large bias, which causes the performance of inference-time alignment of flow models to degrade significantly.

In this paper, we regard the inference-time alignment of flow and diffusion models as a sequential optimization problem in the space of probability measures, and propose a novel method called *Steepest Guidance* based on the local improvement of the reward functional. For linear reward functionals, we show that it can be estimated in an unbiased manner, leading to improved performance in inference-time alignment. In addition, steepest guidance can be naturally applied to non-linear reward functionals through a particle approximation. Our contributions are summarized as follows:

- We regard the inference-time alignment of flow and diffusion models as a sequential optimization problem in the space of probability measures, and propose *Steepest Guidance*, which guides the generative process towards the steepest ascent direction of the reward functional. Furthermore, we generalize it to an entropy-regularized objective and develop *Regularized Steepest Guidance*.
- Accounting for the evolution of the objective, we establish provable reward improvement and global convergence of our proposed method under suitable assumptions. This can be seen as a functional extension of the existing analysis of Wasserstein gradient flows (Bakry et al., 2014; Chizat, 2022; Nitanda et al., 2022) from fixed-objective optimization to the optimization of objectives that evolve along the generative process.
- Through extensive experiments, we demonstrate that the proposed method consistently outperforms existing approaches and successfully handles non-linear reward functionals.

1.1 Related Work

Here, we briefly review the related work and refer to Appendix A for a more detailed discussion.

Inference-time Alignment of Flow and Diffusion Models. Inference-time alignment of flow and diffusion models can be categorized into two main approaches: selection-based methods and guidance-based methods. Selection-based methods include sequential Monte Carlo (Wu et al., 2023), SVDD (Li et al., 2024), and Best-of-N (Nakano et al., 2021), which utilize multiple particles and select or resample them to obtain better samples. Guidance-based methods include DOIT (Zhu et al., 2026) and Gradient Guidance (Guo et al., 2024), which are often efficient since they can utilize gradient information of the reward function. However, they require computationally expensive backpropagation or suffer from large bias.

Optimization in the Space of Probability Measures. Several works (Marion et al., 2024; Kawata et al., 2025; Santi et al., 2025) have regarded reward-guided generation as an optimization problem in the space of probability measures and developed fine-tuning methods for general reward functionals, which require additional training. On the other hand, (Mean-field) Langevin dynamics (Welling & Teh, 2011; Chizat, 2022; Nitanda et al., 2022) is based on a Wasserstein gradient flow and optimizes a functional over the space of probability measures through gradient-based particle updates. However, the distribution of the real-world data is often complex and multimodal, which hinders the convergence of such methods. Recently, Slowly Annealed Langevin Dynamics (SALD) (Nitanda et al., 2026) has been applied to inference-time alignment of flow and diffusion models, but it cannot be applied to non-linear reward functionals.

2 Preliminaries

In this section, we introduce the flow and diffusion-based generative models and formulate the inference-time alignment as an optimal control problem. For $d \in \mathbb{N}$, let $\mathcal{P}$ be the space of probability measures on $\mathbb{R}^d$ which have density functions, and finite entropy and second moment. We denote a data distribution by $\pi_1 \in \mathcal{P}$.

2.1 Flow and Diffusion-based Models

A flow matching model is formulated as the following ordinary differential equation (ODE):

$$\mathrm{d}Y_t = v_t(Y_t)\,\mathrm{d}t, \quad Y_0 \sim \mathcal{N}(0, I),$$

where $v_t(Y_t)$ is a time-dependent vector field, which is approximated by a neural network. Typically, $v_t(x)$ is defined as $\mathbb{E}\left[\frac{\mathrm{d}Y_t}{\mathrm{d}t} \mid Y_t = x\right]$, where $Y_t := (1-t)Y_0 + tY_1$, $Y_0 \sim \mathcal{N}(0, I)$, and $Y_1 \sim \pi_1$. Furthermore, for any $\sigma_t \in \mathbb{R}$, the solution of the SDE

$$\mathrm{d}Y_t = \left(v_t(Y_t) + \frac{\sigma_t^2}{2}\nabla \log \pi_t(Y_t)\right)\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \quad Y_0 \sim \mathcal{N}(0, I) \tag{1}$$

has the same marginal distribution as the solution of the ODE, where π_t is the marginal distribution of Y_t. In this paper, we consider a memoryless noise schedule $\sigma_t^2 = 2(1-t)/t$ following Domingo-Enrich et al. (2024); Bergmeister et al. (2026).

On the other hand, for diffusion models, we first define the following (forward) SDE (Jiao et al., 2025):

$$\mathrm{d}X_t = -\frac{1}{2(1-t)}X_t\,\mathrm{d}t + \frac{1}{\sqrt{1-t}}\,\mathrm{d}W_t, \quad X_0 \sim \pi_1.$$

Then, the reverse-time SDE is given by

$$\mathrm{d}Y_t = \left(\frac{1}{2}Y_t + \nabla \log \pi_t(Y_t)\right)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\,\mathrm{d}W_t, \quad Y_0 \sim \mathcal{N}(0, I),$$

where Y_t has the same marginal distribution as X_{1-t}. Here, $\nabla \log \pi_t(y)$ is the score function of π_t and is approximated by a neural network.

In this paper, we consider the following SDE as a general form of flow and diffusion models:

$$\mathrm{d}Y_t = b_t(Y_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \quad Y_\varepsilon \sim \pi_\varepsilon, \quad t \in [\varepsilon, 1].$$

Here, we introduce a small constant $\varepsilon \geq 0$ and assume that $\varepsilon > 0$ for theoretical analysis to avoid blow-up of σ_t at $t = 0$. We denote its path measure by $\mathbb{P}_\pi$ and the marginal distribution of Y_t by π_t. In the following, we assume that the second moment of π_t is finite.

2.2 Inference-time Alignment of Flow Models

In this paper, we aim to maximize a reward functional $R : \mathcal{P} \to \mathbb{R}$ over the generated samples by adding a guidance term $g_t(y)$ to the generative process:

$$\mathrm{d}Y_t = (b_t(Y_t) + g_t(Y_t))\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \quad Y_\varepsilon \sim \pi_\varepsilon, \quad t \in [\varepsilon, 1].$$

This guidance based formula has been widely employed by several existing works such as classifier/classifier-free guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022). We call a guidance $g_t(y)$ admissible if it satisfies $\int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\|g_t(Y_t)\|^2/\sigma_t^2\right]\mathrm{d}t < \infty$. We denote its path measure by $\mathbb{P}_{\pi^g}$ and the marginal distribution of Y_t by π_t^g. Then, we consider the following optimization problem:

$$\max_g R[\pi_1^g] - \frac{1}{\lambda}\mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi), \tag{2}$$

where KL is the KL divergence and $\lambda > 0$ is a regularization parameter. Typically, the reward functional is defined as the expected value of a reward function $r : \mathbb{R}^d \to \mathbb{R}$ over the distribution μ, i.e., $R[\mu] = \int r(y)\mu(dy)$. On the other hand, in some applications, we may want to consider more complex rewards such as risk-averse objectives like CVaR or diversity-promoting objectives like Rao's quadratic entropy:

$$R_{\mathrm{CVaR}}[\mu] = \mathbb{E}_\mu\left[r(Y) \mid r(Y) \leq q_\alpha\right], \quad R_{\mathrm{Rao}}[\mu] = -\frac{1}{2}\int\int k(y, y')\mu(dy)\mu(dy'),$$

where q_α is the α-quantile of $r(Y)$ and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a positive definite kernel. See Table 3 for other examples of (non-linear) reward functionals.

In this paper, we assume that the reward functional R has a functional derivative $\frac{\delta R}{\delta \mu}[\mu](y)$ with $\|\nabla \frac{\delta R}{\delta \mu}[\mu](y)\|, |\frac{\delta R}{\delta \mu}[\mu](y)| \leq C_V$ and $\mu_1^* := \mathrm{argmax}_{\mu_1 \in \mathcal{P}} R[\mu_1] - \frac{1}{\lambda}\mathrm{KL}(\mu_1 \mid \pi_1)$ exists.

Definition 2.1 (Functional Derivative)**.** The functional derivative of a reward functional $R : \mathcal{P} \to \mathbb{R}$ with respect to a measure $\mu \in \mathcal{P}$ is a function $\frac{\delta R}{\delta \mu}[\mu](y)$ such that for any $\nu \in \mathcal{P}$ and the mixture path $\mu_\epsilon := (1-\epsilon)\mu + \epsilon\nu$, it holds that $\frac{\mathrm{d}}{\mathrm{d}\epsilon} R[\mu_\epsilon]\Big|_{\epsilon=0} = \int \frac{\delta R}{\delta \mu}[\mu](y)\,(\nu - \mu)(\mathrm{d}y)$.

2.3 Optimal Control via Doob's h-transform

As a theoretical principle to determine the guidance for the reward maximization, Doob's h-transform has been utilized by several existing works (Uehara et al., 2024; Domingo-Enrich et al., 2024; Kawata et al., 2025; Bergmeister et al., 2026; Zhu et al., 2026) to guide the generative process. Formally, Doob's h-function is defined as $h_t(y) = \mathbb{E}\left[\exp(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](Y_1)) \mid Y_t = y\right]$, and also it can be defined through the following SDE in the sense that the density ratio between the marginal distributions of Y_t for the original process and the following process is proportional to h_t:

$$\mathrm{d}Y_t = (b_t(Y_t) + g_t^*(Y_t))\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \quad Y_0 \sim \mathcal{N}(0, I),$$

where $g_t^*(y) = \sigma_t^2 \nabla \log h_t(y)$. As discussed in Domingo-Enrich et al. (2024), under the memoryless noise schedule this Doob transform solves the control problem (2) with $\varepsilon = 0$.

Unfortunately, Doob's h-transform does not admit a closed-form expression and requires numerical estimation, presenting a significant computational challenge. To tackle this problem, some recent works have proposed to estimate the optimal guidance $g_t^*(y)$ at inference time using Monte Carlo samples from the conditional distribution $\pi_1(\cdot \mid Y_t = y)$. From the chain rule, the optimal guidance $g_t^*(y)$ can be expressed as

$$g_t^*(y) = \sigma_t^2 \frac{\nabla_y \mathbb{E}_{\pi_1(\cdot|Y_t=y)}\left[\exp(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](Y_1))\right]}{\mathbb{E}_{\pi_1(\cdot|Y_t=y)}\left[\exp(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](Y_1))\right]}.$$

Let $z_1, \ldots, z_k$ be i.i.d. samples from the conditional distribution $\pi_1(Z \mid Y_t = y)$. Then, the denominator can be estimated as $\frac{1}{k}\sum_{i=1}^k \exp(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](z_i))$. To estimate the numerator, there are two main approaches: Plug-in and REINFORCE estimators.

Plug-in Estimator. Assume that the sampling process can be expressed as $z_i = f(y, \epsilon_i)$ for some deterministic function f and random variable ϵ_i. Then, the plug-in estimator of $g_t^*(y)$ is given by

$$\nabla_y \mathbb{E}_{\pi_1(\cdot|Y_t=y)}\left[\exp\left(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](Y_1)\right)\right] \simeq \frac{1}{k}\sum_{i=1}^k \nabla_y \exp\left(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](f(y, \epsilon_i))\right).$$

Note that the plug-in estimator requires the reward and sampling process to be differentiable. Furthermore, even if the reward and sampling process are differentiable, computing the gradient is computationally expensive since it requires backpropagation through the generative process.

REINFORCE Estimator. Following Zhu et al. (2026), let us consider the following identity (Williams, 1992):

$$\nabla_y \mathbb{E}_{\pi_1(\cdot|Y_t=y)}\left[f(Y_1)\right] = \mathbb{E}_{\pi_1(\cdot|Y_t=y)}\left[f(Y_1)\,\nabla_y \log \pi_1(Y_1 \mid Y_t = y)\right].$$

Then, the REINFORCE estimator of $g_t^*(y)$ is given by

$$\nabla_y \mathbb{E}_{\pi_1(\cdot|Y_t=y)}\left[\exp\left(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](Y_1)\right)\right] \simeq \frac{1}{k}\sum_{i=1}^k \exp\left(\lambda \frac{\delta R}{\delta \mu}[\mu_1^*](z_i)\right) \nabla_y \log \pi_1(z_i \mid Y_t = y).$$

Instead of the differentiability of the reward and sampling process, the REINFORCE estimator only requires the differentiability of the log-likelihood of the conditional distribution $\pi_1(\cdot \mid Y_t = y)$. Therefore, the REINFORCE estimator can be applied to non-differentiable reward functions. As shown in the following lemma, we can compute the gradient of the log-likelihood of the conditional distribution $\pi_1(\cdot \mid Y_t = y)$ from the score function of the marginal distribution π_t.

Lemma 2.2. *For any $t \in [\varepsilon, 1)$, we have*

$$\nabla_y \log \pi_1(z \mid Y_t = y) = \begin{cases} -\frac{1}{(1-t)^2}(y - tz) - \frac{t v_t(y) - y}{1-t} & (\textit{Flow Matching Model}), \\ -\frac{1}{1-t} \cdot (y - \sqrt{t}z) - \nabla \log \pi_t(y) & (\textit{Diffusion Model}). \end{cases}$$

See Appendix E for the proof.

Challenges in Estimating the Optimal Guidance. While Doob's h-transform provides a principled way to compute the optimal guidance, there are two main challenges in computing the optimal guidance $g_t^*(y)$.

Non-linearity of the reward functional: First, while $\frac{\delta R}{\delta \mu}[\mu_1^*](y) = r(y)$ for linear reward functionals, we cannot compute $\frac{\delta R}{\delta \mu}[\mu_1^*](y)$ for general non-linear reward functionals because it requires the knowledge of the optimal distribution μ_1^*, which is exactly what we are looking for.

Non-linearity of the guidance: Second, even if we can compute $\frac{\delta R}{\delta \mu}[\mu_1^*]$, we cannot obtain unbiased estimates of $g_t^*(y)$ since a non-linear transformation (logarithm) is applied to the expectation. Therefore, if we use a finite number of samples, the estimator of $g_t^*(y)$ is biased. For the plug-in estimator, Dandapanthula & Boffi (2026) have shown that the bias of the plug-in estimator leads to suboptimal performance in reward-guided generation. Here, we show that the REINFORCE estimator may be biased even in the case where the plug-in estimator is unbiased. Specifically, we consider a simple toy problem, where the terminal distribution is the one-dimensional Gaussian and the reward function is $r(y) = y$. In this case, the plug-in estimator is unbiased but the REINFORCE estimator suffers from large bias as shown in Fig. 1. See Appendix L for detailed experimental settings.

