SCORE AS ACTION: FINE-TUNING DIFFUSION GENERATIVE MODELS BY CONTINUOUS-TIME REINFORCEMENT LEARNING

Hanyang Zhao[1†], **Haoxian Chen**[1], **Ji Zhang**[2], **David D. Yao**[1], **Wenpin Tang**[1]
[1]Columbia University, [2] Stony Brook University

ABSTRACT

Reinforcement learning from human feedback (RLHF), which aligns a diffusion model with input prompt, has become a crucial step in building reliable generative AI models. Most works in this area use a *discrete-time* formulation, which is prone to induced errors, and often not applicable to models with higher-order/black-box solvers. The objective of this study is to develop a disciplined approach to fine-tune diffusion models using *continuous-time* RL, formulated as a stochastic control problem with a reward function that aligns the end result (terminal state) with input prompt. The key idea is to treat score matching as controls or actions, and thereby making connections to policy optimization and regularization in continuous-time RL. To carry out this idea, we lay out a new policy optimization framework for continuous-time RL, and illustrate its potential in enhancing the value networks design space via leveraging the structural property of diffusion models. We validate the advantages of our method by experiments in downstream tasks of fine-tuning large-scale Text2Image models of Stable Diffusion v1.5.

1 INTRODUCTION

Diffusion models Sohl-Dickstein et al. (2015), with the capacity to turn a noisy/non-informative initial distribution into a desired target distribution through a well-designed denoising process Ho et al. (2020); Song et al. (2020; 2021b), have recently found applications in diverse areas such as high-quality and creative image generation Ramesh et al. (2022); Shi et al. (2020); Saharia et al. (2022); Rombach et al. (2022), video synthesis Ho et al. (2022), and drug design Xu et al. (2022). And, the emergence of human-interactive platforms like ChatGPT Ouyang et al. (2022) and Stable Diffusion Rombach et al. (2022) has further increased the demand for diffusion models to align with human preference or feedback.

To meet such demands, Hao et al. (2022) proposed a natural way to fine-tune diffusion models using reinforcement learning (RL, Sutton & Barto (2018)). Indeed, RL has already demonstrated empirical successes in enhancing the performance of LLM (large language models) using human feedback Christiano et al. (2017); Ouyang et al. (2022); Bubeck et al. (2023), and Fan & Lee (2023) is among the first to utilize RL-like methods to train diffusion models for better image synthesis. Moreover, Lee et al. (2023); Fan et al. (2023); Black et al. (2023) have improved the text-to-image (T2I) diffusion model performance by incorporating reward models to align with human preference (e.g., CLIP Radford et al. (2021), BLIP Li et al. (2022), ImageReward Xu et al. (2024)). Notably, all studies referenced above that combine diffusion models with RL are formulated as *discrete-time* sequential optimization problems, such as Markov decision processes (MDPs, Puterman (2014)), and solved by discrete-time RL algorithms like REINFORCE Sutton et al. (1999) or PPO Schulman et al. (2017).

Yet, diffusion models are intrinsically *continuous-time* as they were originally created to model the evolution of thermodynamics Sohl-Dickstein et al. (2015). Notably, the continuous-time formalism of diffusion models provides a unified framework for various existing discrete-time algorithms as shown in Song et al. (2021b): the denoising steps in DDPM Ho et al. (2020) can be viewed as a discrete approximation of a stochastic differential equation (SDE) and are implicitly *score-based*

[†]Corresponce to: {hz2684,yao,wt2319}@columbia.edu

under a specific variance-preserving SDE Song et al. (2021b); and DDIM Song et al. (2020), which underlies the success of Stable Diffusion Rombach et al. (2022), can also be seen as a numerical integrator of an ODE (ordinary differential equation) sampler Salimans & Ho (2022). Awareness of the continuous-time nature informs the design structure of the discrete-time SOTA large-scale T2I generative models (e.g.,Dhariwal & Nichol (2021); Rombach et al. (2022); Esser et al. (2024)), and enables simple controllable generations by classifier guidance to solve inverse problems Song et al. (2021b;a). It also motivates more efficient diffusion models with continuous-time samplers, including the ODE-governed probability (normalizing) flows Papamakarios et al. (2021); Song et al. (2021b) and rectified flows Liu et al. (2022; 2023) underpinning Stable Diffusion v3 Esser et al. (2024). A discrete-time formulation of RL algorithms for fine-tuning diffusion models, if/when directly applied to continuous-time diffusion models via discretization, can *nullify* the models' continuous nature and fail to capture or utilize their structural properties.

For fine-tuning diffusion models, discrete-time RL algorithms (such as DDPO Black et al. (2023)) require a prior chosen time discretization in sampling. We thus examine the robustness of a fine-tuned model to the inference time discretization, and observe an "overfitting" phenomenon as illustrated in Figure 1. Specifically, improvements observed during inference at alternative discretization timesteps (25 and 100) are significantly smaller than that of sampling timestep (50) in RL.

Figure 1: Reward curve of model checkpoints sampling under different timesteps (25, 50, 100): After training Stable Diffusion v1.4 for a fixed prompt with 60 training steps by DDPO with 50 discretization steps, the average reward of images generated by checkpoints obtained (under 50 discretization steps) evaluated by ImageReward increases by 0.046, while the average reward of images generated with 100 discretization steps only increases by less than 0.016.

In addition, for high-order solvers (such as 2$^{\text{nd}}$ order Heun in EDM Karras et al. (2022)), discrete-time RL methods will require solving a high-dimension root-finding problem for each inference step, which is inefficient in practice.

Main contributions. To address the above issues, we develop a unified continuous-time RL framework to fine-tune score-based diffusion models.

Our first contribution is a continuous-time RL framework for fine-tuning diffusion models by treating *score functions as actions*. This framework naturally accommodates discrete-time diffusion models with any solver as well as continuous-time diffusion models, and overcomes the afore-mentioned limitations of discrete-time RL methods. (See Section 3.)

Second, we illustrate the promise of leveraging the structural property of diffusion models to generate tractable optimization problems and to enhance the design space of value networks. This includes transforming the KL regularization to a tractable running reward over time, and a novel design of value networks that involves "sample prediction" by sharing parameters with policy networks and fine-tuned diffusion models. Through experiments, we demonstrate the drastic improvements over naive value network designs. We also provide a new theory for RL in continuous-time and space, which leads to the first scalable policy optimization algorithm for continuous-time RL.

1.1 Related Works

Other papers that relate to our work are briefly reviewed below.

Continuous-time RL. Wang et al. (2020) models the noise or randomness in the environment dynamics as following an SDE, and incorporates an entropy-based regularizer into the objective function to facilitate the exploration-exploitation tradeoff. Follow-up works include designing model-free methods and algorithms under either finite horizon Jia & Zhou (2022a;b; 2023) or infinite horizon Zhao et al. (2024).

Stochastic Control. Uehara et al. (2024), which also formulated the diffusion models alignment as a continuous-time stochastic control problem with a different parameterization of the control; Tang (2024) also provides a more rigorous review and discussion. Domingo-Enrich et al. (2024) proposes to use adjoint to solve a similar control problem. In a concurrent work to ours, Gao et al. (2024) uses

q-learning Jia & Zhou (2023) for inferring the score of diffusion models (instead of fine tuning a pretrained model).

2 PRELIMINARIES

2.1 CONTINUOUS-TIME RL

Diffusion Process. We consider the state space $\mathbb{R}^d$, and denote by $\mathcal{A}$ the action space. Let $\pi(\cdot \mid t, x)$ be a feedback policy given $t \in [0, T]$ and $x \in \mathbb{R}^d$. The state dynamics $(X_t^\pi,\, 0 \le t \le T)$ is governed by the following SDE:

$$\mathrm{d}X_t^\pi = b\left(t, X_t^\pi, a_t\right) \mathrm{d}t + \sigma(t) \mathrm{d}B_t, \quad X_0^\pi \sim \rho, \tag{1}$$

where $(B_t,\, t \ge 0)$ is a d-dimensional Brownian motion; $b : \mathbb{R}_+ \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ and $\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ [1] are given functions; the action a_t follows the distribution $\pi\left(\cdot \mid t, X_t^\pi\right)$ by external randomization; and ρ is the initial distribution over the state space.

