ART for Diffusion Sampling: Continuous-Time Control and Actor–Critic Learning

Yilie Huang* Wenpin Tang† Xun Yu Zhou‡

June 30, 2026

Abstract

We study timestep allocation for score-based diffusion sampling, where a learned reverse-time dynamics is discretized on a finite grid. Uniform and hand-crafted schedules are standard choices, but they rely on *ad hoc*, fixed prescriptions and can therefore be suboptimal. To address this limitation, we propose Adaptive Reparameterized Time (ART), a continuous-time control formulation that learns a time change by treating the speed of the sampling clock as the control, so that a uniform grid on the learned clock induces adaptive timesteps in the original diffusion time. Based on a leading-order Euler error surrogate, ART provides a principled objective for allocating timesteps along the sampling trajectory. To solve this potentially high-dimensional deterministic control problem, we introduce ART-RL, an auxiliary randomized formulation with Gaussian policies that turns schedule learning into a continuous-time reinforcement learning problem. We prove that ART-RL is equivalent to ART at optimality, in the sense that the mean of the former's optimal Gaussian policy is optimal for the latter. We further establish policy evaluation and policy improvement characterizations and derive trajectory-based moment identities that yield implementable actor–critic updates for solving ART-RL. We conduct experiments ranging from controlled low-dimensional settings to image generation, and show that ART-learned schedules, when plugged into existing diffusion samplers by changing only the timestep grid, consistently improve sample quality over strong baseline schedules at matched budgets. The learned schedules also exhibit broad generalizability with superior performances, transferring without retraining across sampling budgets, datasets, solvers, pipelines, and representation spaces.

Key words: generative AI, diffusion model, sampling, adaptive reparameterized time, optimal control, reinforcement learning, distillation, transfer learning

*Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong. Email: yilie.huang@polyu.edu.hk.

†Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA. Email: wt2319@columbia.edu.

‡Department of Industrial Engineering and Operations Research and Data Science Institute, Columbia University, New York, NY 10027, USA. Email: xz2574@columbia.edu.

1 Introduction

Diffusion models (Ho et al., 2020; Song and Ermon, 2019; Song et al., 2021b) generate samples by transforming noise into data, thereby producing draws from a target distribution learned from examples. They now underpin a broad range of modern generative systems, including text-to-image models such as DALL·E 2 (Ramesh et al., 2022) and Stable Diffusion (Rombach et al., 2022), text-to-video generators such as Sora (OpenAI, 2024), Make-A-Video (Singer et al., 2023) and Veo (Google, 2024) and, more recently, diffusion-based large language models such as Mercury (Khanna et al., 2025) and LLaDA (Nie et al., 2025).

A diffusion pipeline typically separates *training* from *sampling*: the score or denoising model is learned during pretraining, and inference generates samples by running a reverse-time dynamics with a numerical discretization. This paper focuses on the sampling stage, where one must choose a finite set of timesteps to discretize the learned reverse-time process. Because each step requires evaluating the learned model, the choice of time grids directly dictates how a fixed computational budget is spent and can substantially affect both efficiency and sample quality. Most existing approaches adopt uniform grids or hand-crafted schedules (Song et al., 2021a,b; Karras et al., 2022; Chen et al., 2023a; Lu et al., 2022), but these choices are rarely derived from a *principled* optimization framework. Our goal is to provide a control-theoretic framework and approach that treat timestep selection as a systematic design problem for diffusion sampling.

The main contributions of this paper are summarized below:

- *Methodology*: We formulate timestep allocation for diffusion sampling as a continuous-time optimal control problem, termed *Adaptive Reparameterized Time* (ART). ART introduces a time change called the sampling clock and models the local progression in diffusion time as a controllable rate, which reallocates function evaluations along the reverse sampling trajectory while respecting a fixed overall time budget. To solve the resulting control problem, which is inherently in high-dimensional state spaces in most applications including image generation, we develop *ART-RL*, a continuous-time reinforcement learning (CTRL) approach that learns this rate via randomized, *Gaussian* policies and actor–critic iterations, leveraging recent theoretical advances in CTRL (Wang et al., 2020; Jia and Zhou, 2022a,b).