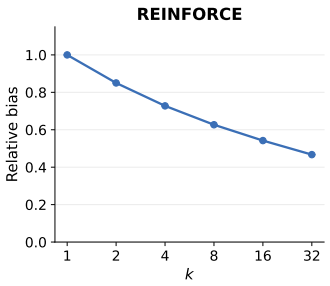

Figure 1: Relative bias of the REINFORCE estimators for the toy Gaussian problem.

3 Proposed Method: Steepest Guidance

As shown in the previous section, the optimal guidance $g_t^*(y)$ is difficult to estimate in practice due to the non-linearity of the reward functional and the guidance. Instead of estimating the globally optimal guidance, we propose to consider the local optimality of the guidance and design a *steepest guidance* that improves the reward functional in a local manner. The following proposition is fundamental to analyzing the improvement of the reward functional with respect to the guidance.

Proposition 3.1. *For any admissible guidance $g_t(y)$, we have*

$$R[\pi_1^g] - R[\pi_1] = \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[g_t(Y_t)\cdot\nabla\frac{\delta V(t,\pi_t^g)}{\delta\mu}(Y_t)\right]\mathrm{d}t,\ \mathrm{KL}(\mathbb{P}_{\pi^g}\mid\mathbb{P}_\pi) = \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\frac{\|g_t(Y_t)\|^2}{2\sigma_t^2}\right]\mathrm{d}t,$$

where $V(t,\mu) := R[K_t\mu]$ and K_t is the transition kernel from t to 1.

See Appendix F for the proof. We observe that this proposition provides the *steepest direction* to improve the objective at each time t, and that the functional derivative $\frac{\delta V(t,\pi_t^g)}{\delta\mu}(Y_t)$ directly appears in the expectation without any non-linear transformation, in contrast to Doob's h-transform. Similar results can be found for linear functionals in Jiao et al. (2025) but we extend them to general reward functionals through a generator-based argument that extends to non-linear functionals.

3.1 Local Improvement of the Reward Functional

Here, we explain the intuitive idea behind our proposed method. Let μ_τ be the marginal distribution of Y_τ at time $\tau \in [\varepsilon, 1)$. Then, we consider the following SDE, in which a time-independent guidance is applied over a small interval $[\tau, \tau+\Delta\tau]$ with $\tau + \Delta\tau \le 1$

$$\mathrm{d}Y_t = (b_t(Y_t) + g(Y_t))\cdot\mathrm{d}t + \sigma_t\,\mathrm{d}W_t \quad (t\in[\tau,\tau+\Delta\tau]),$$
$$\mathrm{d}Y_t = b_t(Y_t)\cdot\mathrm{d}t + \sigma_t\,\mathrm{d}W_t \quad (t\in[\tau+\Delta\tau,1]).$$

Intuitively, Proposition 3.1 implies that for a small time interval $\Delta\tau$, we have

$$R[\mu_1^g] - R[\mu_1] - \tfrac{1}{\lambda}\mathrm{KL}(\mathbb{P}_{\mu^g}\mid\mathbb{P}_\mu) \simeq \Delta\tau\cdot\mathbb{E}_{\mu_\tau}\left[g(Y_\tau)\cdot\nabla_y\frac{\delta V(\tau,\mu_\tau)}{\delta\mu}(Y_\tau) - \frac{\|g(Y_\tau)\|^2}{2\lambda\sigma_\tau^2}\right],$$

which is a quadratic function of g and can be maximized by setting $g(y) = \lambda\sigma_\tau^2\nabla_y\frac{\delta V(\tau,\mu_\tau)}{\delta\mu}(y)$.

Based on the above observation, we propose a *steepest guidance* defined as

$$g_t(y) = \lambda\sigma_t^2\nabla_y\frac{\delta V(t,\pi_t^g)}{\delta\mu}(y), \tag{3}$$

which is characterized by the local optimality of the reward improvement. The following theorem shows that the steepest guidance provably improves the reward functional while controlling the KL divergence between the original and guided processes.

Theorem 3.2. *Assume that the steepest guidance defined in Eq.* (3) *is admissible. Then, we have*

$$R[\pi_1^g] - R[\pi_1] = \lambda\Gamma_1, \quad \mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi) = \tfrac{\lambda^2\Gamma_1}{2},$$

where $\Gamma_t := \int_\varepsilon^t \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(Y_t)\right\|^2\right] dt$ *for* $t \in [\varepsilon, 1]$.

See Appendix G for the proof. We may interpret this theorem as a functional version of the descent lemma, analogous to the standard analysis of finite-dimensional gradient descent. It also suggests that a larger λ is expected to yield a greater improvement in the reward functional. Indeed, in Section 4.1, we establish global convergence for a regularized variant of steepest guidance by taking sufficiently large λ, under suitable assumptions.

3.2 Practical Implementation of Steepest Guidance

To implement the steepest guidance in practice, we need to discretize the process and estimate $g_t(y)$ from finite Monte Carlo samples. First, we focus on the case where R is linear in μ, i.e., $R[\mu] = \int r(y)\mu(dy)$ for some reward function $r : \mathbb{R}^d \to \mathbb{R}$. In this case, we have $V(t,\mu) = \int r(y_1)K_t\mu(dy_1)$ and $\frac{\delta V(t,\mu)}{\delta\mu}(y) = E[r(Y_1) \mid Y_t = y]$. Therefore, the steepest guidance can be estimated as follows:

$$\hat{g}_t^{\mathrm{steepest}}(y) = \lambda\sigma_t^2 \frac{1}{k}\sum_{i=1}^k r(y_i)\,\nabla_y \log \pi_1(y_i \mid Y_t = y), \tag{4}$$

where y_i are i.i.d. samples from the conditional distribution $\pi_1(\cdot \mid Y_t = y)$. An important property of the above estimator is that it is unbiased, i.e., $\mathbb{E}\left[\hat{g}_t^{\mathrm{steepest}}(y)\right] = g_t(y)$, which is in contrast to the case of the optimal guidance.

Remark 3.3. In practice, it is often computationally expensive to sample from the conditional distribution $\pi_1(\cdot \mid Y_t = y)$. Thus, previous works have proposed to use the approximate sampling processes such as GLASS flow (Holderrieth et al., 2025) and Diamond Map (Holderrieth et al., 2026). Under the approximate sampling process, the estimator in Eq. (4) is biased but can still be effective in practice, as shown in our experiments.

In the case where R is non-linear in μ, the population first variation $\frac{\delta R}{\delta\mu}[\mu_{1,t}](y)$ depends on the unknown terminal distribution $\mu_{1,t} := K_t\pi_t^g$. Following previous work (Takakura et al., 2026), we approximate this first variation using lookahead particles. Let $\{y^{(j)}\}_{j=1}^N$ be samples in a batch. Then, for each $j = 1, \ldots, N$, we generate k i.i.d. samples $\{y_i^{(j)}\}_{i=1}^k$ from the conditional distribution $\pi_1(\cdot \mid Y_t = y^{(j)})$ and define the empirical measure $\hat{\mu}_1 = \frac{1}{Nk}\sum_{i=1}^k \sum_{j=1}^N \delta_{y_i^{(j)}}$. With a slight abuse of notation, we write $\frac{\delta R}{\delta\mu}[\hat{\mu}_1](y)$ for the sample-based plug-in approximation obtained by replacing the population quantities in $\frac{\delta R}{\delta\mu}[\mu_{1,t}](y)$ with their empirical counterparts. We then estimate the steepest guidance by

$$\hat{g}_t^{\mathrm{steepest}}(y^{(j)}) = \lambda\sigma_t^2 \frac{1}{k}\sum_{i=1}^k \frac{\delta R}{\delta\mu}[\hat{\mu}_1](y_i^{(j)})\,\nabla_y \log \pi_1(y_i^{(j)} \mid Y_t = y^{(j)}). \tag{5}$$

Unlike the linear-reward estimator in Eq. (4), this sample-based estimator is generally biased for finite N because the estimated first variation depends non-linearly on the empirical samples. We show the detailed algorithm of steepest guidance in Algorithm 1.

4 Regularized Steepest Guidance

We are often interested in optimizing the terminal KL-regularized objective instead of the original reward functional R to ensure that the generated samples are close to the original distribution:

$$\max_g J_\eta(\pi_1^g) := \max_g R[\pi_1^g] - \frac{1}{\eta}\mathrm{KL}(\pi_1^g \mid \pi_1), \tag{6}$$

for $\eta > 0$. From the data-processing inequality, we have $\mathrm{KL}(\pi_1^g \mid \pi_1) \leq \mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi)$. Therefore, we can obtain the following results:

Corollary 4.1. *Applying steepest guidance $g_t(y) = \lambda\sigma_t^2 \nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y)$ with $0 < \lambda \leq 2\eta$, we have*

$$J_\eta(\pi_1^g) - J_\eta(\pi_1) \geq \lambda\left(1 - \frac{\lambda}{2\eta}\right)\int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(Y_t)\right\|^2\right] \mathrm{d}t \geq 0.$$

See Appendix H for the proof. This corollary shows that the steepest guidance improves the KL-regularized objective for $0 < \lambda \leq 2\eta$. However, this guarantee constrains the guidance strength relative to the KL regularization parameter, and thus λ cannot be chosen independently of η.

Instead of controlling the terminal KL divergence via pathwise KL divergence, we can consider the terminal KL-regularized objective $J_\eta(K_t\mu)$ directly thanks to our general formulation. However, computing the first-order variation of $J_\eta(K_t\mu)$ requires the density ratio $\frac{K_t\mu(y)}{K_t\pi_t(y)}$, which is generally intractable. To deal with this issue, let us consider the quantity $\mathcal{V}(t,\mu) := V(t,\mu) - \frac{1}{\eta}\mathrm{KL}(\mu \mid \pi_t)$. From the data-processing inequality, we can show that $\mathcal{V}(t,\mu)$ is a lower bound on $J_\eta(K_t\mu)$. Based on this observation, we propose to use $\mathcal{V}(t,\mu)$ to construct the following guided SDE:

$$\mathrm{d}Y_t^g = (b_t(Y_t^g) + g_t(Y_t^g)) \cdot \mathrm{d}t + \sigma_t \,\mathrm{d}W_t, \tag{7}$$

where

$$g_t(y) = \lambda\sigma_t^2 \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta\mu}(y) = \lambda\sigma_t^2 \nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y) - \frac{\lambda}{\eta}\sigma_t^2 \nabla_y \log\frac{\pi_t^g(y)}{\pi_t(y)}.$$

Even in this case, the steepest guidance requires the density ratio but we can construct an SDE that has the same marginal distribution as Eq. (7) without computing it.

Proposition 4.2. *The following SDE has the same marginal distribution as Eq.* (7)*:*

$$\mathrm{d}Y_t^g = (b_t(Y_t^g) + g_t(Y_t^g)) \cdot \mathrm{d}t + \sqrt{1 + \tfrac{2\lambda}{\eta}} \cdot \sigma_t \cdot \mathrm{d}W_t,$$
$$\textit{where} \quad g_t(y) := \lambda\sigma_t^2 \nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y) + \tfrac{\lambda}{\eta}\sigma_t^2 \,\nabla_y \log \pi_t(y).$$

See Appendix I for the proof. We call the above method *regularized* steepest guidance. As shown in the following theorem, regularized steepest guidance improves J_η.

Theorem 4.3. *For the regularized steepest guidance, we have*

$$J_\eta(\pi_1^g) - J_\eta(\pi_1) \geq \lambda\int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta\mu}(Y_t)\right\|^2\right] \mathrm{d}t.$$

See Appendix J for the proof. In contrast to Corollary 4.1, the above theorem ensures the improvement of the KL-regularized objective even when $\lambda > 2\eta$.

4.1 Global Convergence for Concave Reward Functionals

If the reward functional R is concave, i.e., for any $\mu, \nu \in \mathcal{P}$ and $\theta \in [0,1]$, we have $R(\theta\mu + (1-\theta)\nu) \geq \theta R(\mu) + (1-\theta)R(\nu)$, then we can prove the global convergence of the regularized steepest guidance under some structural assumptions. For each $t \in [\varepsilon, 1]$, let $\pi_t^* \in \mathrm{argmax}_{\mu\in\mathcal{P}} \mathcal{V}(t,\mu)$. Following the literature (Nitanda et al., 2022; Chizat, 2022), we assume the *uniform log-Sobolev inequality* and additional regularity conditions:

Assumption 4.4. (Uniform LSI) For any $t \in [\varepsilon, 1]$, there exists a constant $\alpha_t > 0$ such that for any $\mu \in \mathcal{P}$ and smooth function $g : \mathbb{R}^d \to \mathbb{R}$, we have

$$\mathbb{E}_{\nu_t^\mu}\left[g^2(Y_t)\log g^2(Y_t)\right] - \mathbb{E}_{\nu_t^\mu}\left[g^2(Y_t)\right]\log E_{\nu_t^\mu}[g^2(Y_t)] \leq \frac{2}{\alpha_t}\mathbb{E}_{\nu_t^\mu}\left[\|\nabla_y g(Y_t)\|^2\right],$$

where $\nu_t^\mu(y) \propto \exp(\eta\frac{\delta V(t,\mu)}{\delta\mu}(y)) \cdot \pi_t(y)$. (Regularity condition) Furthermore, defining

$$s_t(y) := \nabla_y \log\frac{\pi_t^*(y)}{\pi_t(y)},\ \Phi_t(y) := \|s_t(y)\|^2,\ \Psi_t(y) := \mathrm{div}\, s_t(y) + s_t(y)\cdot\nabla_y \log\pi_t^*(y),$$

there exists $C_t < \infty$ such that $\sup_y \Phi_t(y) \leq C_t,\ \sup_y|\Psi_t(y)| \leq C_t$.

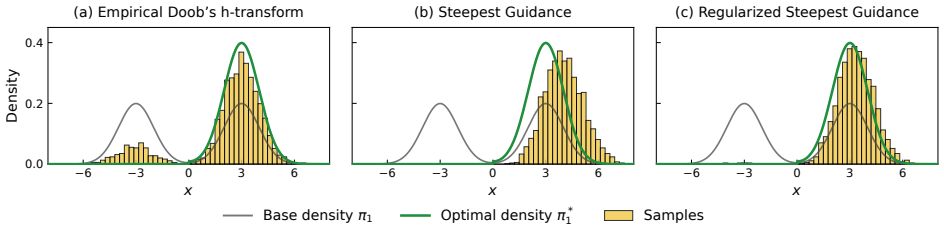

Figure 2: Toy experiment comparing Doob's h-transform (left), Steepest Guidance (middle), and Regularized Steepest Guidance (right).