Performance Metric. Our goal is to find the optimal feedback policy π^* that maximizes the expected reward over a finite time horizon:

$$V^* := \max_\pi \mathbb{E}\left[\int_0^T r\left(t, X_t^\pi, a_t^\pi\right) \mathrm{d}t + h(X_T^\pi) \mid X_0^\pi \sim \rho\right], \tag{2}$$

where $r : \mathbb{R}_+ \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ and $h : \mathbb{R}^d \to \mathbb{R}$ are the running and terminal rewards respectively. Given a policy $\pi(\cdot)$, let $\tilde{b}(t, x, \pi(\cdot)) := \int_{\mathcal{A}} b(t, x, a)\pi(a)\mathrm{d}a$. We consider the following equivalent representation of equation 1:

$$\mathrm{d}\tilde{X}_t = \tilde{b}\left(t, \tilde{X}_t, \pi(\cdot \mid t, \tilde{X}_t)\right) \mathrm{d}t + \sigma(t) \mathrm{d}\tilde{B}_t, \quad \tilde{X}_0 \sim \rho, \tag{3}$$

in the sense that there exists a probability measure $\tilde{\mathbb{P}}$ that supports a d-dimensional Brownian motion $(\tilde{B}_t,\, t \ge 0)$, and for each $t \ge 0$, the distribution of $\tilde{X}_t$ under $\tilde{\mathbb{P}}$ agrees with that of X_t under $\mathbb{P}$ defined by equation 1. The value function associated with the feedback policy $\{\pi(\cdot \mid t, x) : x \in \mathbb{R}^d\}$ is

$$V(t, x; \pi) := \mathbb{E}\left[\int_t^T r\left(s, X_s^\pi, a_s^\pi\right) \mathrm{d}s + h\left(X_T^\pi\right) \mid X_t^\pi = x\right] \tag{4}$$

The performance metric is $V^\pi := \int_{\mathbb{R}^d} V(0, x; \pi)\rho(dx)$, and $V^* := \max_\pi V^\pi$.

q-Value. Following the definition in Jia & Zhou (2023), given a policy π and $(t, x, a) \in [0, \infty) \times \mathbb{R}^n \times \mathcal{A}$, we construct a "perturbed" policy, denoted by $\hat{\pi}$: It takes the action $a \in \mathcal{A}$ on $[t, t + \Delta t)$, and then follows π on $[t + \Delta t, \infty)$. Specifically, the corresponding state process $X^{\hat{\pi}}$, given $X_t^{\hat{\pi}} = x$, breaks into two pieces: on $[t, t + \Delta t)$, it is X^a following equation 1 with $a_t \equiv a$ (i.e., $\pi(t, x, a) = 1$); while on $[t + \Delta t, \infty)$, it is X^π following (3) but with the initial time-state pair $\left(t + \Delta t, X_{t+\Delta t}^a\right)$. The q-value measures the rate of the performance difference between the two policies when $\Delta t \to 0$, and is shown in Jia & Zhou (2023) to take the following form:

$$q(t, x, a; \pi) = \frac{\partial V}{\partial t}(t, x; \pi) + \mathcal{H}\left(t, x, a, \frac{\partial V}{\partial x}(t, x; \pi), \frac{\partial^2 V}{\partial x^2}(t, x; \pi)\right), \tag{5}$$

where $\mathcal{H}(t, x, a, y, A) := b(t, x, a) \cdot y + \frac{1}{2}\sigma^2(t) \sum_i A_{ii} + r(t, x, a)$ is the (generalized) Hamilton function in stochastic control theory Yong & Zhou (1999).

2.2 SCORE-BASED DIFFUSION MODELS

Forward and Backward SDE. We follow the presentation in Tang & Zhao (2024). Consider the following SDE that governs the dynamics of a process $(X_t,\, 0 \le t \le T)$ in $\mathbb{R}^d$ Song et al. (2021b),

$$\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + g(t)\mathrm{d}B_t, \quad X_0 \sim p_{\text{data}}(\cdot), \tag{6}$$

[1] For our applications here we assume that the diffusion coefficient $\sigma(t)$ only depends on time t. Note, however, that the general continuous-time RL theory also holds for time-, state- and action-dependent $\sigma(t, x, a)$, see Jia & Zhou (2022a;b).

where $(B_t,\, t \geq 0)$ is a d-dimensional Brownian motion, $f : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}_+ \to \mathbb{R}_+$ are two given functions (up to the designer to choose), and the initial state X_0 follows a distribution with density $p_{\text{data}}(\cdot)$, which is shaped by data yet unknown *a priori*. Denote by $p_t(\cdot)$ the probability density of X_t.

Run the SDE in equation 6 until a given time $T > 0$, to obtain $X_T \sim p(T, \cdot)$. Next, consider the "time reversal" of X_t, denoted X_t^{rev}, such that the distribution of X_t^{rev} agrees with that of X_{T-t} on $[0, T]$. Then, $(X_t^{\text{rev}},\, 0 \leq t \leq T)$ satisfies the following SDE under mild conditions on f and g:

$$\mathrm{d}X_t^{\text{rev}} = \big(-f(T-t, X_t^{\text{rev}}) + g^2(T-t)\nabla \log p_{T-t}(X_t^{\text{rev}})\big)\,\mathrm{d}t + g(T-t)\mathrm{d}B_t, \tag{7}$$

where $\nabla \log p_t(x)$ is known as *Stein's score function*. Below we will refer to the two SDE's in equation 6 and equation 7, respectively, as the forward and the backward SDE.

For sampling from the backward SDE, we replace $p_T(\cdot)$ with some $p_{\text{noise}}(\cdot)$ as an approximation. The initialization $p_{\text{noise}}(\cdot)$ is commonly independent of $p_{\text{data}}(\cdot)$, which is the reason why diffusion models are known for generating data from "noise".

Inference Process. Once the best approximation $s_{\theta_{\text{pre}}}$ is obtained by e.g. score matching for the stein score function, we use it to replace $\nabla \log p_t(x)$ in equation 7. The corresponding approximation to the reversed process X_t^{rev}, denoted as $X_t^{\leftarrow}$, then follows the SDE:

$$\mathrm{d}X_t^{\leftarrow} = \big(-f(T-t, X_t^{\leftarrow}) + g^2(T-t)s_{\theta_{\text{pre}}}(T-t, X_t^{\leftarrow})\big)\,\mathrm{d}t + g(T-t)\mathrm{d}B_t, \tag{8}$$

with $X_0^{\leftarrow} \sim p_{\text{noise}}(\cdot)$. At time $t = T$, the distribution of $X_T^{\leftarrow}$ is expected to be close to $p_{\text{data}}(\cdot)$. The well-known DDPM Ho et al. (2020) can be viewed as a discretized version of the SDE in equation 8. This has been established in Song et al. (2021b); Salimans & Ho (2022); Zhang & Chen (2022); Zhang et al. (2022); also refer to further discussions in Appendix B. Throughout the rest of the paper, we will focus on the continuous formalism (via SDE).

3 Continuous-time RL for Diffusion Models Fine Tuning

Here we formulate the task of fine-tuning diffusion models as a continuous-time stochastic control problem, by treating score function approximation as a control process applied to the backward SDE.