- *Theory*: We establish a rigorous link between ART and its randomized counterpart ART-RL. First, we show that the auxiliary randomized formulation is not merely a relaxation: it aligns with the original deterministic ART objective in that the mean of the optimal ART-RL Gaussian policy solves the ART control problem. Second, we develop continuous-time actor–critic theory specialized to time reparameterization, including characterizations of policy evaluation and policy improvement that yield explicit, implementable update rules. These results lead to moment conditions for both the critic and the actor and provide improvement guarantees that underpin the resulting learning algorithm for the optimal time schedule.

- *Experiments*: We evaluate ART across low- and high-dimensional settings, multiple numerical solvers, sampling pipelines, representation spaces, and a broad range of sampling budgets. In controlled ex-

periments with an analytical score model, in MNIST with a deliberately small score network, and in the standard EDM pipeline for CIFAR–10, ART consistently improves over Uniform, DPM, and EDM schedules at matched evaluation budgets, including the largest budgets where the hand-designed EDM schedule is the strongest. See a quick overview of these empirical gains in Figure 3. All comparisons keep the trained score model, network backbone, solver, and sampling protocol fixed; so the gains are purely from our choice of timestep allocation. This makes ART a principled schedule-learning method rather than an *ad hoc* one tied to a particular architecture, sampler, or diffusion pipeline.

- *Transfer/Generalization*: Our experiments show that the schedule trained on CIFAR–10 under a given number of time step budget transfers directly, without retraining, across timestep counts, datasets, sampling pipelines, and representation spaces. It outperforms the benchmarks not only on AFHQv2, FFHQ, and ImageNet–64 under the pixel-space EDM pipeline, but also on ImageNet–512 under EDM2 which simultaneously involves a modern backbone and sampling pipeline, a latent-space representation, and high-resolution image generation. These results demonstrate that ART-RL learns a reusable timestep schedule rather than a dataset-specific tuning artifact; its one-time training cost can therefore be amortized beyond the particular dataset, solver, and pipeline on which it is learned.

To our best knowledge, this is the first work that develops a control theory based framework for learning timestep schedules in generative diffusion sampling, providing a theoretically grounded alternative to the existing fixed heuristic grids. The proposed ART-RL method is purely data-driven and learns a reusable schedule that improves both direct sampling performance and transfer performance.

Relevant literature: Diffusion models were first developed in discrete time, including DDPM (Ho et al., 2020) and DDIM (Song et al., 2021a). The continuous-time viewpoint of Song et al. (2021b) recasts diffusion modeling through a stochastic differential equation (SDE) formulation, unifying and extending earlier discrete constructions. On the inference side, many samplers can be interpreted as numerical methods for the learned reverse-time dynamics, including the predictor-corrector scheme of Song et al. (2021b), exponential-integrator style approaches (Zhang and Chen, 2023), and higher-order solvers (Zhang et al., 2023; Wu et al., 2024). In addition, convergence analyses for diffusion inference under uniform or hand-crafted discretizations are studied in Lee et al. (2022); Chen et al. (2023a,b); Li et al. (2024); Benton et al. (2024); Li and Yan (2024); Huang et al. (2025a).

Continuous-time reinforcement learning (CTRL) was introduced and formulated by Wang et al. (2020) as entropy-regularized stochastic control in continuous time and spaces, where exploration is represented by *relaxed* (randomized) controls, formalizing the trial-and-error mechanism central to reinforcement learning. Following Wang et al. (2020), a sequence of works have built a model-free theory for CTRL through martingale-based analyses (Jia and Zhou, 2022a,b, 2023; Tang and Zhou, 2024), complemented by explicit performance guarantees (Huang et al., 2024, 2025b). A related study also investigates policy optimization in the continuous-time setting (Zhao et al., 2023). CTRL theory has been applied to financial portfolio selection

(Huang et al., 2024; Dai et al., 2023) and fine-tuning generative AI diffusion models (Gao et al., 2025; Zhao et al., 2024, 2025). In particular, Dai et al. (2023) exploit the special structure of the Merton problem with power utility and introduce a family of Gaussian policies without considering entropy regularization. They then prove the mean of the optimal Gaussian policy solves the original problem. This idea indeed inspires the formulation of ART-RL in this paper, even though the application domain and problem setting are very different here.