Table 1: Text-to-image reward summary for a prompt "A portrait photo of a golden-yellow lion". Results are reported as mean $\pm$ standard deviation over 32 generated images for each model, reward function, and method.

Model	Reward	Unguided	Steepest (Ours)	DOIT	SVDD
SD	Blueness	-0.72 ± 0.06	$\mathbf{-0.12} \pm 0.21$	-0.69 ± 0.06	-0.67 ± 0.06
	ImageReward	-0.39 ± 0.49	$\mathbf{0.86} \pm 0.53$	0.10 ± 0.45	0.68 ± 0.44
	PickScore	20.97 ± 0.48	$\mathbf{21.90} \pm 0.48$	21.38 ± 0.45	21.84 ± 0.44
	Compressibility	-14.35 ± 1.80	$\mathbf{-5.25} \pm 1.65$	-14.17 ± 1.74	-13.54 ± 1.77
FLUX	Blueness	-0.33 ± 0.05	$\mathbf{-0.13} \pm 0.06$	-0.29 ± 0.04	-0.23 ± 0.04
	ImageReward	0.05 ± 0.28	$\mathbf{1.66} \pm 0.24$	0.45 ± 0.30	0.92 ± 0.29
	PickScore	21.45 ± 0.53	$\mathbf{22.62} \pm 0.31$	22.10 ± 0.38	22.42 ± 0.30
	Compressibility	-8.26 ± 0.77	$\mathbf{-4.68} \pm 0.56$	-7.61 ± 0.67	-6.59 ± 0.49

The LSI condition can be established through several technical tools (Chewi et al., 2022); for example, if π_t satisfies the LSI and the oscillation of $\eta \frac{\delta V(t,\mu)}{\delta \mu}(\cdot)$ is bounded, then the Bakry-Emery and Holley-Stroock arguments yield the LSI condition of ν_t^μ (Bakry & Émery, 1985; Holley & Stroock, 1987). Since π_t $(t < 1)$ is smoothed by a Gaussian, we expect the LSI constant α_t to be larger than that of the target distribution. Under the above assumption, we have the following result.

Theorem 4.5. *Assume that R is concave and Assumption 4.4 holds. Then, we have*

$$J_\eta(\pi_1^*) - J_\eta(\pi_1^g) = O\left(\varepsilon + \frac{\eta}{\lambda^2} \int_\varepsilon^1 \frac{C_t^2}{\alpha_t^2} w_\lambda(t)\, \mathrm{d}t\right).$$

where $w_\lambda(t) := \lambda \sigma_t^2 \alpha_t / \eta \cdot \exp(-\frac{\lambda}{\eta} \int_t^1 \sigma_s^2 \alpha_s \, \mathrm{d}s)$ is a weight function that satisfies $\int_\varepsilon^1 w_\lambda(t)\, \mathrm{d}t \le 1$.

See Appendix K for the proof. Thus, if $\int_\varepsilon^1 C_t^2/\alpha_t^2 \cdot w_\lambda(t)\, \mathrm{d}t$ is finite, by setting λ sufficiently large and ε sufficiently small, we can ensure that the generated distribution π_1^g is close to the optimal distribution π_1^*. Note that while the assumptions are similar, the proof techniques are different from those of mean-field Langevin dynamics (Nitanda et al., 2022; Chizat, 2022) since the objective functional $\mathcal{V}(t, \mu)$ is time-dependent and we carefully handle this time-dependency in the proof. In that sense, this convergence theorem can be viewed as a functional generalization of the well-known convergence analysis for Langevin dynamics (Bakry et al., 2014); specifically, it extends to distribution-dependent (mean-field) and time-dependent objective.

5 Numerical Experiments

In this section, we evaluate the performance of the proposed steepest guidance method on synthetic and image generation tasks.

5.1 Toy Experiments

First, we consider a toy experiment to illustrate the differences between Steepest Guidance and Doob's h-transform with finite lookahead samples. We consider a simple 1D Gaussian mixture model $\pi_1 = \frac{1}{2}\mathcal{N}(x \mid -3.0, 1.0) + \frac{1}{2}\mathcal{N}(x \mid 3.0, 1.0)$ as the target distribution and a step reward $r(y) = 10 \cdot \mathbf{1}\{y \ge 0\}$. That is, the objective is to select the right mode in the positive region.

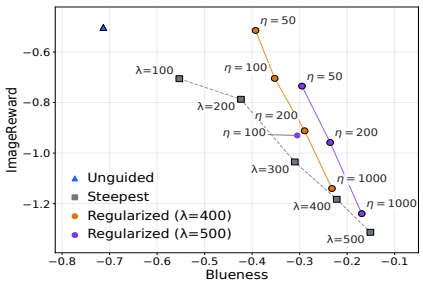

Figure 3: Trade-off between blueness and image reward for (Regularized) Steepest Guidance.

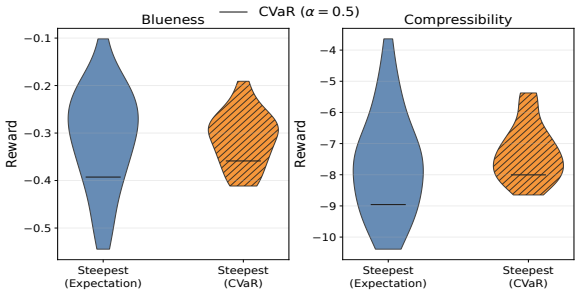

Figure 4: Comparison between Steepest Guidance with and without CVaR. Horizontal lines indicate the lower-tail CVaR with $\alpha = 0.5$.

Fig. 2 illustrates the distributions obtained by each method and the optimal distribution derived analytically. We see that 1) empirical Doob's guidance sometimes fails to select the right mode while ideally it matches the optimal distribution, 2) steepest guidance successfully selects the right mode but the distribution fails to match the optimal distribution, and 3) regularized steepest guidance successfully approximates the optimal distribution. The results match our theoretical analysis.

5.2 Image Generation

Next, we evaluate the proposed methods on image generation tasks using Stable Diffusion v1.5 (SD) (Rombach et al., 2022) and FLUX.1 [dev] (FLUX) (Black Forest Labs, 2024). As reward functions, we consider Blueness (how blue an image is), PickScore (Kirstain et al., 2023), ImageReward (Xu et al., 2023), and Compressibility (scaled negative compression size). For SD experiments using ImageReward or PickScore, we set $k = 8$; for all other experiments, we use $k = 4$. See Appendix L for details of the experimental setup and additional results including results for other prompts, generated images and ablation studies on k.

Steepest Guidance consistently outperforms baselines: Here, we consider expected reward maximization and evaluate the performance of the proposed steepest guidance against baselines including empirical Doob's transform and SVDD (Li et al., 2024), which is a state-of-the-art selection-based approach. We tune the hyperparameter λ separately for steepest guidance and Doob's h-transform for each model and reward, using random seeds distinct from those used for final evaluation. Table 1 shows that the steepest guidance outperforms baselines across reward functions and models.

Regularized Steepest Guidance improves the reward–quality trade-off: We compare the performance of regularized steepest guidance and (vanilla) steepest guidance. Since the terminal KL divergence is not directly measurable, we use ImageReward as a proxy to assess the extent to which the guided distribution deviates from the original distribution in terms of prompt alignment. Fig. 3 shows that regularized steepest guidance (orange and purple curves) achieves better trade-offs between Blueness and ImageReward than vanilla steepest guidance (gray dotted curve) by controlling the strength of the regularization parameter η.

Steepest Guidance can handle non-linear rewards: To demonstrate the effectiveness of steepest guidance for non-linear reward functionals, we consider CVaR as a reward functional. We set $\alpha = 0.5$. Fig. 4 shows the distribution of the rewards obtained by maximizing the expectation (left) and CVaR (right). We see that the proposed method achieves better CVaR (horizontal line) compared to maximizing the expectation.

6 Conclusion

We proposed *steepest guidance*, a training-free approach to aligning flow and diffusion models. Different from existing approaches based on Doob's h-transform, steepest guidance directly optimizes the reward functional in a local manner. As a result, our approach simplifies guidance estimation and accommodates general reward functionals. Theoretically, we established provable improvement guarantees for the ideal dynamics and global convergence under suitable assumptions. The experiments demonstrated the effectiveness of our proposed method in text-to-image generation tasks.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with writing, method implementation, and literature review. We have not used generative AI tools for other tasks with required disclosure. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. Specifically, the content and artifacts of this paper are fully reviewed and verified by the authors, and bibliography entries are collected manually.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide detailed descriptions of our experimental setup, including model architectures and hyperparameters in Appendix L. Furthermore, we provide self-contained source code as supplementary material.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

D. Bakry and M. Émery. Diffusions hypercontractives. In Jacques Azéma and Marc Yor (eds.), *Séminaire de Probabilités XIX 1983/84*, pp. 177–206, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. ISBN 978-3-540-39397-9.

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and geometry of Markov diffusion operators*. Springer, 2014.

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.

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

Zander W Blasingame and Chen Liu. Greed is good: A unifying perspective on guided generation. *arXiv preprint arXiv:2502.08006*, 2025.

Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of Langevin monte carlo from Poincare to Log-Sobolev. In Po-Ling Loh and Maxim Raginsky (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 1–2. PMLR, 02–05 Jul 2022.

Lénaïc Chizat. Mean-field langevin dynamics : Exponential convergence and annealing. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

G Corso, Y Xu, VD Bortoli, R Barzilay, and TS Jaakkola. Particle guidance: non-IID diverse sampling with diffusion models. *arXiv preprint arXiv:2310.13102*, 2023.

Sanjit Dandapanthula and Nicholas M Boffi. Are we really tilting? the mechanics of reward guidance in flow and diffusion models. *arXiv preprint arXiv:2606.02884*, 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *Advances in Neural Information Processing Systems*, 37:90736–90770, 2024.

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

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

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky TQ Chen, Yaron Lipman, and Brian Karrer. GLASS flows: Transition sampling for alignment of flow and diffusion models. *arXiv preprint arXiv:2509.25170*, 2025.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. *arXiv preprint arXiv:2602.05993*, 2026.

Richard Holley and Daniel Stroock. Logarithmic sobolev inequalities and stochastic ising models. *Journal of statistical physics*, 46(5-6):1159–1194, 1987.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Yuchen Jiao, Yuxin Chen, and Gen Li. Towards a unified framework for guided diffusion models. *arXiv preprint arXiv:2512.04985*, 2025.

Ryotaro Kawata, Kazusato Oko, Atsushi Nitanda, and Taiji Suzuki. Direct distributional optimization for provable alignment of diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 72792–72840, 2025.

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.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-LM improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

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

Pierre Marion, Anna Korba, Peter Bartlett, Mathieu Blondel, Valentin De Bortoli, Arnaud Doucet, Felipe Llinares-López, Courtney Paquette, and Quentin Berthet. Implicit diffusion: Efficient optimization through stochastic sampling. *arXiv preprint arXiv:2402.05468*, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Atsushi Nitanda, Denny Wu, and Taiji Suzuki. Convex analysis of the mean field langevin dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 9741–9757. PMLR, 2022.

Atsushi Nitanda, Dake Bu, Yueming Lyu, and Tanya Veeravalli. Slowly annealed langevin dynamics: Theory and applications to training-free guided generation. *arXiv preprint arXiv:2605.07950*, 2026.

L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Flow density control: Generative optimization beyond entropy-regularized fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Shokichi Takakura, Akifumi Wachi, Rei Higuchi, Kohei Miyaguchi, and Taiji Suzuki. Inference-aware meta-alignment of LLMs via non-linear GRPO. *arXiv preprint arXiv:2602.01603*, 2026.

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.

Gal Vinograd, Idan Achituve, and Ethan Fetaya. Diverse sampling in diffusion models with marginal preserving particle guidance. *arXiv preprint arXiv:2605.06553*, 2026.

Zifan Wang, Riccardo De Santi, Xiaoyu Mo, Michael M Zavlanos, Andreas Krause, and Karl H Johansson. Efficient tail-aware generative optimization via flow model fine-tuning. *arXiv preprint arXiv:2602.16796*, 2026.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation, 2023.

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, 33:4572–4583, 2020.

Qijie Zhu, Zeqi Ye, Han Liu, Zhaoran Wang, and Minshuo Chen. Training-free adaptation of diffusion models via Doob's h-transform. *arXiv preprint arXiv:2602.16198*, 2026.

Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Repulsive latent score distillation for solving inverse problems. *arXiv preprint arXiv:2406.16683*, 2024.

A Related Work

In this section, we provide a detailed discussion on related work and compare our method with previous works on reward-guided generation.

A.1 Systematic Comparison with Previous Works

Here, we show in Table 2 a systematic comparison of our method with previous works on reward-guided generation from the perspective of three key properties: training-free, derivative-free, and the ability to handle non-linear reward functionals.

A.2 Other Related Works

Diversity-seeking Generation A line of work (Corso et al., 2023; Vinograd et al., 2026; Zilberstein et al., 2024) proposes diversity-seeking generation methods utilizing repulsive forces between particles. Such diversity-seeking generation methods can be interpreted as an optimization of non-linear reward functionals but they are not applicable to general reward functionals.

Table 2: Systematic comparison with previous works on reward-guided generation. (1) Training-free: The method does not require additional training of models. (2) Derivative-free: The method does not require gradients of the reward functional or the generative process. (3) Non-linear reward functional: The method can handle non-linear reward functionals.

Method	(1)	(2)	(3)
Steepest Guidance (Ours)	✓	✓	✓
DOIT (Zhu et al., 2026)	✓	✓	×
FDC (Santi et al., 2025)	×	×	✓
Adjoint Matching (Domingo-Enrich et al., 2024)	×	×	×
Reinforce Adjoint Matching (Bergmeister et al., 2026)	×	✓	×
SALD (Nitanda et al., 2026)	✓	✓	×
Greedy Guidance (Blasingame & Liu, 2025)	✓	×	×
Gradient Guidance (Guo et al., 2024)	✓	×	×

Optimization of Non-linear Reward Functionals The optimization of general functionals over probability measures is studied in various contexts, including training of two-layer neural networks (Nitanda et al., 2022; Chizat, 2022), reinforcement learning (Zhang et al., 2020), and inference-aware training of LLMs (Takakura et al., 2026).