Scores as Actions. First, to broaden the application context of the diffusion model, we add a parameter c to the score function, interpreted as a "class" index or label (e.g., for input prompts). Then, the backward SDE in equation 8 becomes:

$$\mathrm{d}X_t^{\leftarrow} = \big(-f(T-t, X_t^{\leftarrow}) + g^2(T-t)s_{\theta_{\text{pre}}}(T-t, X_t^{\leftarrow}, c)\big)\,\mathrm{d}t + g(T-t)\mathrm{d}B_t. \tag{9}$$

Next, comparing the continuous RL process in equation 3 and the inference process equation 9, we choose b and σ in the RL dynamics in equation 3 as:

$$b\,(t, x, a) := -f(T-t, x) + g^2(T-t)a, \quad \sigma(t) := g(T-t), \tag{10}$$

In the sequel, we will stick to this definition of b and σ. Define a specific feedback control, $a_t^{\theta_{\text{pre}}} = s_{\theta_{\text{pre}}}(T-t, X_t^{\leftarrow}, c)$, and the backward SDE in (9) is expressed as:

$$\mathrm{d}X_t^{\leftarrow} = b\left(t, X_t^{\leftarrow}, a_t^{\theta_{\text{pre}}}\right)\mathrm{d}t + \sigma(t)\mathrm{d}B_t. \tag{11}$$

This way, the score function is replaced by the action, and finding the optimal score becomes a policy optimization problem in RL. Denote by $p^{\theta_{\text{pre}}}(t, \cdot, c)$ the probability density of $X_t^{\leftarrow}$ in equation 11.

Exploratory SDEs. As we will deal with the time-reversed process $X_t^{\leftarrow}$ exclusively from now on, the superscript $^{\leftarrow}$ will be dropped to lighten the notation. To enhance exploration, we will use a Gaussian control:

$$a_t^{\theta} \sim \pi^{\theta}(\cdot \mid t, X_t^{\theta}, c) = N(\mu^{\theta}(t, X_t^{\theta}, c), \Sigma_t). \tag{12}$$

Specifically, the dependence on θ is through that of the mean function μ^{θ}, while the covariance matrix Σ_t only depends on time t, representing a chosen exploration level at t. For brevity, write X_t^{θ} for the (time-reversed) process $X_t^{\pi^{\theta}}$ driven by the policy π^{θ}. (We further denote by $p^{\theta}(t, \cdot, c)$ the probability density of X_t^{θ}.) Then $(X_t^{\theta},\, 0 \leq t \leq T)$ is governed by the SDE:

$$\mathrm{d}X_t^{\theta} = \big[-f(T-t, X_t^{\theta}) + g^2(T-t)\mu^{\theta}(t, X_t^{\theta}, c)\big]\,\mathrm{d}t + g(T-t)\mathrm{d}B_t, \quad X_0^{\theta} \sim \rho. \tag{13}$$

Objective Function. The objective function of the RL problem consists of two parts. The first part is the terminal reward, i.e., a given reward model (RM) that is a function of both X_T and c. For instance, if the task is T2I generation, then $\text{RM}(X_T, c)$ represents how well the generated image X_T aligns with the input prompt c. The second part is a penalty (i.e., regularization) term, which takes the form of the KL divergence between $p^\theta(T, \cdot, c)$ and its pretrained counterpart. This is similar in spirit to previous works on fine-tuning diffusion models by discrete-time RL, see e.g., Ouyang et al. (2022); Fan et al. (2023). As for exploration, note that it has been represented by the Gaussian noise in a_t^θ; refer to (12), and more on this below. So, here is the problem we want to solve:

$$\max_\theta \mathbb{E}\left[\text{RM}(c, X_T^\theta) - \beta\,\text{KL}\left(p^\theta(T, \cdot, c) \| p^{\theta_{pre}}(T, \cdot, c)\right)\right], \tag{14}$$

where $\beta > 0$ is a (given) penalty cost.

To connect the problem in equation 14 to the objective function of the RL model in equation 2, we need the following explicit expression for the KL divergence term in equation 14.

Theorem 3.1. *For any given c, the KL divergence between p^θ and $p^{\theta_{pre}}$ is:*

$$\text{KL}(p^\theta(T, \cdot, c) \| p^{\theta_{pre}}(T, \cdot, c)) = \mathbb{E}\int_0^T \frac{g^2(T-t)}{2} \|\mu^\theta(t, X_t^\theta, c) - \mu^{\theta_{pre}}(t, X_t^\theta, c)\|^2 \mathrm{d}t. \tag{15}$$

Proof Sketch. The full proof is given in Appendix C.1.

As a remark, it is important to use the "reverse"-KL divergence $\text{KL}\left(p^\theta(T, \cdot, c) \| p^{\theta_{pre}}(T, \cdot, c)\right)$, because it yields the expectation under the current policy π^θ that can be estimated from sample trajectories. By Theorem 3.1, the objective function in equation 14 is equivalent to the following:

$$\eta^\theta := \mathbb{E}\int_0^T \underbrace{-\frac{\beta}{2} g^2(T-t) \|\mu_t^\theta - \mu_t^{\theta_{pre}}\|^2}_{r(t, X_t^\theta, a_t^\theta)} \mathrm{d}t + \mathbb{E}\underbrace{\text{RM}(X_T^\theta, c)}_{h(X_T^\theta, c)}, \tag{16}$$

where we abbreviate $\mu^\theta(t, X_t^\theta, c)$ and $\mu^{\theta_{pre}}(t, X_t^\theta, c)$ by μ_t^θ and $\mu_t^{\theta_{pre}}$ respectively. Thus, maximizing the objective function in equation 14 aligns with the RL model formulated in equation 2. We can also define the corresponding value function as:

$$V^\theta(t, x; c) = \mathbb{E}\left[\int_t^T -\frac{\beta}{2} g^2(T-t) \|\mu_t^\theta - \mu_t^{\theta_{pre}}\|^2 \mathrm{d}t + \text{RM}(X_T^\theta, c) \mid X_t^\theta = x\right], \tag{17}$$

Value Network Design. We also adopt a function approximation to learn the value function (i.e., the critic). For the value function $V^\theta(t, x; c)$ associated with policy π^θ, there is the boundary condition:

$$V^\theta(T, x; c) = \mathbb{E}\left[\text{RM}(X_T^\theta, c) \mid X_T^\theta = x\right] = \text{RM}(x, c). \tag{18}$$

To meet this condition, we propose the following parametrization that leverages the structural property of diffusion models:

$$V^\theta(t, x; c) \approx \mathcal{V}_\phi^\theta(t, x; c) := \underbrace{c_{\text{skip}}(t) \cdot \text{RM}(\hat{x}_\theta(t, x, c))}_{\text{reward mean predictor}} + \underbrace{c_{\text{out}}(t) \cdot F_\phi(t, x, c)}_{\text{residual term corrector}}, \tag{19}$$

where $\mathcal{V}_\phi^\theta$ denotes the function family parameterized by (θ, ϕ), $\hat{x}_\theta(t, x, c) = \frac{1}{\alpha_t}\left(\sigma_t^2 s_\theta(t, x, c) + x\right)$, with α_t and σ_t being noise schedules of diffusion models (see Appendix B.2 for details). When $\theta = \theta_{\text{pre}}$, $\hat{x}_\theta$ predicts a denoised sample given the current x and the score estimate $s_\theta(t, x, c)$, which is known as *Tweedie's formula*. To treat the second term in equation 17, our intuition comes from that

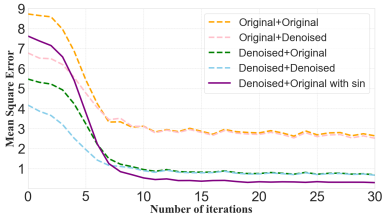
$$\text{RM}(\mathbb{E}(X_T \mid X_t)) \approx \mathbb{E}(\text{RM}(X_T) \mid X_t), \tag{20}$$

if we are allowed to exchange the conditional expectation and the reward model score (though generally it's not true). $F_\phi(t, x, c)$ are effectively approximations to the residual term, which can be seen as a composition of the possible reward error and the first term in equation 17.