Organization of the paper: Section 2 reviews score-based diffusion and probability flow ODE sampling. Section 3 introduces ART as a time-reparameterization control formulation. Section 4 presents ART-RL as a randomized auxiliary problem, with a provable connection to ART. Section 5 develops the theory and an actor–critic algorithm for solving ART-RL, including policy evaluation and policy improvement. Section 6 reports the experimental results. Section 7 concludes. Additional numerical results are placed in the appendix.

2 Revisiting Continuous-Time Score-Based Diffusion Models

We briefly revisit continuous-time score-based diffusion models for generative AI (GenAI) along with key notations; see Tang and Zhao (2025) for a detailed exposition. A diffusion model consists of a forward diffusion process that progressively corrupts data with an unknown target distribution over a physical time interval $\tau \in [0, T]$, driving the distribution toward a simple reference law (e.g. Gaussian), and a backward generative process that transports samples from this reference back toward the original target distribution (Figure 1).

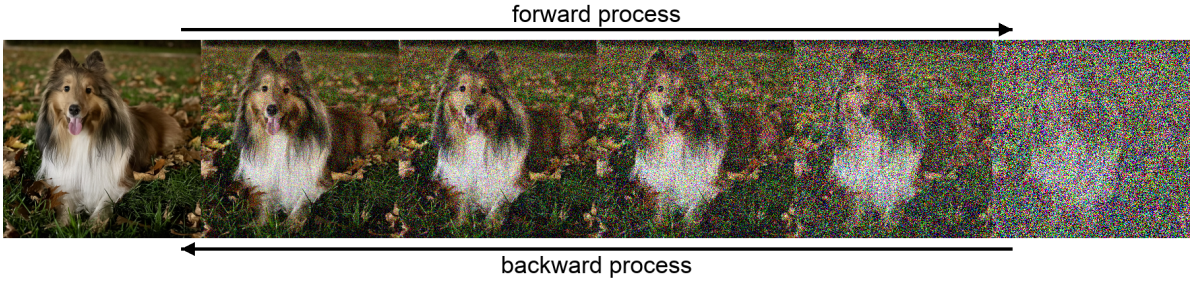


Figure 1: Illustration of the forward noising process and the corresponding backward generative process in a score-based diffusion model.

Forward diffusion. The forward dynamics are given by the Itô SDE

$$\mathrm{d}\,\bar{x}(\tau) = -f(\tau)\bar{x}(\tau)\,\mathrm{d}\tau + g(\tau)\,\mathrm{d}w(\tau), \qquad \tau \in [0, T], \quad \bar{x}(0) \sim p_0 \in \mathcal{P}(\mathbb{R}^d), \tag{1}$$

where $w = \{w(\tau) : \tau \in [0, T]\}$ is a standard Wiener process (Brownian motion) in $\mathbb{R}^d$, $f : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ and $g : [0, T] \to \mathbb{R}_+$ are measurable coefficients, $\mathcal{P}(\mathbb{R}^d)$ denotes the set of Borel probability measures on $\mathbb{R}^d$,

and p_0 is the unknown target distribution. Here d is usually a very large number in a typical task such as image generation. Let p_τ be the law of $\bar{x}(\tau)$ and write its score function as $S(\tau, x) = \nabla_x \log p_\tau(x)$. Under standard well-posedness assumptions for common choices of (f, g), the SDE (1) maps p_0 along the family $\{p_\tau\}_{\tau\in[0,T]}$ toward a tractable reference distribution at time T.