Relation to SALD SALD (Nitanda et al., 2026) employs dynamics similar to ours. Indeed, for linear reward functionals, greedy guidance can be regarded as a special case of SALD since SALD considers general guidance which satisfies $g_1(y) = \nabla r(y)$. However, our approach is completely different in its design principles. First of all, SALD is designed for sampling from tilted distributions, which is the optimal solution to KL-regularized expected reward maximization. Therefore, it cannot be applied to general non-linear reward functionals. On the other hand, greedy guidance is designed from the perspective of optimization of probability measures and can handle general reward functionals. Furthermore, SALD's guidance does not generally align with a direction that improves the expected reward. Consequently, its improvement guarantee relies on sufficiently slow annealing. In practice, such slowdown can significantly affect the efficiency of generation.

B Examples of Reward Functionals

In this section, we provide representative examples of (non-linear) reward functionals for various applications. Table 3 summarizes representative reward functionals. We mainly follow Santi et al. (2025) and have added several reward functionals that are not included in their work.

For entropy maximization, we cannot directly compute the first-order variation of the entropy functional since it requires knowledge of the density $\mu(x)$. Thus, we propose to use the following relation instead:

$$\mathrm{Ent}(\mu) = -\mathrm{KL}(\mu \mid \pi_1) - \mathbb{E}_{x\sim\mu}\left[\log \pi_1(x)\right].$$

Utilizing the above relation, entropy maximization can be interpreted as KL-regularized expected reward maximization. In general, we cannot compute $\log \pi_1(x)$ but we can use the score function $\nabla \log \pi_1(x)$. Thus, if the sampling process is differentiable, we can estimate the guidance using a plug-in estimator and the chain rule.

B.1 First-order Variation of Reward Functionals

Here, we provide the first-order variation of CVaR and Rao's quadratic entropy, which are used in our experiments.

Lemma B.1 (First-order variation of CVaR)**.** *Let $R_{\mathrm{CVaR}}[\mu] = \mathbb{E}_\mu\left[r(Y) \mid r(Y) \le q_\alpha\right]$ be the CVaR functional, where q_α is the α-quantile of $r(Y)$. Assume that μ has a strictly positive density on $\mathbb{R}^d$ and that the law of $r(Y)$ under μ has a continuous density f_μ in a neighborhood of q_α, with $f_\mu(q_\alpha) > 0$. Then, the first-order variation of R_{CVaR} is given by*

$$\frac{\delta R_{\mathrm{CVaR}}}{\delta \mu}[\mu](y) = \frac{1}{\alpha} \cdot \min\{r(y) - q_\alpha, 0\}.$$

Table 3: Representative reward functionals for various applications.

Application	Functional $R[\mu]$
Expected reward	$\mathbb{E}_{x\sim\mu}[r(x)]$
KL divergence	$D_{\mathrm{KL}}(\mu\|\pi_1) := \int \mu(x)\log\frac{\mu(x)}{\pi_1(x)}\,\mathrm{d}x$
Conditional value-at-risk	$\mathbb{E}_{x\sim\mu}[r(x) \mid r(x)\le q_\alpha]$
Variance	$\mathbb{E}_{x\sim\mu}\left[(r(x)-\mathbb{E}_{x\sim\mu}[r(x)])^2\right]$
Entropy	$\mathrm{Ent}[\mu] := -\mathbb{E}_{x\sim\mu}[\log\mu(x)]$
Optimal experimental design	$\mathsf{s}\big(\mathbb{E}_{x\sim\mu}\left[\Phi(x)\Phi(x)^\top - \lambda I\right]\big)$
Log-barrier	$-\beta\log(\mathbb{E}_{x\sim\mu}[c(x)] - C)$
Maximum mean discrepancy	$\mathrm{MMD}_k(\mu\|\pi_1) := \|m_\mu - m_{\pi_1}\|,\ m_\mu := \mathbb{E}_{x\sim\mu}[k(x,\cdot)]$
Best- or worst-case reward	$\mathbb{E}_{x_1,\ldots,x_N\sim\mu^N}[\max_i r(x_i)]\,,\ \mathbb{E}_{x_1\ldots,x_N\sim\mu^N}[\min_i r(x_i)]$

Proof. For $\nu\in\mathcal{P}$, let $\mu_\epsilon = (1-\epsilon)\mu + \epsilon\nu$, and denote the reward distribution function under a measure ρ by F_ρ. Write $q = q_\alpha$ and define $q_\epsilon = \inf\{z : F_{\mu_\epsilon}(z)\ge\alpha\}$.

First, we prove the differentiability of q_ϵ with respect to ϵ. Since

$$F_{\mu_\epsilon}(q_\epsilon) = (1-\epsilon)F_\mu(q_\epsilon) + \epsilon F_\nu(q_\epsilon) = \alpha, \tag{8}$$

we have $|F_\mu(q_\epsilon)-\alpha|\le\epsilon$. For sufficiently small ϵ, we have

$$\frac{f_\mu(q)}{2}|q_\epsilon - q| \le |F_\mu(q_\epsilon) - F_\mu(q)| \le \epsilon,$$

where the last inequality follows from $|F_\mu(q_\epsilon)-\alpha|\le\epsilon$. Since $f_\mu(q)>0$, we obtain $q_\epsilon - q = O(\epsilon)$, which establishes the differentiability of q_ϵ with respect to ϵ at $\epsilon=0$. By differentiating Eq. (8) with respect to ϵ, we obtain

$$f_{\mu_\epsilon}(q_\epsilon)q'_\epsilon = F_\mu(q_\epsilon) - F_\nu(q_\epsilon),$$

where $q'_\epsilon = \frac{\mathrm{d}q_\epsilon}{\mathrm{d}\epsilon}$.

Let

$$H_\rho(a) := \int r(y)\mathbf{1}\{r(y)\le a\}\rho(\mathrm{d}y).$$

Note that $R_{\mathrm{CVaR}}[\rho] = \alpha^{-1}H_\rho(q_\alpha)$. Then, we have

$$\begin{aligned} R_{\mathrm{CVaR}}[\mu_\epsilon] - R_{\mathrm{CVaR}}[\mu] &= \frac{1}{\alpha}\left(H_{\mu_\epsilon}(q_\epsilon) - H_\mu(q)\right) \\ &= \frac{1}{\alpha}\left(\underbrace{H_{\mu_\epsilon}(q_\epsilon) - H_\mu(q_\epsilon)}_{=:D_1} + \underbrace{H_\mu(q_\epsilon) - H_\mu(q)}_{=:D_2}\right). \end{aligned}$$

For D_1, we have

$$D_1 = H_{\mu_\epsilon}(q_\epsilon) - H_\mu(q_\epsilon) = \epsilon\int r(y)\mathbf{1}\{r(y)\le q_\epsilon\}(\nu-\mu)(\mathrm{d}y)$$

and

$$\lim_{\epsilon\to0}\frac{D_1}{\epsilon} = \int r(y)\mathbf{1}\{r(y)\le q\}(\nu-\mu)(\mathrm{d}y).$$

For D_2, we have

$$D_2 = H_\mu(q_\epsilon) - H_\mu(q) = qf_\mu(q)(q_\epsilon - q) + o(|q_\epsilon - q|),$$

and

$$\lim_{\epsilon\to 0}\frac{D_2}{\epsilon} = qf_\mu(q)q_0' = q(F_\mu(q) - F_\nu(q)) = \int q\mathbf{1}\{r(y)\le q\}(\mu-\nu)(\mathrm{d}y).$$

Combining the above results, we obtain

$$\begin{aligned}\lim_{\epsilon\to 0}\frac{R_{\mathrm{CVaR}}[\mu_\epsilon] - R_{\mathrm{CVaR}}[\mu]}{\epsilon} &= \frac{1}{\alpha}\left(\lim_{\epsilon\to 0}\frac{D_1}{\epsilon} + \lim_{\epsilon\to 0}\frac{D_2}{\epsilon}\right)\\ &= \frac{1}{\alpha}\int (r(y)-q)\mathbf{1}\{r(y)\le q\}(\nu-\mu)(\mathrm{d}y).\end{aligned}$$

This implies that the first-order variation of R_{CVaR} with respect to μ is

$$\frac{\delta R_{\mathrm{CVaR}}}{\delta\mu}[\mu](y) = \frac{1}{\alpha}\min\{r(y)-q, 0\}.$$

□

Lemma B.2 (First-order variation of Rao's quadratic entropy)**.** *Let $R_{\mathrm{Rao}}[\mu] = -\frac{1}{2}\int\int k(y,y')\mu(\mathrm{d}y)\mu(\mathrm{d}y')$ be Rao's quadratic entropy functional, where $k:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ is a symmetric positive definite kernel. Then, the first-order variation of R_{Rao} is given by*

$$\frac{\delta R_{\mathrm{Rao}}}{\delta\mu}[\mu](y) = -\int k(y,y')\mu(\mathrm{d}y').$$

Proof. For any $\nu\in\mathcal{P}(\mathbb{R}^d)$, let $h=\nu-\mu$. By the symmetry of k, we have

$$R_{\mathrm{Rao}}[\nu] - R_{\mathrm{Rao}}[\mu] = -\int\int k(y,y')h(\mathrm{d}y)\mu(\mathrm{d}y') - \frac{1}{2}\int\int k(y,y')h(\mathrm{d}y)h(\mathrm{d}y').$$

The first term is linear in h, while the second term is quadratic. Therefore,

$$\frac{\delta R_{\mathrm{Rao}}}{\delta\mu}[\mu](y) = -\int k(y,y')\mu(\mathrm{d}y').$$

□

C Detailed Algorithm of Steepest Guidance

Algorithm 1 Steepest Guidance for Reward-Guided Generation

1: Choose a time grid $\varepsilon = t_0 < t_1 < \cdots < t_L = 1$ and set $\Delta t_l = t_{l+1} - t_l$.
2: Sample M particles $\{y_0^{(j)}\}_{j=1}^M$ from π_ε by evolving the unguided base process from its native prior to time ε.
3: **for** $l = 0,\dots,L-1$ **do**
4: Sample $y_{l,i}^{(j)}$ from $\pi_1(\cdot \mid Y_{t_l} = y_l^{(j)})$ for $j=1,\dots,M$ and $i=1,\dots,k$.
5: Construct the empirical terminal measure $\hat\mu_{1,l} = \frac{1}{Mk}\sum_{j=1}^M\sum_{i=1}^k \delta_{y_{l,i}^{(j)}}$.
6: Compute the plug-in approximations $\frac{\delta R}{\delta\mu}[\hat\mu_{1,l}](y_{l,i}^{(j)})$ for all i and j, using the abuse of notation introduced in the main text.
7: Estimate the steepest guidance for each $j=1,\dots,M$:

$$\hat g_{t_l}^{\mathrm{steepest}}(y_l^{(j)}) = \lambda\sigma_{t_l}^2\frac{1}{k}\sum_{i=1}^k \frac{\delta R}{\delta\mu}[\hat\mu_{1,l}](y_{l,i}^{(j)})\,\nabla_{y_l^{(j)}}\log\pi_1(y_{l,i}^{(j)} \mid Y_{t_l} = y_l^{(j)}).$$

8: Update the particles:

$$y_{l+1}^{(j)} = y_l^{(j)} + \left(b_{t_l}(y_l^{(j)}) + \hat g_{t_l}^{\mathrm{steepest}}(y_l^{(j)})\right)\Delta t_l + \sigma_{t_l}\sqrt{\Delta t_l}\,\xi_l^{(j)},\quad \xi_l^{(j)}\sim\mathcal{N}(0,I).$$

9: **end for**
10: **Return:** $\{y_L^{(j)}\}_{j=1}^M$ as the generated samples.

D AUXILIARY LEMMAS

Lemma D.1 (Corollary 18 in (Albergo et al., 2025)). *Let Y_t be the solution of the SDE*

$$\mathrm{d}Y_t = b_t(Y_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t,$$

and let $\tilde{Y}_t$ be the solution of the SDE

$$\mathrm{d}\tilde{Y}_t = (b_t(\tilde{Y}_t) + g_t(\tilde{Y}_t))\,\mathrm{d}t + \sqrt{1+2\lambda}\cdot\sigma_t\,\mathrm{d}W_t,$$

where $g_t(y) = \lambda\sigma_t^2\,\nabla_y \log \pi_t(y)$. Then the marginal distribution $\tilde{\pi}_t$ of $\tilde{Y}_t$ coincides with the marginal distribution π_t of Y_t for all $t \in [0,1]$, if $\tilde{\pi}_0 = \pi_0$.

Lemma D.2 (Score Function of Flow Models). *For a flow model, the score function $\nabla \log \pi_t(y)$ can be expressed as*

$$\nabla \log \pi_t(y) = \frac{t v_t(y) - y}{1-t}$$

where $v_t(y)$ is the velocity field of the flow model.