We refer these two parts to as *reward mean predictor* and *residual corrector*. There $c_{\text{skip}}(t)$ and $c_{\text{out}}(t)$ are differentiable functions such that $c_{\text{skip}}(T) = 1$ and $c_{\text{out}}(T) = 0$, so the boundary condition

Architecture	Predictor	Corrector	c_{out}	MSE
Baseline	RM(x, c)	$F(x, c)$	1-cos$\left(\frac{\pi}{2T}t\right)$	2.63
Org+Denoised	RM(x, c)	$F(\hat{x}_\theta, c)$	1-cos$\left(\frac{\pi}{2T}t\right)$	2.51
Denoised+Orig	RM$(\hat{x}_\theta, c)$	$F(x, c)$	1-cos$\left(\frac{\pi}{2T}t\right)$	0.67
Denoised+Denoised	RM$(\hat{x}_\theta, c)$	$F(\hat{x}_\theta, c)$	1-cos$\left(\frac{\pi}{2T}t\right)$	0.66
Denoised+Orig	RM$(\hat{x}_\theta, c)$	$F(x, c)$	sin$\left(\frac{\pi}{2T}t\right)$	**0.29**

(a) Architecture configurations: $\hat{x}_\theta$ abbreviates $\hat{x}_\theta(t, x, c)$.



(b) Different Architecture MSE.

Figure 2: Architecture comparison and pretraining value function MSE.

equation 18 is satisfied. Notably, similar parametrization trick has also been used to train successful diffusion models such as EDM Karras et al. (2022) and consistency models Song et al. (2023).

For learning the value function, we use trajectory-wise Monte Carlo estimation to update ϕ by minimizing the mean square error (MSE). In our experiments, we observe that choosing $c_{\text{skip}}(t) = \cos(\frac{\pi}{2T}t)$ and $c_{\text{out}}(t) = \sin(\frac{\pi}{2T}t)$ yields the smallest loss (see Table 2a). Also refer to Section 4.2 for more architecture details.

Continuous-time Policy Optimization. To efficiently optimize the continuous-time RL problem raised above, we further develop the theory of policy optimization in continuous time and space for fine-tuning diffusion models. Different from the general formalism in the literature Schulman et al. (2015); Zhao et al. (2024), we focus on the case of (1) KL regularized rewards, and (2) state-independent diffusion coefficients in the continuous-time setup, which yield new results not only in the analysis but also in the resulting algorithms.

We show that the continuous-time policy gradient can be directly computed without any prior discretization of the time variable.

Theorem 3.2. *The gradient of an admissible policy π^θ parameterized by θ takes the form:*

$$\nabla_\theta V^\theta = \mathbb{E}\left[\int_0^T \nabla_\theta \log \pi^\theta(a_t^\theta | t, X_t^\theta) q(t, X_t^\theta, a_t^\theta; \pi^\theta) \mathrm{d}t\right], \tag{21}$$

where π^θ, a_t^θ and q are as defined in equation 12 and equation 5.

Note that the only terms in the q-value function that involve action a are (the second order term is irrelevant to action a):

$$g^2(T-t)a\frac{\partial V^\theta}{\partial x}(t, x) =: \tilde{q}^\theta(t, x, a).$$

In addition, the value function approximation can be computed by Monte Carlo or the martingale approach as in Jia & Zhou (2022a), and then $\frac{\partial V}{\partial x}$ can be evaluated by backward propagation. Since the reward can be non-differentiable, and also for the sake of efficient computation, we can approximate $\tilde{q}^\theta(t, x, a) \approx \left(V(t, x + \sigma\, g^2(T-t)a) - V(t, x)\right)/\sigma$, where σ is a scaling parameter. We further apply the same technique as in PPO Schulman et al. (2017) by clipping the ratio and replacing q with $\tilde{q}$ (which is equivalent to adapting a baseline function). This yields the policy update rule as:

$$\theta_{n+1} = \max_\theta \mathbb{E}\int_0^T \min\left(\rho_t^\theta q_t^{\theta_n}, \text{clip}\left(\rho_t^\theta, \epsilon\right) q_t^{\theta_n}\right) \mathrm{d}t, \tag{22}$$

where the advantage rate function and the likelihood ratio are defined by $q_t^{\theta_n} = \tilde{q}(t, X_t^{\theta_n}, a_t^{\theta_n}; \pi_n^\theta)$, $\rho_t^\theta = \frac{\pi^\theta(a_t^{\theta_n} | t, X_t^{\theta_n})}{\pi^{\theta_n}(a_t^{\theta_n} | t, X_t^{\theta_n})}$. The surrogate objective can then be optimized by stochastic gradient descent.

4 Experiments

4.1 Enhancing Small-Steps Diffusion Models

Setup. We evaluate the ability of our proposed algorithm to train short-run diffusion models with significantly reduced generation steps T, while maintaining high sample quality. In the experiment, we take $T = 10$. Our experiments are conducted on the CIFAR-10 (32×32) dataset Krizhevsky et al.

(2009). We fine-tune pretrained diffusion model backbone using DDPM Ho et al. (2020). The primary evaluation metric is the Fréchet Inception Distance (FID) Heusel et al. (2017), which measures the quality of generated samples.

To benchmark our method, we compare it against DxMI Yoon et al. (2024), which formulates the diffusion model training as an inverse reinforcement learning (IRL) problem. DxMI jointly trains a diffusion model and an energy-based model (EBM), where the EBM estimates the log data density and provides a reward signal to guide the diffusion process. To ensure a fair comparison, we replace the policy improvement step in DxMI with our continuous-time RL counterpart, maintaining consistency while evaluating the effectiveness of our approach. We set the learning rate of the value network to 2×10^{-5} and U-net to 3×10^{-7}.

Result. Figure 3 shows our approach converges significantly faster than DxMI, and achieves consistently lower FID scores throughout training. The samples from the two fine-tuned models are shown in Figures 4 and 5. In comparison, the samples generated from the model fine-tuned by continuous-time RL have clearer contours, better aligned with real-world features, and exhibit superior aesthetic quality.

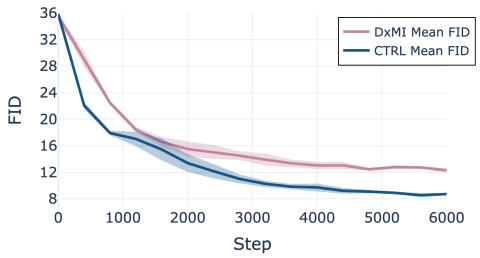


Figure 3: Training curves of DxMI and continuous-time RL.

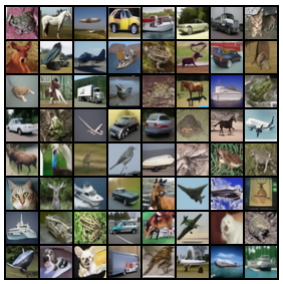


Figure 4: DxMI samples at the 6000-th step

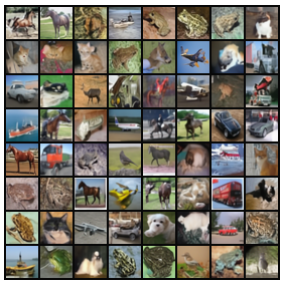


Figure 5: CTRL samples at the 6000-th step

4.2 Fine-Tuning Stable Diffusion

Setup. We also validate our proposed algorithm for fine-tuning large-scale T2I diffusion models, Stable Diffusion v1.5 [2]. We adopt the pretrained ImageReward Xu et al. (2024) as the reward signal during RL, as it has been shown in previous studies to achieve better alignment with human preferences to other metrics such as aesthetic scores, CLIP and BLIP scores.

We train the value networks with full parameter tuning, while we use LoRA Hu et al. (2021) for tuning the U-nets of diffusion models. We adopt a learning rate of 10^{-7} for optimizing the value network, 3×10^{-5} for optimizing the U-net and $\beta = 5 \times 10^{-5}$ for regularization. We train the models on 8 H200 GPUs with 128 effective batch sizes.