Backward sampling For sampling, one can work with the reverse-time diffusion or, equivalently, with a deterministic probability flow ordinary differential equation (ODE) that shares the same marginals as the reverse-time SDE. As shown by Tang and Zhao (2025, Theorem 5.1), the ODE and the reverse SDE yield the same family $\{p_\tau\}$. Denoting the backward state by $\tilde{x}(\tau) \coloneqq \bar{x}(T-\tau)$ with initialization $\tilde{x}(0) \sim p_T$ and using a trained score model $\hat{S}(\tau, x)$ in place of the unknown $S(\tau, x)$, the implementable backward probability flow ODE is

$$\frac{\mathrm{d}\tilde{x}(\tau)}{\mathrm{d}\tau} = f\big(T-\tau\big)\tilde{x}(\tau) + \frac{1}{2}\, g\big(T-\tau\big)^2\, \hat{S}\big(T-\tau, \tilde{x}(\tau)\big), \qquad \tau \in [0,T], \quad \tilde{x}(0) \sim p_T. \tag{2}$$

Euler discretization. To generate samples, we numerically integrate (2) on a grid $0 = \tau_0 < \tau_1 < \cdots < \tau_K = T$ with step sizes $h_i = \tau_{i+1} - \tau_i$, and denote $\tilde{x}_i \coloneqq \tilde{x}(\tau_i)$. Using the explicit Euler method, we obtain

$$\tilde{x}_{i+1} = \tilde{x}_i + h_i\Big[f\big(T-\tau_i\big)\tilde{x}_i + \tfrac{1}{2}\, g\big(T-\tau_i\big)^2\, \hat{S}\big(T-\tau_i, \tilde{x}_i\big)\Big], \qquad i = 0, \ldots, K-1, \quad \tilde{x}(0) \sim p_T. \tag{3}$$

A uniform grid $\tau_i = iT/K$ is simple to implement and widely used, but such a single and naïve global step size cannot capture the inevitable variation in numerical characteristics along the trajectory. When K is small, sampling is computationally efficient but discretization error can be significant. When K is large, under a uniform grid the additional function evaluations are spread evenly rather than concentrated on where dominant errors can be most effectively reduced, in which case a nontrivial fraction of the extra computation is poorly utilized. Intuitively, early stages of the reverse process, where samples are close to noise, may tolerate coarser resolution, whereas later stages typically benefit from finer steps. These considerations motivate *adaptive*, data-driven time discretizations that redistribute steps under a fixed total time budget T, allocating computation to where it has the greatest impact on accuracy and sample quality.

3 ART: Time Reparameterization as Control

We now formulate adaptive timestep selection as a continuous-time control problem. The central idea is to introduce a reparameterized sampling clock and to treat the progression of physical diffusion time as a controlled process. By allowing the sampling trajectory to advance at variable speeds at different epochs, this formulation strategically redistributes computational effort under a fixed total time budget. In this section, we first describe the reparameterized dynamics induced by the time change, and subsequently introduce an objective function that formalizes optimal timestep allocation.

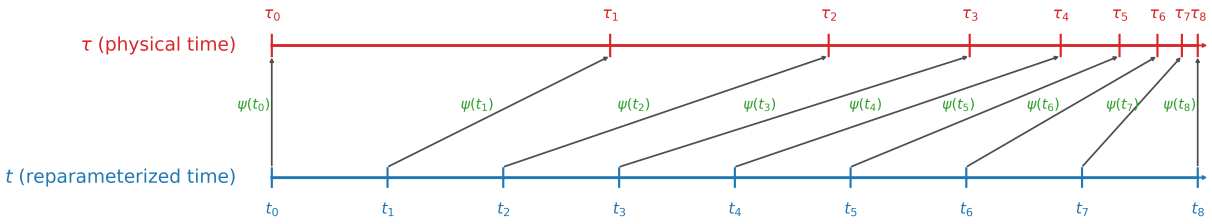


Figure 2: ART as a time change between two clocks. The physical diffusion time τ (top) and the reparameterized time t (bottom) are linked by a continuous map $\tau = \psi(t)$ with $\psi(0) = 0$ and $\psi(T) = T$. A uniform grid in t induces a generally nonuniform grid in τ, with local speed $\theta(t) = \dot{\psi}(t)$ controlling how function evaluations are redistributed along the trajectory.

3.1 Time reparameterization and controlled dynamics

Rather than evolving the reverse process on a fixed physical-time grid, we introduce a reparameterized clock that governs how diffusion time is traversed during sampling. This auxiliary time variable decouples numerical resolution from the original diffusion horizon and enables adaptive redistribution of function evaluations. Specifically, let $\psi : [0, T] \to \mathbb{R}$ be a continuous mapping from the reparameterized time t to the physical diffusion time τ, so that $\tau = \psi(t)$ with $\psi(0) = 0$ and $\psi(T) = T$. Figure 2 depicts this correspondence between the two clocks and the induced nonuniform physical-time grid.