Proof. From the definition of the flow model, we have $Y_t = tY_1 + (1-t)Y_0$, where $Y_0 \sim \mathcal{N}(0, I)$ and $Y_1 \sim \pi_1$. Therefore, we have $\frac{Y_t - tY_1}{1-t} = Y_0 \sim \mathcal{N}(0,I)$ and

$$\begin{aligned}
\pi_t(y) &= \int \pi_t(y \mid Y_1 = z)\pi_1(z)\,\mathrm{d}z \\
&= \int C \cdot \exp\left(-\frac{1}{2}\left\|\frac{y - tz}{1-t}\right\|^2\right)\pi_1(z)\,\mathrm{d}z,
\end{aligned}$$

where C is a normalization constant. Differentiating $\pi_t(y)$ with respect to y, we have

$$\begin{aligned}
\nabla\pi_t(y) &= \int C \cdot \exp\left(-\frac{1}{2}\left\|\frac{y - tz}{1-t}\right\|^2\right)\cdot\frac{tz - y}{(1-t)^2}\pi_1(z)\,\mathrm{d}z \\
&= \int \pi_t(y \mid Y_1 = z)\cdot\frac{tz - y}{(1-t)^2}\pi_1(z)\,\mathrm{d}z \\
&= \int \pi_1(z \mid Y_t = y)\cdot\frac{tz - y}{(1-t)^2}\pi_t(y)\,\mathrm{d}z \\
&= \pi_t(y)\cdot\mathbb{E}\left[\frac{tY_1 - Y_t}{(1-t)^2} \mid Y_t = y\right].
\end{aligned}$$

Thus, the score function is given by

$$\begin{aligned}
\nabla \log \pi_t(y) &= \frac{\nabla\pi_t(y)}{\pi_t(y)} \\
&= \mathbb{E}\left[\frac{tY_1 - Y_t}{(1-t)^2} \mid Y_t = y\right] \\
&= \mathbb{E}\left[\frac{t\{(1-t)(Y_1 - Y_0) + \overbrace{tY_1 + (1-t)Y_0}^{=Y_t}\} - Y_t}{(1-t)^2} \mid Y_t = y\right] \\
&= \mathbb{E}\left[\frac{t(1-t)(Y_1 - Y_0) - (1-t)Y_t}{(1-t)^2} \mid Y_t = y\right] \\
&= \mathbb{E}\left[\frac{t(Y_1 - Y_0) - Y_t}{(1-t)} \mid Y_t = y\right] \\
&= \frac{t v_t(y) - y}{1-t},
\end{aligned}$$

where the last equality follows from the definition of the velocity field $v_t(y) = \mathbb{E}\left[Y_1 - Y_0 \mid Y_t = y\right]$. □

Lemma D.3 (Conditional Distribution of Memoryless Flow)**.** *Assume that the flow model uses the memoryless noise schedule*

$$\sigma_t^2 = \frac{2(1-t)}{t}.$$

Then, for any $t \in [\varepsilon, 1)$*,*

$$Y_t \mid Y_1 = z \sim \mathcal{N}\big(tz, (1-t)^2 I\big).$$

Proof. The time reversal of Eq. (1) is given by

$$\mathrm{d}X_t = \left(-v_{1-t}(X_t) + \frac{\sigma_{1-t}^2}{2} \nabla \log \pi_{1-t}(X_t)\right) \mathrm{d}t + \sigma_{1-t}\, \mathrm{d}W_t, \qquad X_0 \sim \pi_1.$$

Since $\sigma_{1-t}^2 = 2t/(1-t)$, Lemma D.2 yields

$$\begin{aligned}\mathrm{d}X_t &= \left(-v_{1-t}(X_t) + \frac{t}{1-t} \cdot \frac{(1-t)v_{1-t}(X_t) - X_t}{t}\right) \mathrm{d}t + \sqrt{\frac{2t}{1-t}}\, \mathrm{d}W_t \\ &= -\frac{X_t}{1-t}\, \mathrm{d}t + \sqrt{\frac{2t}{1-t}}\, \mathrm{d}W_t.\end{aligned}$$

To solve this linear SDE, define $Z_s := X_s/(1-s)$. Then, we have

$$\begin{aligned}\mathrm{d}Z_s &= \frac{1}{1-s}\, \mathrm{d}X_s + \frac{X_s}{(1-s)^2}\, \mathrm{d}s \\ &= \frac{1}{1-s}\left(-\frac{X_s}{1-s}\, \mathrm{d}s + \sqrt{\frac{2s}{1-s}}\, \mathrm{d}W_s\right) + \frac{X_s}{(1-s)^2}\, \mathrm{d}s \\ &= \sqrt{\frac{2s}{(1-s)^3}}\, \mathrm{d}W_s.\end{aligned}$$

Conditional on $X_0 = z$, we have $Z_0 = z$. Integrating the above equation from 0 to s and multiplying by $1-s$ gives

$$X_s = (1-s)z + (1-s)\int_0^s \sqrt{\frac{2r}{(1-r)^3}}\, \mathrm{d}W_r.$$

The stochastic integral is centered Gaussian with covariance

$$(1-s)^2 \int_0^s \frac{2r}{(1-r)^3}\, \mathrm{d}r \cdot I = s^2 I.$$

Hence,

$$X_s \mid X_0 = z \sim \mathcal{N}\big((1-s)z, s^2 I\big).$$

Under the time-reversal coupling, $X_s = Y_{1-s}$ and $X_0 = Y_1$. Taking $s = 1-t$ proves

$$Y_t \mid Y_1 = z \sim \mathcal{N}\big(tz, (1-t)^2 I\big).$$

□

Lemma D.4 (Kolmogorov Backward Equation)**.** *Let* K_t *be the transition kernel of the base SDE from time* t *to time* 1*, and define* $V(t,\mu) := R[K_t\mu]$*. Then, for any* $t \in [\varepsilon, 1)$ *and* $\mu \in \mathcal{P}(\mathcal{Y})$*,*

$$\partial_t V(t,\mu) + \mathfrak{L}_t V(t,\mu) = 0,$$

where

$$\begin{aligned}\mathfrak{L}_t V(t,\mu) &:= \mathbb{E}_\mu\left[\mathcal{L}_t \frac{\delta V(t,\mu)}{\delta \mu}(y)\right], \\ \mathcal{L}_t f(y) &:= b_t(y) \cdot \nabla f(y) + \frac{\sigma_t^2}{2} \Delta f(y).\end{aligned}$$

Proof. The adjoint of $\mathcal{L}_t$ is defined as

$$\mathcal{L}_t^* \rho(y) := -\operatorname{div}(b_t(y)\rho(y)) + \frac{\sigma_t^2}{2}\Delta\rho(y).$$

Fix $t \in [\varepsilon, 1)$ and $\mu \in \mathcal{P}(\mathcal{Y})$, and let $(\mu_s)_{s\in[t,1]}$ be the marginals of the base SDE with $\mu_t = \mu$. By the Chapman–Kolmogorov property, $V(t,\mu) = R[\mu_1] = V(s,\mu_s)$ for every $s \in [t,1]$. Differentiating this identity with respect to s at $s = t$, and using the Fokker–Planck equation $\partial_s \mu_s = \mathcal{L}_s^* \mu_s$, gives

$$\begin{aligned} 0 &= \partial_t V(t,\mu) + \int \frac{\delta V(t,\mu)}{\delta \mu}(y) \mathcal{L}_t^* \mu(\mathrm{d}y) \\ &= \partial_t V(t,\mu) + \mathbb{E}_\mu \left[\mathcal{L}_t \frac{\delta V(t,\mu)}{\delta \mu}(y) \right] \\ &= \partial_t V(t,\mu) + \mathfrak{L}_t V(t,\mu), \end{aligned}$$

where the second equality follows from integration by parts. This proves the claim. □

Lemma D.5 (Relative Entropy Dissipation Formula)**.** *Let μ_τ and π_τ be the marginal distributions of Y_τ at time $\tau \in [\varepsilon, 1]$. Assume that μ_t and π_t for $t \in [\tau, 1]$ are defined by the marginal distributions of the SDE:*

$$\mathrm{d}Y_t = b_t(Y_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t,$$

with initial distributions μ_τ and π_τ, respectively. Then, for almost every $t \in (\tau, 1)$, we have

$$\frac{\mathrm{dKL}(\mu_t \mid \pi_t)}{\mathrm{d}t} = -\frac{\sigma_t^2}{2}\mathbb{E}_{\mu_t}\left[\left\| \nabla_y \log \frac{\mathrm{d}\mu_t}{\mathrm{d}\pi_t}(Y_t) \right\|^2 \right].$$

Proof. Let $f_t := \mathrm{d}\mu_t / \mathrm{d}\pi_t$. Both μ_t and π_t satisfy the Fokker–Planck equation $\partial_t \rho_t = \mathcal{L}_t^* \rho_t$. Hence, differentiating the relative entropy and using integration by parts gives

$$\begin{aligned} \frac{\mathrm{dKL}(\mu_t \mid \pi_t)}{\mathrm{d}t} &= \int \log f_t(y) \mathcal{L}_t^* \mu_t(\mathrm{d}y) - \int f_t(y) \mathcal{L}_t^* \pi_t(\mathrm{d}y) \\ &= \mathbb{E}_{\mu_t}\left[\mathcal{L}_t \log f_t(Y_t)\right] - \mathbb{E}_{\pi_t}\left[\mathcal{L}_t f_t(Y_t)\right]. \end{aligned}$$

Since

$$\mathcal{L}_t \log f_t = \frac{\mathcal{L}_t f_t}{f_t} - \frac{\sigma_t^2}{2}\|\nabla_y \log f_t\|^2,$$

the preceding display becomes

$$\frac{\mathrm{dKL}(\mu_t \mid \pi_t)}{\mathrm{d}t} = -\frac{\sigma_t^2}{2}\mathbb{E}_{\mu_t}\left[\|\nabla_y \log f_t(Y_t)\|^2\right],$$

which is the desired result. □

Lemma D.6 (Optimality Gap Bounds)**.** *Fix $t \in [\varepsilon, 1]$, and assume that $V(t,\cdot)$ is concave. Then, for every $\mu \in \mathcal{P}$, $\pi_t^* \in \operatorname{argmax}_{\mu\in\mathcal{P}} \mathcal{V}(t,\mu)$ satisfies*

$$\mathrm{KL}(\mu \mid \pi_t^*) \le \eta\left(\mathcal{V}(t,\pi_t^*) - \mathcal{V}(t,\mu)\right) \le \mathrm{KL}(\mu \mid \nu_t^\mu).$$

Proof. See Proposition 1 of Nitanda et al. (2022). □

Lemma D.7 (Initial optimality gap at the cutoff)**.** *Let*

$$\begin{aligned} \phi(y) &:= \frac{\delta R}{\delta \mu}[\pi_1](y), \\ L_\phi &:= \operatorname*{ess\,sup}_{\pi_1} \phi - \operatorname*{ess\,inf}_{\pi_1} \phi, \\ S_2 &:= \mathbb{E}_{\pi_1}\left[\|Y_1 - \mathbb{E}_{\pi_1}[Y_1]\|^2\right]. \end{aligned}$$

Under the bounded first-order-variation assumption, $L_\phi \le 2C_V$. Suppose that R is concave, $L_\phi < \infty$, and $S_2 < \infty$. Let $\pi_\varepsilon^* \in \operatorname{argmax}_{\mu\in\mathcal{P}} \mathcal{V}(\varepsilon,\mu)$ and define

$$\Delta_\varepsilon := \mathcal{V}(\varepsilon,\pi_\varepsilon^*) - \mathcal{V}(\varepsilon,\pi_\varepsilon).$$

Then, for the linear flow interpolation with independent endpoints,

$$\Delta_\varepsilon \le \frac{S_2}{4\eta}\left(e^{\eta L_\phi} - 1 - \eta L_\phi\right)\frac{\varepsilon^2}{(1-\varepsilon)^2} = O(\varepsilon^2).$$

For the diffusion interpolation,

$$\Delta_\varepsilon \le \frac{S_2}{4\eta}\left(e^{\eta L_\phi} - 1 - \eta L_\phi\right)\frac{\varepsilon}{1-\varepsilon} = O(\varepsilon).$$

Proof. Write

$$Y_\varepsilon = a_\varepsilon Y_1 + b_\varepsilon Z, \qquad Z \sim \mathcal{N}(0,I), \quad Z \perp Y_1,$$

where $(a_\varepsilon, b_\varepsilon) = (\varepsilon, 1-\varepsilon)$ for the linear flow interpolation and $(a_\varepsilon, b_\varepsilon) = (\sqrt{\varepsilon}, \sqrt{1-\varepsilon})$ for the diffusion interpolation. Since $K_\varepsilon \pi_\varepsilon = \pi_1$, the chain rule gives

$$f_\varepsilon(y) := \frac{\delta V(\varepsilon,\pi_\varepsilon)}{\delta \mu}(y) = \mathbb{E}\left[\phi(Y_1) \mid Y_\varepsilon = y\right].$$

Since K_ε is linear, concavity of R implies concavity of $V(\varepsilon,\cdot)$. Lemma D.6 with $\mu = \pi_\varepsilon$ and the definition of $\nu_\varepsilon^{\pi_\varepsilon}$ imply

$$\begin{aligned}\Delta_\varepsilon &\le \frac{1}{\eta}\mathrm{KL}(\pi_\varepsilon \mid \nu_\varepsilon^{\pi_\varepsilon}) \\ &= \frac{1}{\eta}\log \mathbb{E}_{\pi_\varepsilon}\left[\exp\left(\eta\left\{f_\varepsilon(Y_\varepsilon) - \mathbb{E}_{\pi_\varepsilon}\left[f_\varepsilon(Y_\varepsilon)\right]\right\}\right)\right]. \end{aligned} \tag{9}$$

Let P_y denote the conditional law of Y_1 given $Y_\varepsilon = y$. Since ϕ has essential range of length L_ϕ, Pinsker's inequality gives

$$\begin{aligned}\left|f_\varepsilon(y) - \mathbb{E}_{\pi_1}\left[\phi(Y_1)\right]\right|^2 &\le L_\phi^2\,\mathrm{TV}(P_y,\pi_1)^2 \\ &\le \frac{L_\phi^2}{2}\mathrm{KL}(P_y \mid \pi_1).\end{aligned}$$

Averaging over Y_ε yields

$$\mathrm{Var}\left(f_\varepsilon(Y_\varepsilon)\right) \le \frac{L_\phi^2}{2}\,\mathrm{MI}(Y_1;Y_\varepsilon). \tag{10}$$

Let $Q_\varepsilon := \mathcal{N}(a_\varepsilon \mathbb{E}_{\pi_1}[Y_1], b_\varepsilon^2 I)$. Then,

$$\begin{aligned}\mathrm{MI}(Y_1;Y_\varepsilon) &\le \mathbb{E}_{\pi_1}\left[\mathrm{KL}\left(\mathcal{N}(a_\varepsilon Y_1, b_\varepsilon^2 I) \mid Q_\varepsilon\right)\right] \\ &= \frac{a_\varepsilon^2}{2b_\varepsilon^2}S_2.\end{aligned}$$

Combining this with (10) gives

$$\mathrm{Var}\left(f_\varepsilon(Y_\varepsilon)\right) \le \frac{L_\phi^2 S_2}{4}\frac{a_\varepsilon^2}{b_\varepsilon^2}. \tag{11}$$

Set $U_\varepsilon := f_\varepsilon(Y_\varepsilon) - \mathbb{E}_{\pi_\varepsilon}\left[f_\varepsilon(Y_\varepsilon)\right]$. Then $\mathbb{E}\left[U_\varepsilon\right] = 0$ and $U_\varepsilon \le L_\phi$. If $L_\phi = 0$, then $U_\varepsilon = 0$ almost surely and (9) gives $\Delta_\varepsilon = 0$. Suppose that $L_\phi > 0$. For any centered random variable $U \le L$, Bennett's moment-generating-function inequality gives

$$\log \mathbb{E}\left[e^{\eta U}\right] \le \frac{\mathrm{Var}(U)}{L^2}\left(e^{\eta L} - 1 - \eta L\right).$$

Applying this inequality to (9) and using (11) gives

$$\Delta_\varepsilon \le \frac{S_2}{4\eta}\left(e^{\eta L_\phi} - 1 - \eta L_\phi\right)\frac{a_\varepsilon^2}{b_\varepsilon^2}.$$

Substituting the two choices of $(a_\varepsilon, b_\varepsilon)$ proves the claim. □