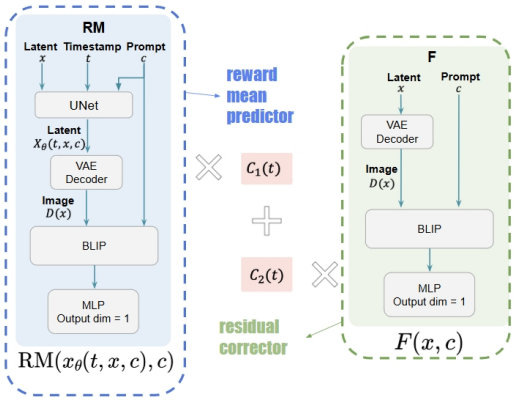
Value Network Architecture. Since we fix the reward model as ImageReward, we design the value network by using a similar backbone to the ImageReward model, which is composed of BLIP and a MLP header (see Figure 6a). To ensure the boundary condition, we fix the parameters (i.e., BLIP and MLP) in the left part (skyblue) of the value network and only tune 30% of the parameters of BLIP in the right part (green). The VAE Decoder on both parts is fixed for efficiency and stabilized training.

As a remark, replacing x with $x_\theta(t, x)$ in the "residual corrector" leads to minimum gain, compared to the drastic improvement brought forth by using $x_\theta(t, x)$ as the input in the "reward mean predictor". See Figure 6a and Table 2a for our ablation of network architecture and MSE statistics.

Policies trained by Continuous-time RL are robust to time discretization. We find that the policies trained by continuous-time RL achieve coherent performance in terms of the reward mean evaluated by ImageReward. In Figure 7a, three line plots that correspond to 25, 50, and 100 steps almost always overlap after 20 epochs of training, which is consistent with our theoretical analyses.

Qualitative examples with the same prompt of the base model, continuous-time RL training for 50 steps and 100 steps can be found in Figure 6b.

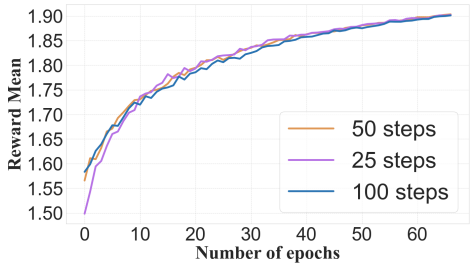
[2] https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5



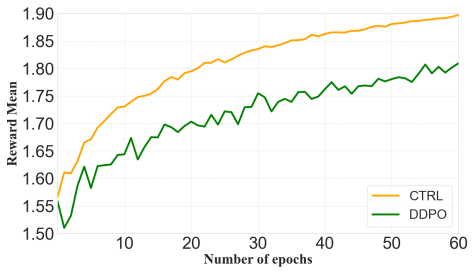
(a) We adopt the similar backbone of ImageReward for two parts in the value network, both by adding an MLP layer over the BLIP encoded latents.

(b) Model generations with prompt "A unicorn in a clearing. it has a single shining horn. volumetric light." a) **Top:** Base model Stable Diffusion v1.5; b) **Mid:** continuous-time RL after 50 training steps; c) **Bot:** continuous-time RL after 100 training steps.

Figure 6: (Left) Value network architecture; (Right) Model generations.



(a) Performance of continuous-time RL's checkpoints with respect to discretization timesteps.



(b) Performance of continuous-time RL against discrete-time RL under the same 50 discretization timesteps.

Figure 7: CTRL's performance vs discretization timesteps and Comparison of CTRL and DTRL.

Continuous-time RL outperforms Discrete-time RL baseline methods in both efficiency and stability. We also compare the reward curves of discrete-time RL with our continuous-time RL algorithms. In Figure 7b, the performance of the continuous-time RL is much more stable, and is more efficient in achieving a high average reward.

Why continuous-time approaches show better performance? Here we provide a heuristic explanation. Discrete-time RL methods optimize the objective with a priori time-discretization, which induces an error such that the resulting optimal policy can be significantly away from the true optimum in continuous time. Continuous-time RL methods, on the other hand, only require time-discretization in estimating the policy gradient. The error caused by this discretization — the gap between the resulting optimum and the true (continuous-time objective) optimum — is bounded by a polynomial of the step size (in gradient estimation) under suitable regularity conditions.

5 Discussion and Conclusion

We have proposed in this study a continuous-time reinforcement learning (RL) framework for fine-tuning diffusion models. Our work introduces novel policy optimization theory for RL in continuous time and space, alongside a scalable and effective RL algorithm that enhances the generation quality of diffusion models, as validated by our experiments.

In addition, our algorithm and network designs exhibit a striking versatility that allows us to incorporate and leverage some of the advantages of prior works in diffusion models design, so as to better exploit model structures and to improve value network architectures. In view of this, we believe the continuous-time RL, in providing cross-pollination between diffusion models and RLHF, presents a highly promising direction for future research.

REFERENCES

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2305.16381*, 2023.

Xuefeng Gao, Jiale Zha, and Xun Yu Zhou. Reward-directed score-based diffusion models via q-learning. *arXiv preprint arXiv:2409.04832*, 2024.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *arXiv preprint arXiv:2212.09611*, 2022.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Neurips*, volume 33, pp. 6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yanwei Jia and Xun Yu Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *J. Mach. Learn. Res.*, 23(154):1–55, 2022a.

Yanwei Jia and Xun Yu Zhou. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *J. Mach. Learn. Res.*, 23(275):1–50, 2022b.

Yanwei Jia and Xun Yu Zhou. q-learning in continuous time. *J. Mach. Learn. Res.*, 24(161):1–61, 2023.

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

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

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

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

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

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

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. *arXiv preprint arXiv:2006.11807*, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

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

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Wenpin Tang. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. *arXiv:2403.06279*, 2024.

Wenpin Tang and Hanyang Zhao. Score-based diffusion models via stochastic differential equations–a technical tutorial. *arXiv preprint arXiv:2402.07487*, 2024.

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.

Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *J. Mach. Learn. Res.*, 21(198):1–34, 2020.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.

Sangwoong Yoon, Himchan Hwang, Dohyun Kwon, Yung-Kyun Noh, and Frank C Park. Maximum entropy inverse reinforcement learning of diffusion models with energy-based models. *arXiv preprint arXiv:2407.00626*, 2024.

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

Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022.

Hanyang Zhao, Wenpin Tang, and David Yao. Policy optimization for continuous reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

A APPENDIX

B CONNECTION BETWEEN DISCRETE-TIME AND CONTINUOUS-TIME SAMPLER

In this section, we summarize the discussion of popular samplers like DDPM, DDIM, stochastic DDIM and their continuous-time limits being a Variance Preserving (VP) SDE.