On the reparameterized clock, the reverse-time state is represented as $x(t) \coloneqq \tilde{x}(\psi(t))$ with initialization $x(0) \sim p_T$. The time-change is characterized by the control $\theta(t) \coloneqq \dot{\psi}(t)$, the instantaneous rate at which diffusion time advances relative to the new clock. The boundary conditions on ψ translates into an integral constraint $\int_0^T \theta(t)\,\mathrm{d}t = \psi(T) - \psi(0) = T$, ensuring that the full diffusion horizon is traversed over the sampling interval.

Importantly, we do not restrict ψ to be monotone a priori, consequently allowing $\theta(t)$ to take either sign. This modeling choice is deliberate: it yields a formulation that is closed under optimization and avoids prematurely excluding admissible control trajectories that may arise when learning θ in a data-driven manner. In particular, monotone time reparameterizations, corresponding to $\theta(t) \geqslant 0$ almost everywhere, are naturally recovered as a special case without being hard-coded into the dynamics.

If the reparameterized time is discretized uniformly as $0 = t_0 < t_1 < \cdots < t_K = T$, the resulting physical-time grid is given by $\tau_i = \psi(t_i)$, with step sizes that generally vary across i. From a numerical perspective, the trajectory $x(\cdot)$ evolves on the new clock, while the control $\theta(\cdot)$ determines where progression in physical diffusion time is accelerated or slowed down. This mechanism enables the sampler to allocate resolution adaptively along the reverse trajectory, placing finer discretization where it is most beneficial. We refer to this time-reparameterized sampling framework as *Adaptive Reparameterized Time* (ART).

We now formulate the controlled dynamics under ART. Since the reparameterized state is defined by $x(t) = \tilde{x}(\psi(t))$, the evolution of x on the new clock follows directly from the chain rule. Taking ψ as another

state variable and θ as the control variable, the state dynamics is

$$\begin{cases} \dot{x}(t) = \theta(t)\, F\big(x(t), \psi(t)\big), \qquad x(0) \sim p_T, & \text{(4a)} \\ \dot{\psi}(t) = \theta(t), \qquad \psi(0) = 0,\ \psi(T) = T, & \text{(4b)} \end{cases}$$

where F is the backward probability-flow vector field evaluated at the physical time $T - \psi$, namely

$$F(x, \psi) \coloneqq f\big(T - \psi\big)\, x + \frac{1}{2}\, g\big(T - \psi\big)^2\, \hat{S}\big(T - \psi, x\big), \tag{5}$$

with f and g being the coefficients of the forward diffusion, and $\hat{S}$ a learned score function. The equation (4b) simply records that ψ accumulates at rate θ; so θ can be interpreted as a local time-scaling factor under the new clock. Moreover, we have the time budget constraint

$$\int_0^T \theta(t)\, \mathrm{d}t \ = T, \tag{6}$$

which formalizes that the sampler must allocate a total amount T of physical-time progression across the interval $t \in [0, T]$. This constraint underlines an important feature of ART: any local deceleration of the dynamics, corresponding to smaller values of $\theta(t)$ and hence finer resolution in a given region, must be compensated by acceleration elsewhere, implying that improvements in numerical accuracy are achieved not by increasing the overall computational budget but by redistributing it along the trajectory in a strategic manner.

3.2 Euler error surrogate and control objective

To motivate the formulation of an appropriate objective for selecting the time-warping rate θ on the t-clock, we quantify how the Euler discretization behaves under the controlled dynamics (4a). The basic principle is that the leading-order one-step error is governed by the local stiffness indicator of the probability-flow dynamics, and hence can be used as a proxy for where additional resolution is most valuable. We proceed in the same spirit as the Euler discretization in (3), but now on a fixed, generic step $[t_i, t_{i+1})$ of the t-clock with stepsize $h_i \coloneqq t_{i+1} - t_i$, where the implementation uses a *constant* control value per step, denoted by θ_i.