Lemma D.8 (Fisher Information Comparison via Pinsker's Inequality)**.** *Define*

$$\begin{aligned} s_t(y) &:= \nabla_y \log \frac{\pi_t^*(y)}{\pi_t(y)}, \\ \Phi_t(y) &:= \|s_t(y)\|^2, \\ \Psi_t(y) &:= \operatorname{div} s_t(y) + s_t(y) \cdot \nabla_y \log \pi_t^*(y), \end{aligned}$$

and suppose that there exists $C_t < \infty$ such that

$$\sup_y \|s_t(y)\|^2 \le C_t, \qquad \sup_y |\Psi_t(y)| \le C_t.$$

Then, for every $\mu \in \mathcal{P}$,

$$\begin{aligned} I(\pi_t^* \mid \pi_t) - I(\mu \mid \pi_t) &\le \frac{5C_t}{\sqrt{2}} \sqrt{\mathrm{KL}(\mu \mid \pi_t^*)} \\ &\quad - I(\mu \mid \pi_t^*) \\ &\le \frac{5C_t}{\sqrt{2}} \sqrt{\mathrm{KL}(\mu \mid \pi_t^*)}. \end{aligned}$$

Proof. We have

$$\nabla_y \log \frac{\mu(y)}{\pi_t(y)} = \nabla_y \log \frac{\mu(y)}{\pi_t^*(y)} + s_t(y).$$

Expanding the square yields

$$I(\mu \mid \pi_t) = I(\mu \mid \pi_t^*) + 2 \int \mu(y) \nabla_y \log \frac{\mu(y)}{\pi_t^*(y)} \cdot s_t(y) \, \mathrm{d}y + \mathbb{E}_\mu \left[\Phi_t\right].$$

By integration by parts,

$$\int \mu(y) \nabla_y \log \frac{\mu(y)}{\pi_t^*(y)} \cdot s_t(y) \, \mathrm{d}y = -\mathbb{E}_\mu \left[\Psi_t\right].$$

Furthermore,

$$\mathbb{E}_{\pi_t^*} \left[\Psi_t\right] = 0, \qquad I(\pi_t^* \mid \pi_t) = \mathbb{E}_{\pi_t^*} \left[\Phi_t\right].$$

Consequently,

$$\begin{aligned} I(\pi_t^* \mid \pi_t) - I(\mu \mid \pi_t) = \; & \left(\mathbb{E}_{\pi_t^*} \left[\Phi_t\right] - \mathbb{E}_\mu \left[\Phi_t\right]\right) \\ & + 2 \left(\mathbb{E}_\mu \left[\Psi_t\right] - \mathbb{E}_{\pi_t^*} \left[\Psi_t\right]\right) - I(\mu \mid \pi_t^*). \end{aligned}$$

Since $0 \le \Phi_t \le C_t$ and $|\Psi_t| \le C_t$, we have

$$\begin{aligned} \mathbb{E}_{\pi_t^*} \left[\Phi_t\right] - \mathbb{E}_\mu \left[\Phi_t\right] &\le C_t \, \mathrm{TV}(\mu, \pi_t^*), \\ \mathbb{E}_\mu \left[\Psi_t\right] - \mathbb{E}_{\pi_t^*} \left[\Psi_t\right] &\le 2C_t \, \mathrm{TV}(\mu, \pi_t^*). \end{aligned}$$

Pinsker's inequality gives

$$\mathrm{TV}(\mu, \pi_t^*) \le \sqrt{\frac{1}{2} \mathrm{KL}(\mu \mid \pi_t^*)}.$$

Combining these inequalities gives the first inequality in the statement. The second follows from $I(\mu \mid \pi_t^*) \ge 0$. □

E Proof of Lemma 2.2

By Bayes' rule, we have

$$\log \pi_1(z \mid Y_t = y) = \log \pi_t(y \mid Y_1 = z) + \log \pi_1(z) - \log \pi_t(y),$$

and differentiating both sides with respect to y gives

$$\nabla_y \log \pi_1(z \mid Y_t = y) = \nabla_y \log \pi_t(y \mid Y_1 = z) - \nabla_y \log \pi_t(y).$$

The second term $\nabla \log \pi_t(y)$ is the score function of π_t, which is estimated by a neural network for diffusion models and can be computed from the velocity field for flow models using Lemma D.2. Therefore, we only need to compute the first term $\nabla \log \pi_t(y \mid Y_1 = z)$.

Flow Models. Under the memoryless noise schedule, Lemma D.3 gives

$$\pi_t(y \mid Y_1 = z) = C \cdot \exp\left(-\frac{1}{2(1-t)^2}\|y - tz\|^2\right),$$

where C is a normalization constant. Thus,

$$\nabla_y \log \pi_t(y \mid Y_1 = z) = -\frac{1}{(1-t)^2}(y - tz).$$

Diffusion Models. For $t \in [\varepsilon, 1)$, let $s = 1 - t \in (0, 1 - \varepsilon]$. The solution of the forward SDE satisfies

$$X_s \mid X_0 = z \sim \mathcal{N}(\sqrt{1-s}z, sI).$$

The reverse process is the time reversal of the forward process, so $Y_t = X_{1-t}$ and $Y_1 = X_0$ under the time-reversal coupling. Therefore,

$$Y_t \mid Y_1 = z \sim \mathcal{N}(\sqrt{t}z, (1-t)I)$$

and we have

$$\nabla_y \log \pi_t(y \mid Y_1 = z) = -\frac{1}{1-t}(y - \sqrt{t}z).$$

F Proof of Proposition 3.1

The marginal distribution π_t^g of the guided SDE satisfies the Fokker–Planck equation

$$\partial_t \pi_t^g = \mathcal{L}_t^* \pi_t^g - \operatorname{div}(g_t \pi_t^g).$$

Therefore, the chain rule for functions of measures and integration by parts give

$$\begin{aligned}
\frac{\mathrm{d}V(t, \pi_t^g)}{\mathrm{d}t} &= \partial_t V(t, \pi_t^g) + \int \frac{\delta V(t, \pi_t^g)}{\delta \mu}(y)\, \partial_t \pi_t^g(\mathrm{d}y) \\
&= \partial_t V(t, \pi_t^g) + \mathbb{E}_{\pi_t^g}\left[\mathcal{L}_t \frac{\delta V(t, \pi_t^g)}{\delta \mu}(y)\right] + \mathbb{E}_{\pi_t^g}\left[g_t(y) \cdot \nabla \frac{\delta V(t, \pi_t^g)}{\delta \mu}(y)\right] \\
&= \partial_t V(t, \pi_t^g) + \mathfrak{L}_t^g V(t, \pi_t^g),
\end{aligned}$$

where

$$\mathfrak{L}_t^g V(t, \mu) := \mathfrak{L}_t V(t, \mu) + \mathbb{E}_\mu\left[g_t(y) \cdot \nabla \frac{\delta V(t, \mu)}{\delta \mu}(y)\right].$$

Applying Lemma D.4, we have

$$\begin{aligned}
\frac{\mathrm{d}V(t, \pi_t^g)}{\mathrm{d}t} &= \partial_t V(t, \pi_t^g) + \mathfrak{L}_t V(t, \pi_t^g) + \mathbb{E}_{\pi_t^g}\left[g_t(y) \cdot \nabla \frac{\delta V(t, \pi_t^g)}{\delta \mu}(y)\right] \\
&= \mathbb{E}_{\pi_t^g}\left[g_t(y) \cdot \nabla \frac{\delta V(t, \pi_t^g)}{\delta \mu}(y)\right].
\end{aligned}$$

Integrating both sides from $t = \varepsilon$ to $t = 1$ and using $K_\varepsilon \pi_\varepsilon = \pi_1$, we have

$$\begin{aligned}
V(1, \pi_1^g) - V(\varepsilon, \pi_\varepsilon) &= \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[g_t(y) \cdot \nabla \frac{\delta V(t, \pi_t^g)}{\delta \mu}(y)\right] \mathrm{d}t \\
&= R[\pi_1^g] - R[\pi_1].
\end{aligned}$$

This completes the proof of the reward improvement.

Since the base and guided processes share the initial marginal π_ε, the KL identity follows from Girsanov's theorem and the admissibility conditions:

$$\mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi) = \frac{1}{2}\int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\frac{\|g_t(Y_t)\|^2}{\sigma_t^2}\right] \mathrm{d}t.$$

G Proof of Theorem 3.2

Applying Proposition 3.1 to the steepest guidance $g_t(y) = \lambda\sigma_t^2 \nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y)$, we have

$$R[\pi_1^g] - R[\pi_1] = \lambda \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y)\right\|^2\right] \mathrm{d}t,$$
$$\mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi) = \frac{\lambda^2}{2} \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y)\right\|^2\right] \mathrm{d}t.$$

Combining the above two equations, we have

$$R[\pi_1^g] - R[\pi_1] - \frac{1}{\lambda}\mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi) = \frac{\lambda}{2} \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(y)\right\|^2\right] \mathrm{d}t$$

This completes the proof.

H Proof of Corollary 4.1

Define

$$A := \int_\varepsilon^1 \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\|\nabla_y \frac{\delta V(t,\pi_t^g)}{\delta\mu}(Y_t)\right\|^2\right] \mathrm{d}t.$$

From Theorem 3.2, we have

$$R[\pi_1^g] - R[\pi_1] = \lambda A, \qquad \mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi) = \frac{\lambda^2}{2} A.$$

From the information processing inequality, we have $\mathrm{KL}(\pi_1^g \mid \pi_1) \le \mathrm{KL}(\mathbb{P}_{\pi^g} \mid \mathbb{P}_\pi)$. Therefore,

$$\begin{aligned} J_\eta(\pi_1^g) - J_\eta(\pi_1) &= R[\pi_1^g] - \frac{1}{\eta}\mathrm{KL}(\pi_1^g \mid \pi_1) - R[\pi_1] \\ &\ge \lambda A - \frac{\lambda^2}{2\eta} A \\ &= \lambda\left(1 - \frac{\lambda}{2\eta}\right) A \\ &\ge 0, \end{aligned}$$

where the last inequality follows from $0 < \lambda \le 2\eta$. This completes the proof.

I Proof of Proposition 4.2

The result follows from Lemma D.1.

J Proof of Theorem 4.3

By Proposition 4.2, it suffices to analyze the guided SDE with $g_t(y) = \lambda\sigma_t^2 \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta\mu}(y)$. Since $V(t,\mu) = R[K_t\mu]$ satisfies the Kolmogorov backward equation, Lemmas D.4 and D.5 give

$$\partial_t \mathcal{V}(t,\mu) + \mathfrak{L}_t \mathcal{V}(t,\mu) = \frac{\sigma_t^2}{2\eta} I(\mu \mid \pi_t) \ge 0.$$

The chain-rule calculation for the guided SDE, together with $g_t(y) = \lambda\sigma_t^2 \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta \mu}(y)$, then yields

$$\begin{aligned}\frac{\mathrm{d}\mathcal{V}(t,\pi_t^g)}{\mathrm{d}t} &= \partial_t \mathcal{V}(t,\pi_t^g) + \mathfrak{L}_t \mathcal{V}(t,\pi_t^g) + \mathbb{E}_{\pi_t^g}\left[g_t(Y_t)\cdot \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta \mu}(Y_t)\right] \\ &= \frac{\sigma_t^2}{2\eta} I(\pi_t^g \mid \pi_t) + \lambda \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\| \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta \mu}(Y_t)\right\|^2\right] \\ &\geq \lambda \mathbb{E}_{\pi_t^g}\left[\sigma_t^2 \left\| \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta \mu}(Y_t)\right\|^2\right].\end{aligned}$$

Integrating over $t \in [\varepsilon, 1]$ and using

$$\mathcal{V}(\varepsilon, \pi_\varepsilon) = R[K_\varepsilon \pi_\varepsilon] - \eta^{-1}\mathrm{KL}(\pi_\varepsilon \mid \pi_\varepsilon) = R[\pi_1] = J_\eta(\pi_1)$$

and $\mathcal{V}(1,\pi_1^g) = J_\eta(\pi_1^g)$ proves the claim.

K Proof of Theorem 4.5

For $t \in [\varepsilon, 1]$, let $\Delta_t = \mathcal{V}(t,\pi_t^*) - \mathcal{V}(t,\pi_t^g)$. Since $\pi_\varepsilon^g = \pi_\varepsilon$, the initial gap is

$$\Delta_\varepsilon = \mathcal{V}(\varepsilon,\pi_\varepsilon^*) - \mathcal{V}(\varepsilon,\pi_\varepsilon) \geq 0.$$

Since we assume that π_1 has a finite second moment, and the first-order variation of R is bounded, it follows from Lemma D.7 that $\Delta_\varepsilon = O(\varepsilon^2)$ for the linear flow interpolation and $\Delta_\varepsilon = O(\varepsilon)$ for the diffusion interpolation. For simplicity, write $\nu_t^g := \nu_t^{\pi_t^g}$. By the definition of $\mathcal{V}$,

$$\nabla_y \frac{\delta \mathcal{V}(t,\mu)}{\delta \mu}(y) = -\frac{1}{\eta}\nabla_y \log \frac{\mu(y)}{\nu_t^\mu(y)}.$$

From the first-order optimality condition, $\pi_t^* = \nu_t^{\pi_t^*}$. Hence, $\frac{\delta \mathcal{V}(t,\pi_t^*)}{\delta \mu}$ is constant in y and $\mathfrak{L}_t \mathcal{V}(t,\pi_t^*) = 0$. Moreover, Lemmas D.4 and D.5 imply

$$\partial_t \mathcal{V}(t,\mu) + \mathfrak{L}_t \mathcal{V}(t,\mu) = \frac{\sigma_t^2}{2\eta} I(\mu \mid \pi_t).$$

In particular, since $\mathfrak{L}_t \mathcal{V}(t,\pi_t^*) = 0$, we have

$$\partial_t \mathcal{V}(t,\pi_t^*) = \frac{\sigma_t^2}{2\eta} I(\pi_t^* \mid \pi_t),$$

whereas

$$\partial_t \mathcal{V}(t,\pi_t^g) + \mathfrak{L}_t \mathcal{V}(t,\pi_t^g) = \frac{\sigma_t^2}{2\eta} I(\pi_t^g \mid \pi_t).$$

By Proposition 4.2, we may use the Fokker–Planck equation of the marginally equivalent ideal SDE:

$$\partial_t \pi_t^g = \mathcal{L}_t^* \pi_t^g - \lambda\sigma_t^2 \,\mathrm{div}\left(\pi_t^g \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta \mu}\right).$$