B.1 DDPM SAMPLER IS THE DISCRETIZATION OF VP-SDE

We review the forward and backward process in DDPM, and its connection to the VP SDE following the discussion in Song et al. (2021b); Tang & Zhao (2024). DDPM considers a sequence of positive noise scales $0 < \beta_1, \beta_2, \cdots, \beta_N < 1$. For each training data point $x_0 \sim p_{\text{data}}(x)$, a discrete Markov chain $\{x_0, x_1, \cdots, x_N\}$ is constructed such that:

$$x_i = \sqrt{1-\beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}, \quad i = 1, \cdots, N, \tag{23}$$

where $z_{i-1} \sim \mathcal{N}(0, I)$, thus $p(x_i \mid x_{i-1}) = \mathcal{N}\left(x_i; \sqrt{1-\beta_i} x_{i-1}, \beta_i I\right)$. We can further think of x_i as the i^{th} point of a uniform discretization of time interval $[0, T]$ with discretization stepsize $\Delta t = \frac{T}{N}$, i.e. $x_{i\Delta t} = x_i$; and also $z_{i\Delta t} = z_i$. To obtain the limit of the Markov chain when $N \to \infty$, we define a function $\beta : [0, T] \to \mathbb{R}^+$ assuming that the limit exists: $\beta(t) = \lim_{\Delta t \to 0} \beta_i / \Delta t$ with $i = t/\Delta t$. Then when Δt is small, we get:

$$x_{t+\Delta t} \approx \sqrt{1-\beta(t)\Delta t} x_t + \sqrt{\beta(t)\Delta t} z_t \approx x_t - \frac{1}{2}\beta(t) x_t \Delta t + \sqrt{\beta(t)\Delta t} z_t.$$

Further taking the limit $\Delta t \to 0$, this leads to:

$$dX_t = -\frac{1}{2}\beta(t) X_t dt + \sqrt{\beta(t)} dB_t, \quad 0 \le t \le T,$$

and we have:

$$f(t, x) = -\frac{1}{2}\beta(t) x, g(t) = \sqrt{\beta(t)}.$$

Through reparameterization, we have $p_{\bar{\alpha}_i}(x_i \mid x_0) = \mathcal{N}\left(x_i; \sqrt{\bar{\alpha}_i} x_0, (1-\bar{\alpha}_i) I\right)$, where $\bar{\alpha}_i := \prod_{j=1}^{i} (1-\beta_j)$. For the backward process, a variational Markov chain in the reverse direction is parameterized with $p_\theta(x_{i-1} \mid x_i) = \mathcal{N}\left(x_{i-1}; \frac{1}{\sqrt{1-\beta_i}}\left(x_i + \beta_i s_\theta(i, x_i)\right), \beta_i I\right)$, and trained with a re-weighted variant of the evidence lower bound (ELBO):

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} (1-\bar{\alpha}_i) \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{p_{\bar{\alpha}_i}(\tilde{x} \mid x)} \left[\| s_\theta(i, \tilde{x}) - \nabla_{\tilde{x}} \log p_{\bar{\alpha}_i}(\tilde{x} \mid x) \|_2^2 \right].$$

After getting the optimal model $s_{\theta^*}(i, x)$, samples can be generated by starting from $x_N \sim \mathcal{N}(0, I)$ and following the estimated reverse Markov chain as:

$$x_{i-1} = \frac{1}{\sqrt{1-\beta_i}}\left(x_i + \beta_i s_{\theta^*}(i, x_i)\right) + \sqrt{\beta_i} z_i, \quad i = N, N-1, \cdots, 1. \tag{24}$$

Similar discussion as for the forward process, the equation equation 24 can further be rewritten as:

$$\begin{aligned} x_{(i-1)\Delta t} &\approx \frac{1}{\sqrt{1-\beta_{i\Delta t}\Delta t}}\left(x_{i\Delta t} + \beta(i\Delta t)\Delta t \cdot s_{\theta^*}(i\Delta t, x_{i\Delta t})\right) + \sqrt{\beta_i} z_i, \\ &\approx (1 + \frac{1}{2}\beta_{i\Delta t}\Delta t)\left(x_{i\Delta t} + \beta(i\Delta t)\Delta t \cdot s_{\theta^*}(i\Delta t, x_{i\Delta t})\right) + \sqrt{\beta_i} z_i, \\ &\approx (1 + \frac{1}{2}\beta_{i\Delta t}\Delta t) x_{i\Delta t} + \beta(i\Delta t)\Delta t \cdot s_{\theta^*}(i\Delta t, x_{i\Delta t}) + \sqrt{\beta_i} z_i, \end{aligned} \tag{25}$$

when $\beta_{i\Delta t}$ is small. This is indeed the time discretization of the backward SDE:

$$\begin{aligned} \mathrm{d}X_t^{\leftarrow} &= (\frac{1}{2}\beta(T-t) X_t^{\leftarrow} + \beta(T-t) s_{\theta^*}(T-t, X_t^{\leftarrow}))\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}B_t, \\ &= \left(-f(T-t, X_t^{\leftarrow}) + g^2(T-t) s_{\theta^*}(T-t, X_t^{\leftarrow})\right)\mathrm{d}t + g(T-t)\mathrm{d}B_t. \end{aligned} \tag{26}$$

B.2 DDIM SAMPLER IS THE DISCRETIZATION OF ODE

We review the backward process in DDIM, and its connection to the probability flow ODE following the discussion in Song et al. (2021b); Kingma et al. (2021); Salimans & Ho (2022); Zhang & Chen (2022).

(i) DDIM update rule: The concrete updated rule in DDIM paper (same as in the implementation) adopted the following rule (with $\sigma_t = 0$ in Equation (12) of Song et al. (2020)):

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{“predicted } x_0\text{”}} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{“direction pointing to } x_t\text{”}} \tag{27}$$

To show the correspondence between DDIM parameters and continuous-time SDE parameters, we follow one derivation in Salimans & Ho (2022) by considering the “predicted x_0”: note that define the predicted x_0 parameterization as:

$$\hat{x}_\theta(t,x) = \frac{x - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta^{(t)}(x)}{\sqrt{\bar{\alpha}_t}}, \text{ or }, \epsilon_\theta^{(t)}(x) = \frac{x - \sqrt{\bar{\alpha}_t}\hat{x}_\theta(t,x)}{\sqrt{1-\bar{\alpha}_t}},$$

above equation 27 can be rewritten as:

$$x_{t-1} = \frac{\sqrt{1-\bar{\alpha}_{t-1}}}{\sqrt{1-\bar{\alpha}_t}} \left(x_t - \sqrt{\bar{\alpha}_t}\hat{x}_\theta(t,x) \right) + \sqrt{\bar{\alpha}_{t-1}} \cdot \hat{x}_\theta(t,x) \tag{28}$$

Using parameterization $\sigma_t = \sqrt{1-\bar{\alpha}_t}$ and $\alpha_t = \sqrt{\bar{\alpha}_t}$, we have for $t-1 = s < t$:

$$X_s = \frac{\sigma_s}{\sigma_t}\left[X_t - \alpha_t \hat{x}_\theta(t, X_t) \right] + \alpha_s \hat{x}_\theta(t, X_t), \tag{29}$$

which is the same as derived in Kingma et al. (2021); Salimans & Ho (2022).

B.2.1 ODE EXPLANATION BY ANALYZING THE DERIVATIVE

We further assume a VP diffusion process with $\alpha_t^2 = 1 - \sigma_t^2 = \text{sigmoid}(\lambda_t)$ for $\lambda_t = \log\left[\alpha_t^2/\sigma_t^2\right]$, in which λ_t is known as the signal-to-noise ratio. Taking the derivative of equation 29 with respect to λ_s, assuming again a variance preserving diffusion process, and using $\frac{d\alpha_\lambda}{d\lambda} = \frac{1}{2}\alpha_\lambda\sigma_\lambda^2$ and $\frac{d\sigma_\lambda}{d\lambda} = -\frac{1}{2}\sigma_\lambda\alpha_\lambda^2$, gives

$$\begin{aligned}
\frac{X_{\lambda_s}}{d\lambda_s} &= \frac{d\sigma_{\lambda_s}}{d\lambda_s}\frac{1}{\sigma_t}\left[X_t - \alpha_t\hat{x}_\theta(t,X_t) \right] + \frac{d\alpha_{\lambda_s}}{d\lambda_s}\hat{x}_\theta(t,X_t) \\
&= -\frac{1}{2}\alpha_s^2\frac{\sigma_s}{\sigma_t}\left[X_t - \alpha_t\hat{x}_\theta(t,X_t) \right] + \frac{1}{2}\alpha_s\sigma_s^2\hat{x}_\theta(t,X_t).
\end{aligned}$$