The envelope theorem and integration by parts now yield

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\Delta_t =\ & \partial_t \mathcal{V}(t,\pi_t^*) - (\partial_t \mathcal{V}(t,\pi_t^g) + \mathfrak{L}_t \mathcal{V}(t,\pi_t^g)) \\ & - \lambda\sigma_t^2 \mathbb{E}_{\pi_t^g}\left[\left\| \nabla_y \frac{\delta \mathcal{V}(t,\pi_t^g)}{\delta \mu}(Y_t)\right\|^2\right] \\ =\ & -\frac{\lambda\sigma_t^2}{\eta^2} I(\pi_t^g \mid \nu_t^g) + \frac{\sigma_t^2}{2\eta}\left(I(\pi_t^* \mid \pi_t) - I(\pi_t^g \mid \pi_t)\right).\end{aligned}$$

Since K_t is linear, concavity of R implies concavity of $V(t,\cdot)$. Applying Lemma D.6 with $\mu = \pi_t^g$ gives

$$\mathrm{KL}(\pi_t^g \mid \pi_t^*) \le \eta\Delta_t \le \mathrm{KL}(\pi_t^g \mid \nu_t^g).$$

Thus, LSI(α_t) yields

$$I(\pi_t^g \mid \nu_t^g) \ge 2\eta\alpha_t\Delta_t.$$

Applying Lemma D.8 with $\mu = \pi_t^g$ gives

$$\begin{aligned} I(\pi_t^* \mid \pi_t) - I(\pi_t^g \mid \pi_t) &\le \frac{5C_t}{\sqrt{2}}\sqrt{\mathrm{KL}(\pi_t^g \mid \pi_t^*)} \\ &\le 5C_t\sqrt{\frac{\eta}{2}}\sqrt{\Delta_t}. \end{aligned}$$

Consequently,

$$\frac{\mathrm{d}}{\mathrm{d}t}\Delta_t \le -2a_t\Delta_t + 2a_t h_t\sqrt{\Delta_t},$$

where

$$\begin{aligned} a_t &:= \frac{\lambda\sigma_t^2\alpha_t}{\eta}, & h_t &:= \frac{5\sqrt{\eta}C_t}{4\sqrt{2}\lambda\alpha_t}, \\ A_{\lambda,\varepsilon} &:= \int_\varepsilon^1 a_s\,\mathrm{d}s, & w_\lambda(t) &:= a_t\exp\left(-\int_t^1 a_s\,\mathrm{d}s\right). \end{aligned}$$

For $\delta > 0$, let $u_{\delta,t} := \sqrt{\Delta_t + \delta}$. Then,

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{\delta,t} \le -a_t u_{\delta,t} + a_t h_t + a_t\sqrt{\delta}.$$

Applying Grönwall's inequality and sending $\delta \downarrow 0$ implies

$$\sqrt{\Delta_1} \le e^{-A_{\lambda,\varepsilon}}\sqrt{\Delta_\varepsilon} + \int_\varepsilon^1 h_t w_\lambda(t)\,\mathrm{d}t.$$

Moreover,

$$\int_\varepsilon^1 w_\lambda(t)\,\mathrm{d}t = 1 - e^{-A_{\lambda,\varepsilon}}.$$

Hence, $e^{-A_{\lambda,\varepsilon}}\delta_\varepsilon + w_\lambda(t)\,\mathrm{d}t$ is a probability measure on $\{\varepsilon\} \cup [\varepsilon, 1]$. Jensen's inequality gives

$$\begin{aligned} \Delta_1 &\le e^{-A_{\lambda,\varepsilon}}\Delta_\varepsilon + \int_\varepsilon^1 h_t^2 w_\lambda(t)\,\mathrm{d}t \\ &= e^{-A_{\lambda,\varepsilon}}\Delta_\varepsilon + \frac{25\eta}{32\lambda^2}\int_\varepsilon^1 \frac{C_t^2}{\alpha_t^2} w_\lambda(t)\,\mathrm{d}t \\ &\le O(\varepsilon) + \frac{25\eta}{32\lambda^2}\int_\varepsilon^1 \frac{C_t^2}{\alpha_t^2} w_\lambda(t)\,\mathrm{d}t, \end{aligned}$$

where the initialization term is in fact $O(\varepsilon^2)$ for the linear flow interpolation. Finally, since K_1 is the identity kernel, $\mathcal{V}(1,\mu) = J_\eta(\mu)$ and therefore

$$\Delta_1 = \sup_{\mu\in\mathcal{P}} J_\eta(\mu) - J_\eta(\pi_1^g),$$

which proves the claim.

L Experimental Details and Additional Results

L.1 Variance Reduction

Equations (4) and (5) state the uncentered estimators used in the theoretical discussion. In all experiments, we center the coefficients multiplying the conditional score to reduce variance and remove the finite-sample dependence on the additive constant of the first-order variation.

For Steepest Guidance, let

$$\psi_i^{(j)} := \frac{\delta R}{\delta \mu}[\hat{\mu}_{1,l}](y_{l,i}^{(j)}), \qquad b_{-i}^{(j)} := \frac{1}{k-1}\sum_{i' \neq i} \psi_{i'}^{(j)},$$

where $k \geq 2$ and $b_{-i}^{(j)}$ is the leave-one-out baseline. The estimator used in the experiments is

$$\hat{g}_{t_l}^{\text{steepest,LOO}}(y_l^{(j)}) = \lambda \sigma_{t_l}^2 \frac{1}{k}\sum_{i=1}^{k} \left(\psi_i^{(j)} - b_{-i}^{(j)}\right) \nabla_{y_l^{(j)}} \log \pi_1(y_{l,i}^{(j)} \mid Y_{t_l} = y_l^{(j)}).$$

This estimator is exactly invariant to replacing every $\psi_i^{(j)}$ by $\psi_i^{(j)} + c$ for an arbitrary constant c. For a linear reward functional, the leave-one-out baseline is independent of the i-th lookahead sample conditional on $Y_{t_l} = y_l^{(j)}$ and therefore preserves the unbiasedness of the estimator because the conditional score has zero expectation. For non-linear reward functionals, the finite-particle estimator remains generally biased as discussed in the main text; the leave-one-out centering is used for invariance and variance reduction.

For the softmax estimator used by DOIT, the coefficients have empirical mean $1/k$ by construction, where k is the number of lookahead samples. We therefore subtract $1/k$ from each softmax coefficient before multiplying it by the conditional score. Since the conditional score has zero expectation, this centering leaves the target expectation unchanged while removing the constant component of the coefficient and reducing variance.

L.2 Experimental Setup for Bias Evaluation.

Here, we describe the experimental setup of the bias evaluation. We evaluate the finite-sample bias in Fig. 1 on a one-dimensional Gaussian toy problem. The terminal distribution is $X_1 \sim \mathcal{N}(0,1)$, and we evaluate the guidance at $t = 0.5$. We set the linear reward $r(x) = x$ and $\lambda = 5$. In this toy problem, we can sample the conditional distribution exactly.

For each of 128 independently sampled states and each $k \in \{1, 2, 4, 8, 16, 32\}$, we draw 512 independent REINFORCE estimates. Let g^* denote the analytic Doob's guidance, $\bar{g}_k$ the empirical one, and s_k^2 the unbiased empirical variance across repeats. We report the relative bias

$$\frac{\sqrt{\mathbb{E}_y\left[(\bar{g}_k - g^*)^2 - s_k^2/512\right]}}{\sqrt{\mathbb{E}_y\left[(g^*)^2\right]}},$$

where the expectations are empirical averages across states. The subtraction in the numerator removes the finite-repeat variance contribution to the squared error of $\bar{g}_k$. Error bars show the standard error across states.

Finally, we show that the plug-in estimator has no bias in this Gaussian linear-reward setting. The conditional sample can be expressed as follows:

$$X_i(y) = a_t y + b_t \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0,1),$$

where a_t and b_t are functions of t but not of y. Then, the plug-in estimator is

$$\begin{aligned}\hat{g}_k^{\text{plug}}(y) &= \sigma_t^2 \nabla_y \log\left(\frac{1}{k}\sum_{i=1}^{k} \exp\big(\lambda X_i(y)\big)\right) \\ &= \sigma_t^2 \nabla_y \left[\lambda a y + \log\left(\frac{1}{k}\sum_{i=1}^{k} \exp(\lambda b \epsilon_i)\right)\right] = \sigma_t^2 \lambda a.\end{aligned}$$

On the other hand, the analytic Doob's guidance is

$$g^*(y) = \sigma_t^2 \nabla_y \log \mathbb{E}\left[\exp\big(\lambda(ay + b\epsilon)\big)\right] = \sigma_t^2 \nabla_y \left(\lambda a y + \frac{\lambda^2 b^2}{2}\right) = \sigma_t^2 \lambda a.$$

Thus, $\hat{g}_k^{\text{plug}}(y) = g^*(y)$, which implies that the plug-in estimator has no bias in this setting.

L.3 Reward Functions.

ImageReward and PickScore. ImageReward (Xu et al., 2023) and PickScore (Kirstain et al., 2023) are reward models that predict the score of an image based on human preferences.

Blueness. Following (Dandapanthula & Boffi, 2026), we define the blueness reward function as

$$r_{\text{blue}}(x) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} (B_{i,j} - R_{i,j} - G_{i,j}),$$

where H and W are the height and width of the image, respectively, and $R_{i,j}$, $G_{i,j}$, and $B_{i,j}$ are the red, green, and blue pixel values at position (i, j), respectively.

Compressibility. Compressibility is defined as

$$r_{\text{compress}}(x) = -\frac{\text{JPEGSize}_{q=95}(x)}{10000},$$

where $\text{JPEGSize}_{q=95}(x)$ is the size of the JPEG-compressed image x with quality factor $q = 95$. This is an example of a non-differentiable reward function.

L.4 Detailed Setup for Text-to-Image Experiments.

Hyperparameter Settings. We set $L = 50$ for FLUX and $L = 160$ for SD and sample 32 images with batch size $M = 8$ for each method in the main experiments. We summarize the grid for hyperparameter search and optimal λ in Table 4.

Guidance Schedule. For ImageReward and PickScore, we apply the guidance from $l = 10$ to $l = 50$ for FLUX and from $l = 120$ to $l = 160$ for SD, where l is the diffusion step index. For Blueness and Compressibility, we apply the guidance from $l = 0$ to $l = 10$ for FLUX and $l = 0$ to $l = 50$ for SD since the rewards are based on global structure acquired early in the process. For SD, we apply the guidance every 5 steps to reduce the computational cost.

Approximate Posterior Sampling. Following Dandapanthula & Boffi (2026), we use the Diamond Map (Holderrieth et al., 2026) for approximate posterior sampling of FLUX. Pretrained weights are available at `https://huggingface.co/gabeguofanclub/flux-1-dev-flowmap-lsd`. For SD, we cannot find a pretrained Diamond Map, so we utilize Tweedie's formula instead for 1-step denoising. The noise ratio is set to 5.0 for both models.

L.5 Additional Results on Text-to-Image Generation.

In addition to the main results reported in Table 1, we provide results for two additional prompts in Tables 5 and 6. We observe that the proposed method achieves better or comparable performance across the additional prompts, demonstrating its robustness.

L.6 Examples of Generated Images.

Here, we provide examples of images generated with SD and FLUX for the Blueness, ImageReward, PickScore, and Compressibility reward functions.



Figure 5: Selected images generated with SD for the Blueness, ImageReward, PickScore, and Compressibility reward functions (top to bottom).

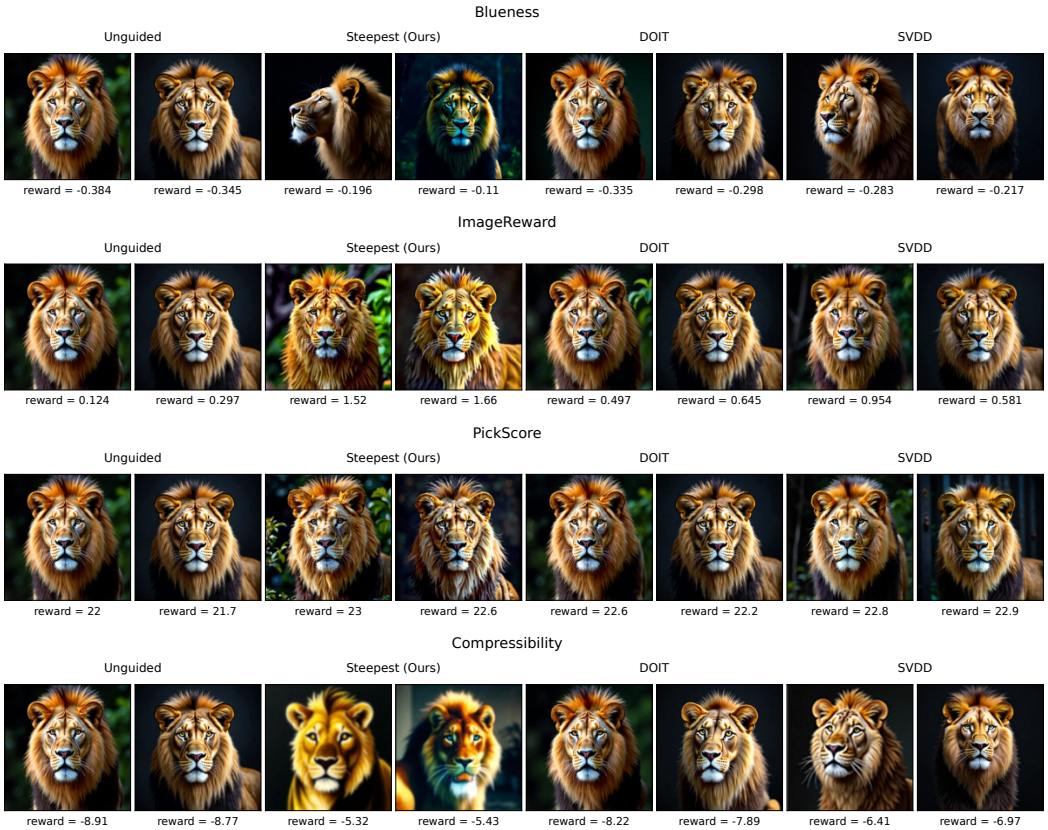

Figure 6: Selected images generated with FLUX for the Blueness, ImageReward, PickScore, and Compressibility reward functions (top to bottom).

Table 4: A summary of the grid search for the guidance scale λ in the text-to-image experiments. The table shows the search grid and selected optimal λ for each prompt, reward function, model, and method. Lion, Tokyo, and Lake denote the prompts in Tables 1, 5, and 6, respectively.

		SD		FLUX	
Reward	Grid	Steepest	Doob	Steepest	Doob
Lion					
Blueness	[50, 100, 200, 300, 400, 500]	500	300	500	400
ImageReward	[20, 30, 40, 50, 60, 70]	50	30	40	30
PickScore	[20, 30, 40, 50, 60, 70]	40	40	30	40
Compressibility	[10, 20, 40, 60, 80, 100]	100	20	100	100
Tokyo					
Blueness	[50, 100, 200, 300, 400, 500]	500	400	400	200
ImageReward	[20, 30, 40, 50, 60, 70]	70	70	40	60
PickScore	[20, 30, 40, 50, 60, 70]	30	60	20	50
Compressibility	[10, 20, 40, 60, 80, 100]	100	10	40	10
Lake					
Blueness	[50, 100, 200, 300, 400, 500]	500	400	500	500
ImageReward	[20, 30, 40, 50, 60, 70]	70	60	50	70
PickScore	[20, 30, 40, 50, 60, 70]	40	50	20	30
Compressibility	[10, 20, 40, 60, 80, 100]	100	80	40	100

Table 5: Text-to-image reward summary for the prompt "A cinematic night photograph of a rainy Tokyo street with neon reflections." Results are reported as mean $\pm$ standard deviation over $n = 32$ generated samples.

Model	Reward	Unguided	Steepest (Ours)	DOIT	SVDD
SD	Blueness	-0.34 ± 0.07	$\mathbf{0.15} \pm 0.21$	-0.31 ± 0.07	-0.28 ± 0.07
	ImageReward	1.32 ± 0.29	$\mathbf{1.78} \pm 0.14$	1.55 ± 0.24	1.75 ± 0.15
	PickScore	22.12 ± 0.47	22.90 ± 0.48	22.46 ± 0.50	$\mathbf{22.95} \pm 0.51$
	Compressibility	-16.35 ± 1.73	$\mathbf{-4.61} \pm 1.99$	-15.98 ± 1.77	-15.60 ± 1.67
FLUX	Blueness	-0.22 ± 0.04	$\mathbf{-0.03} \pm 0.07$	-0.17 ± 0.04	-0.11 ± 0.03
	ImageReward	0.93 ± 0.52	$\mathbf{1.84} \pm 0.07$	1.41 ± 0.48	1.56 ± 0.34
	PickScore	22.23 ± 0.47	$\mathbf{23.53} \pm 0.32$	22.90 ± 0.49	23.30 ± 0.34
	Compressibility	-8.12 ± 1.12	$\mathbf{-5.13} \pm 0.27$	-7.15 ± 0.76	-6.35 ± 0.57

L.7 Additional Results on Non-linear Reward Functionals.

Diversity Seeking Generation. To enhance diversity in generated images, we compare entropy and Rao's quadratic entropy as reward functionals. For Rao's quadratic entropy, we use DINOv2 to extract features and an RBF kernel to compute the entropy. Note that entropy regularization does not require an additional model for feature extraction. The bandwidth of the RBF kernel is adaptively set to the median of pairwise squared distances between features divided by $\max\{\log M, 1\}$. We set the guidance scale λ to $1,000$ and Rao's quadratic entropy reward weight to 1. For entropy maximization, we use the KL-regularized formulation described in Section B and set its reward and KL weights to 2×10^{-4}. Figure 7 shows images generated for the prompt "VAN GOGH CAFE TERASSE" using Rao's quadratic entropy, entropy regularization, and unguided sampling. While unguided sampling produces images with similar scene composition (e.g., the 1st, 3rd, 4th, and 7th images in the bottom row), regularized sampling produces visually varied images. We also report in-batch diversity scores in Table 7. The score is computed as the mean pairwise cosine distance between the normalized DINOv2 image embeddings within the batch. As we discuss in Section A, several methods (Corso et al., 2023; Vinograd et al., 2026; Zilberstein et al., 2024) have been proposed to enhance diversity in generated images but they are based on heuristics or require first- and second-order derivatives of the kernel.

Table 6: Text-to-image reward summary for the prompt "A dreamy watercolor painting of a mountain lake at sunrise." Results are reported as mean ± standard deviation over $n = 32$ generated samples.

Model	Reward	Unguided	Steepest (Ours)	DOIT	SVDD
SD	Blueness	-0.47 ± 0.10	$\mathbf{0.32} \pm 0.20$	-0.44 ± 0.10	-0.40 ± 0.09
	ImageReward	0.03 ± 0.43	$\mathbf{0.64} \pm 0.16$	0.33 ± 0.26	0.58 ± 0.18
	PickScore	21.84 ± 0.54	22.54 ± 0.52	22.12 ± 0.53	$\mathbf{22.57} \pm 0.52$
	Compressibility	-11.76 ± 1.42	$\mathbf{-4.21} \pm 0.96$	-11.41 ± 1.44	-10.98 ± 1.59
FLUX	Blueness	-0.58 ± 0.07	$\mathbf{0.08} \pm 0.09$	-0.48 ± 0.07	-0.38 ± 0.07
	ImageReward	0.97 ± 0.21	$\mathbf{1.42} \pm 0.12$	1.17 ± 0.18	1.34 ± 0.17
	PickScore	23.44 ± 0.49	$\mathbf{24.37} \pm 0.36$	24.02 ± 0.38	24.32 ± 0.45
	Compressibility	-7.78 ± 1.12	$\mathbf{-4.66} \pm 0.29$	-6.83 ± 1.03	-5.54 ± 0.48

Table 7: Within-batch diversity for the eight images in Fig. 7, measured as mean pairwise cosine distance between normalized DINOv2 image embeddings. Higher is more diverse.

Method	DINO diversity ↑
Rao's Quadratic Entropy	0.555
Entropy	0.504
Unguided	0.428

L.8 Effect of Guidance Scaling

Some prior works (Zhu et al., 2026; Dandapanthula & Boffi, 2026) introduce the heuristic of using a guidance scaling factor γ to control the strength of the guidance. That is, the guidance is scaled by the factor γ: $\gamma g_t(y)$. To see the effect of guidance scaling, we vary γ for empirical Doob's h-transform and observe how it influences the resulting reward. Here, we fix $\lambda = 30$ for ImageReward and $\lambda = 100$ for Compressibility, with $k = 4$. We use $\gamma \in \{2, 4, 6, 8, 10, 12\}$ for ImageReward and $\gamma \in \{5, 10, 20, 40, 60, 80\}$ for Compressibility. Fig. 8 shows the results of varying the guidance scaling factor γ and compares them to our steepest guidance. We see that applying a guidance scaling improves the performance to some extent, but it does not outperform steepest guidance. One possible explanation is that guidance scaling may amplify a bias in the guidance, leading to suboptimal performance. This suggests that the observed performance gap in the main experiments cannot be explained solely by insufficient guidance strength

L.9 Comparison with SALD

Here, we compare the performance of the proposed steepest guidance method with SALD (Nitanda et al., 2026). We do not apply slowdown here to reduce the computational cost. Following the original paper, we consider define f_t in SALD as the Gaussian-smoothed reward function, i.e., $f_{t_l}(y) = \mathbb{E}_{Z\sim\mathcal{N}(0,I)}[r(y + \bar{\sigma}_{t_l} Z)]$ for $\bar{\sigma}_{t_l} = \sqrt{\Delta_{t_l}}\sigma_t$. and gradients are estimated using $\nabla f_{t_l}(y) \simeq \frac{1}{k\bar{\sigma}_{t_l}} \sum_{i=1}^{k} r(y + \bar{\sigma}_{t_l} Z_i) Z_i$, where $Z_i \sim \mathcal{N}(0, I)$ independently. Then, the guidance in SALD is defined as $\lambda\sigma_t^2 \hat{\nabla} f_{t_l}(y)$. For a fair comparison, we substract the leave-one-out baseline from the reward. Fig. 9 shows the results of varying the hyperparameter λ in SALD and compares them to our steepest guidance. We see that SALD cannot achieve the same level of performance as our steepest guidance method. In particular, the difference is large for ImageReward experiments. This may be because ImageReward is sensitive to fine-grained features that are lost through Gaussian smoothing.

L.10 Ablation Study on k.

We conduct an ablation study on the number of lookahead particles k. We vary $k \in \{4, 8, 16, 32\}$ while keeping λ fixed. We use the same λ values as in the main experiments, except that empirical

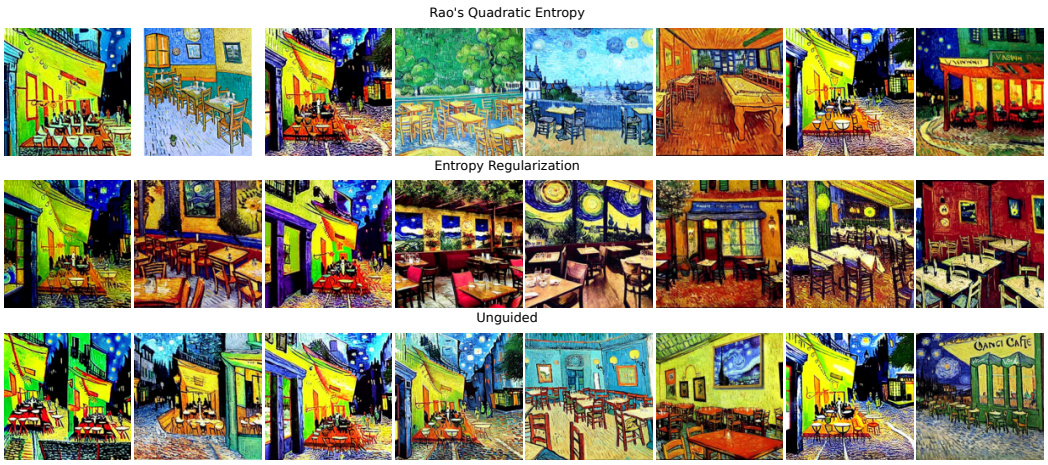

Figure 7: Comparison of diversity-seeking generation using SD. From top to bottom, the rows show Steepest Guidance with Rao's quadratic entropy, entropy regularization, and the unguided baseline.

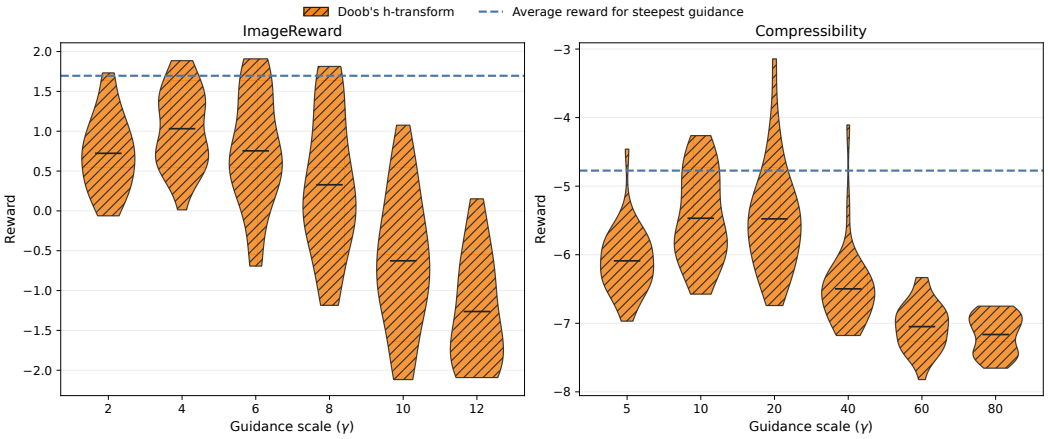

Figure 8: Ablation study on the guidance scaling factor γ for FLUX with ImageReward and Compressibility. Average performance is shown as horizontal lines.

Doob's h-transform uses $\lambda = 100$ for Compressibility. Fig. 10 shows that the performance improves as k increases, but the proposed method consistently outperforms the baselines for all k values.

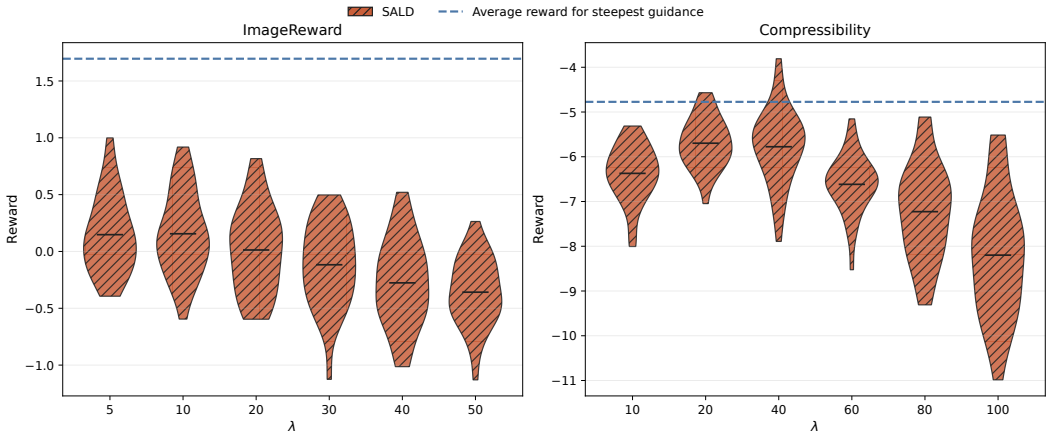

Figure 9: Comparison of the proposed steepest guidance method with SALD (Nitanda et al., 2026) for FLUX with ImageReward and Compressibility. Average performance is shown as horizontal lines.

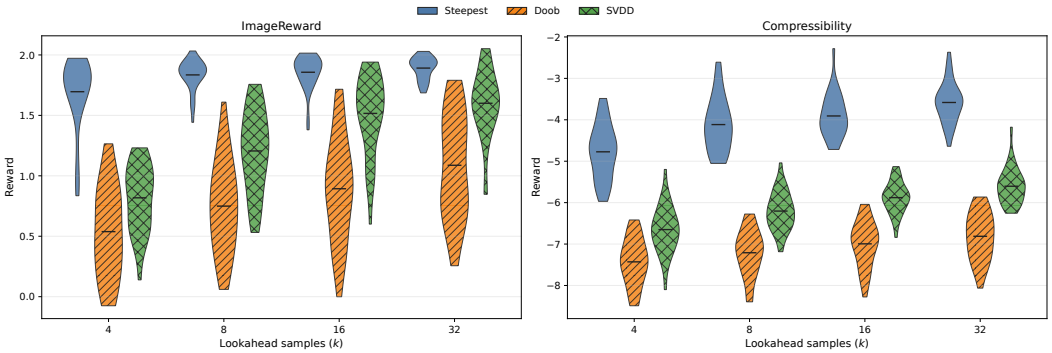

Figure 10: Ablation study on the number of lookahead particles k for FLUX with ImageReward and Compressibility. Average performance is shown as horizontal lines.