Evaluating this derivative at $s = t$ then gives

$$\begin{aligned}
\left.\frac{X_{\lambda_s}}{d\lambda_s}\right|_{s=t} &= -\frac{1}{2}\alpha_\lambda^2\left[X_\lambda - \alpha_\lambda\hat{x}_\theta(t,X_\lambda) \right] + \frac{1}{2}\alpha_\lambda\sigma_\lambda^2\hat{x}_\theta(t,X_\lambda) \\
&= -\frac{1}{2}\alpha_\lambda^2\left[X_\lambda - \alpha_\lambda\hat{x}_\theta(t,X_\lambda) \right] + \frac{1}{2}\alpha_\lambda\left(1-\alpha_\lambda^2\right)\hat{x}_\theta(t,X_\lambda) \\
&= \frac{1}{2}\left[\alpha_\lambda\hat{x}_\theta(t,X_\lambda) - \alpha_\lambda^2 X_\lambda \right].
\end{aligned} \tag{30}$$

Recall that the forward process in terms of an SDE is defined as:

$$\mathrm{d}X_t = f(t,X_t)\mathrm{d}t + g(t)\mathrm{d}B_t, \quad t \in [0,T]$$

and Song et al. (2021b) shows that backward of this diffusion process is an SDE, but shares the same marginal probability density of an associated probability flow ODE (by taking $t := T - t$) :

$$\mathrm{d}X_t = \left[f(t,X_t) - \frac{1}{2}g^2(t)\nabla_x \log p(t,X_t) \right]\mathrm{d}t, \quad t \in [T,0]$$

where in practice $\nabla_x \log p(t,x)$ is approximated by a learned denoising model using

$$\nabla_x \log p(t,x) \approx s_\theta(t,x) = \frac{\alpha_t \hat{x}_\theta(t,x) - x}{\sigma_t^2} = -\frac{\epsilon_\theta^{(t)}(x)}{\sigma_t}. \tag{31}$$

with two chosen noise scheduling parameters α_t and σ_t, and corresponding drift term $f(t,x) = \frac{d\log\alpha_t}{dt}x_t$ and diffusion term $g^2(t) = \frac{d\sigma_t^2}{dt} - 2\frac{d\log\alpha_t}{dt}\sigma_t^2$.

Further assuming a VP diffusion process with $\alpha_t^2 = 1 - \sigma_t^2 = \text{sigmoid}(\lambda_t)$ for $\lambda_t = \log\left[\alpha_t^2/\sigma_t^2\right]$, we get

$$f(t,x) = \frac{d\log\alpha_t}{dt}x = \frac{1}{2}\frac{d\log\alpha_\lambda^2}{d\lambda}\frac{d\lambda}{dt}x = \frac{1}{2}\left(1-\alpha_t^2\right)\frac{d\lambda}{dt}x = \frac{1}{2}\sigma_t^2\frac{d\lambda}{dt}x.$$

Similarly, we get

$$g^2(t) = \frac{d\sigma_t^2}{dt} - 2\frac{d\log\alpha_t}{dt}\sigma_t^2 = \frac{d\sigma_\lambda^2}{d\lambda}\frac{d\lambda}{dt} - \sigma_t^4\frac{d\lambda}{dt} = \left(\sigma_t^4 - \sigma_t^2\right)\frac{d\lambda}{dt} - \sigma_t^4\frac{d\lambda}{dt} = -\sigma_t^2\frac{d\lambda}{dt}.$$

Plugging these into the probability flow ODE then gives

$$\begin{aligned} \mathrm{d}X_t &= \left[f(t,X_t) - \frac{1}{2}g^2(t)\nabla_x \log p(t,x)\right]\mathrm{d}t \\ &= \frac{1}{2}\sigma_t^2\left[X_t + \nabla_x \log p(t,X_t)\right]\mathrm{d}\lambda_t. \end{aligned} \tag{32}$$

Plugging in our function approximation from Equation equation 31 gives

$$\begin{aligned} \mathrm{d}X_t &= \frac{1}{2}\sigma_t^2\left[X_t + \left(\frac{\alpha_t\hat{x}_\theta(t,X_t) - X_t}{\sigma_t^2}\right)\right]\mathrm{d}\lambda_t \\ &= \frac{1}{2}\left[\alpha_t\hat{x}_\theta(t,X_t) + \left(\sigma_t^2 - 1\right)X_t\right]\mathrm{d}\lambda_t \\ &= \frac{1}{2}\left[\alpha_t\hat{x}_\theta(t,X_t) - \alpha_t^2 X_t\right]\mathrm{d}\lambda_t. \end{aligned} \tag{33}$$

Comparison this with Equation equation 30 now shows that DDIM follows the probability flow ODE up to first order, and can thus be considered as an integration rule for this ODE.

B.2.2 Exponential Integrator Explanation

In Zhang & Chen (2022) that the integration role above is referred as "exponential integrator" of equation 33. We adopt two ways of derivations:

(a) Notice that, if we treat the $\hat{x}_\theta(t,X_t)$ as a constant in equation 33 (or assume that it does not change w.r.p. t along the ODE trajectory), we have:

$$\mathrm{d}X_t + \frac{1}{2}\alpha_t^2 X_t \mathrm{d}\lambda_t = \hat{x}_\theta(t,X_t)\cdot\frac{1}{2}\alpha_t\mathrm{d}\lambda_t. \tag{34}$$

Both sides multiplied by $1/\sigma_t$ and integrate from t to s yields:

$$\frac{X_s}{\sigma_s} - \frac{X_t}{\sigma_t} = \hat{x}_\theta(t,X_t)\cdot\left(\exp(\frac{1}{2}\lambda_s) - \exp(\frac{1}{2}\lambda_t)\right) = \hat{x}_\theta(t,X_t)\cdot\left(\frac{\alpha_s}{\sigma_s} - \frac{\alpha_t}{\sigma_t}\right). \tag{35}$$

which is thus

$$X_s = \frac{\sigma_s}{\sigma_t}X_t + \left[\alpha_s - \alpha_t\frac{\sigma_s}{\sigma_t}\right]\hat{x}_\theta(t,X_t), \tag{36}$$

which is the same as DDIM continuous-time intepretation as in equation 29.

(b) We also notice that we can also simplify the whole proof by treating the scaled score (same as in Zhang & Chen (2022)):

$$\sigma_t\nabla_x \log p(t,x) \approx \sigma_t s_\theta(t,x) = \frac{\alpha_t\hat{x}_\theta(t,x) - x}{\sigma_t} \tag{37}$$

as a constant in equation 32 (or assume that it does not change w.r.p. t along the ODE trajectory). Notice that from backward ODE, we have:

$$\mathrm{d}X_t = \frac{1}{2}\sigma_t^2 \left[X_t + \frac{1}{\sigma_t}\sigma_t \nabla_x \log p(t, X_t) \right] \mathrm{d}\lambda_t. \tag{38}$$

Both sides multiplied by $1/\alpha_t$ and integrate from t to s yields:

$$\frac{X_s}{\alpha_s} - \frac{X_t}{\alpha_t} = \left(\frac{\alpha_t \hat{x}_\theta\left(t, X_t\right) - X_t}{\sigma_t} \right) \cdot \left(-\frac{\sigma_s}{\alpha_s} + \frac{\sigma_t}{\alpha_t} \right). \tag{39}$$

which is thus

$$X_s = \frac{\sigma_s}{\sigma_t} X_t + \left[\alpha_s - \alpha_t \frac{\sigma_s}{\sigma_t} \right] \hat{x}_\theta\left(t, X_t\right), \tag{40}$$

which is the same as DDIM continuous-time intepretation as in equation 29.

As a summary, treating the denoised mean or the noise predictor as the constants will both recovery the rule of DDIM. Usually, for ODE flows, the denoised mean assumption naturally holds; however, why the scaled score leads to the same integration rule remains to be an interesting question, probably comes from the design property of DDIM, see e.g. discussions in Karras et al. (2022).

C Theorem Proofs

C.1 Proof of Theorem 3.1

The main proof technique relies on Girsanov's Theorem, which is similar to the argument in Chen et al. (2022). First, we recall a consequence of Girsanov's theorem that can be obtained by combining Pages 136-139, Theorem 5.22, and Theorem 4.13 of Le Gall (2016).

Theorem C.1. *For $t \in [0, T]$, let $\mathcal{L}_t = \int_0^t b_s \, \mathrm{d}B_s$ where B is a Q-Brownian motion. Assume that $\mathbb{E}_Q \int_0^T \|b_s\|^2 \, \mathrm{d}s < \infty$. Then, $\mathcal{L}$ is a Q-martingale in $L^2(Q)$. Moreover, if*

$$\mathbb{E}_Q \mathcal{E}(\mathcal{L})_T = 1, \quad \textit{where } \mathcal{E}(\mathcal{L})_t := \exp\left(\int_0^t b_s \, \mathrm{d}B_s - \frac{1}{2} \int_0^t \|b_s\|^2 \, \mathrm{d}s \right), \tag{41}$$

then $\mathcal{E}(\mathcal{L})$ is also a Q-martingale, and the process

$$t \mapsto B_t - \int_0^t b_s \, \mathrm{d}s \tag{42}$$

is a Brownian motion under $P := \mathcal{E}(\mathcal{L})_T Q$, the probability distribution with density $\mathcal{E}(\mathcal{L})_T$ w.r.t. Q.

If the assumptions of Girsanov's theorem are satisfied (i.e., the condition equation 41), we can apply Girsanov's theorem to Q as the law of the following reverse process (we omit c for brevity),

$$\mathrm{d}\overline{X}_t = \left(-f(T-t, \overline{X}_t) + g^2(T-t) s_{\theta_{pre}}(T-t, \overline{X}_t) \right) \mathrm{d}t + g(T-t) \mathrm{d}B_t, \ \overline{X}_0 \sim p_\infty(\cdot) \tag{43}$$

and

$$b_t = g(T-t) \left[s_\theta(T-t, \overline{X}_t) - s_{\theta_{pre}}(T-t, \overline{X}_t) \right], \tag{44}$$

where $t \in [0, T]$. This tells us that under $P = \mathcal{E}(\mathcal{L})_T Q$, there exists a Brownian motion $(\beta_t)_{t \in [0,T]}$ s.t.

$$\mathrm{d}B_t = g(T-t) \left[s_\theta(T-t, \overline{X}_t) - s_{\theta_{pre}}(T-t, \overline{X}_t) \right] \mathrm{d}t + \mathrm{d}\beta_t. \tag{45}$$

Plugging equation 45 into equation 43 we have P-a.s.,

$$\mathrm{d}\overline{X}_t = \left(-f(T-t, \overline{X}_t) + g^2(T-t) s_\theta(T-t, \overline{X}_t) \right) \mathrm{d}t + g(T-t) \mathrm{d}\beta_t, \ \overline{X}_0 \sim p_\infty(\cdot) \tag{46}$$

In other words, under P, the distribution of $\overline{X}$ is the same as the distribution generated by current policy parameterized by θ, i.e., $p_\theta(\cdot) = P_T = \mathcal{E}(\mathcal{L})_T Q$. Therefore,

$$\begin{aligned}
&D_{KL}\left(p_\theta \| p_{\theta_{pre}}\right) = \mathbb{E}_{P_T} \ln \frac{\mathrm{d}P_T}{\mathrm{d}Q_T} = \mathbb{E}_{P_T} \ln \mathcal{E}(\mathcal{L})_T \\
&= \mathbb{E}_{P_T}\left[\int_0^t b_s \, \mathrm{d}B_s - \frac{1}{2}\int_0^t \|b_s\|^2\right] \\
&= \mathbb{E}_{P_T}\left[\int_0^t b_s \, \mathrm{d}\beta_s + \frac{1}{2}\int_0^t \|b_s\|^2\right] \\
&= \frac{1}{2}\int_0^t g^2(T-t) \underbrace{\mathbb{E}_P \left\| s_\theta(T-t, \overline{X}_t) - s_{\theta_{pre}}(T-t, \overline{X}_t)\right\|^2}_{\epsilon_t^2} \mathrm{d}t
\end{aligned}$$

Thus we can bound the discrepancy between distribution generated by the policy θ and the pretrained parameters θ_{pre} as

$$D_{KL}(p_\theta \| p_{\theta_{pre}}) \leq \frac{1}{2}\int_0^T g^2(T-t)\epsilon_t^2 \mathrm{d}t \tag{47}$$

C.2 Proof of Theorem 3.2

First we include the policy gradient formula theorem for finite horizon in continuous time from Jia & Zhou (2022b):

Lemma C.2 (Theorem 5 of Jia & Zhou (2022b) when $R \equiv 0$)**.** *Under some regularity conditions, given an admissible parameterized policy π_θ, the policy gradient of the value function $V\left(t, x; \pi^\theta\right)$ admits the following representation:*

$$\begin{aligned}
\frac{\partial}{\partial \theta} V(t, x; \pi^\theta) =& \mathbb{E}^{\mathbb{P}}\Bigg[\int_t^T e^{-\beta(s-t)}\Bigg\{\frac{\partial}{\partial \theta}\log \pi^\theta(a_s^{\pi^\theta} | s, X_s^{\pi^\theta})\Big(\mathrm{d}V(s, X_s^{\pi^\theta}; \pi^\theta) \\
&+ \left[r_R(s, X_s^{\pi^\theta}, a_s^{\pi^\theta}) - \beta V(s, X_s^{\pi^\theta}; \pi^\theta)\right]\mathrm{d}s\Big)\Bigg\} \mid X_t^{\pi^\theta} = x\Bigg], \quad (t, x) \in [0, T] \times \mathbb{R}^d
\end{aligned} \tag{48}$$

in which we denote the regularized reward

$$r_R(t, X_t^{\pi^\theta}, a_t^{\pi^\theta}) = \gamma(t)\|a_t^{\pi^\theta} - s^{\theta^*}(t, X_t)\|^2.$$

First, by applying Itô's formula to $V(t, X_t)$, we have:

$$\mathrm{d}V(t, X_t) = \left[\frac{\partial V}{\partial t}(t, X_t) + \frac{1}{2}\sigma(t)^2 \circ \frac{\partial^2 V}{\partial x^2}(t, X_t)\right]\mathrm{d}t + \frac{\partial V}{\partial x}(t, X_t)\mathrm{d}X_t. \tag{49}$$

Further recall that:

$$q(t, x, a; \pi) = \frac{\partial V}{\partial t}(t, x; \pi) + \mathcal{H}\left(t, x, a, \frac{\partial V}{\partial x}(t, x; \pi), \frac{\partial^2 V}{\partial x^2}(t, x; \pi)\right) - \beta V(t, x; \pi), \tag{50}$$

this implies that (similar discussion also appeared in Jia & Zhou (2023))

$$q\left(t, X_t^\pi, a_t^\pi; \pi\right)\mathrm{d}t = \mathrm{d}J\left(t, X_t^\pi; \pi\right) + r\left(t, X_t^\pi, a_t^\pi\right)\mathrm{d}t - \beta J\left(t, X_t^\pi; \pi\right)\mathrm{d}t + \{\cdots\}\mathrm{d}B_t. \tag{51}$$

Plug this equality back in equation 48 yields:

$$\frac{\partial}{\partial \theta} V(t, x; \pi^\theta) = \mathbb{E}^{\mathbb{P}}\left[\int_t^T e^{-\beta(s-t)}\frac{\partial}{\partial \theta}\log \pi^\theta(a_s^{\pi^\theta} | s, X_s^{\pi^\theta}) q\left(t, X_t^\pi, a_t^\pi; \pi\right)\mathrm{d}s \mid X_t^{\pi^\theta} = x\right], \tag{52}$$

Let $t = 0$, $\beta = -\alpha$ and further taking expectation to the initial distribution yields Theorem 3.2.