Let E_i denote the one-step Euler residual:

$$E_i \coloneqq x(t_{i+1}) - \Big(x(t_i) + h_i\, \theta_i\, F\big(x(t_i), \psi(t_i)\big)\Big).$$

A second-order Taylor expansion of the solution map around $(x(t_i), \psi(t_i))$ yields the local error

$$E_i = \frac{h_i^2}{2}\, \theta_i^2\, Q\big(x(t_i), \psi(t_i)\big) + O(h_i^3), \tag{7}$$

where the coefficient Q collects terms arising from differentiating the probability-flow field along the trajectory. More explicitly,

$$\begin{aligned} Q(x,\psi) = &\left[f(T-\psi)I_d + \frac{1}{2}(g(T-\psi))^2\nabla_x\hat{S}(T-\psi,x)\right]\left[f(T-\psi)x + \frac{1}{2}(g(T-\psi))^2\hat{S}(T-\psi,x)\right] \\ &- f'(T-\psi)x - g(T-\psi)g'(T-\psi)\hat{S}(T-\psi,x) - \frac{1}{2}(g(T-\psi))^2\frac{\partial\hat{S}(T-\psi,x)}{\partial\tau}. \end{aligned} \tag{8}$$

Equation (7) shows that the second-order error (in the step size) of the Euler scheme is quadratic in θ_i, modulated by the term $Q(x,\psi)$ evaluated along the trajectory. This function Q captures local geometric and model-induced stiffness of the probability-flow field, and the control θ determines how strongly this stiffness ought to be felt on a given step. As a result, regions where $|Q(x,\psi)|$ is large are precisely where aggressive time progression would amplify discretization error and hence one needs to proceed slowly by taking a small θ, and vice versa.

This motivates interpreting $|Q(x,\psi)|\,\theta(t)^2$ as a local cost density that guides advancing time progression based on $Q(x,\psi)$ adaptively. Together with the time budget constraint (6), we hence introduce the following objective functional

$$J^\theta(s,y,\phi) = \mathbb{E}\Big[\int_s^T \big(-|Q(x(t),\psi(t))|\theta^2(t) - \gamma\theta(t)\big)\,\mathrm{d}t + \gamma T \;\Big|\; x(s)=y,\ \psi(s)=\phi\Big], \tag{9}$$

where $\gamma \in \mathbb{R}$ is the Lagrange multiplier for (6).

We define the optimal value function associated with this control problem as

$$V(s,y,\phi) \coloneqq \sup_{\theta=\theta(\cdot)} J^\theta(s,y,\phi). \tag{10}$$

Consequently, ART reframes timestep allocation as the problem of controlling the time-warping rate θ in the augmented dynamics (4), with the objective (9) capturing how numerical error should be managed along the reverse diffusion trajectory.

4 Randomized Control and Reinforcement Learning Formulation

The formulation in Section 3 provides a principled way to pose timestep allocation as a control problem, but it does not immediately yield a practical solution method in high dimensions. In general, the ART objective (9) admits no closed-form solutions, and the associated Hamilton–Jacobi–Bellman (HJB) equation with a large d is numerically prohibitive due to the curse of dimensionality. To remedy this, we introduce an auxiliary randomized control reformulation in which the time-warping rate is produced by a stochastic policy. The role of randomization here is not for "exploration" due to an unknown environment in the usual reinforcement learning (RL) sense; rather, it is a technical device that enables us to apply the recently developed continuous-time RL theory and algorithms (Wang et al., 2020; Jia and Zhou, 2022a,b). We

refer this reformulation as *Adaptive Reparameterized Time via Reinforcement Learning* (ART-RL), which we develop in the remainder of this section.

4.1 ART-RL: An auxiliary problem with Gaussian policies

We replace the deterministic control θ by a randomized feedback policy that assigns, at each (t, x, ψ), a probability distribution generating time-warping rates. Specifically, we take the following Gaussian policy class whose variance depends on the local numerical sensitivity encoded by Q:[1]

$$\pi^{(\lambda)}(\cdot \mid t, x, \psi) = \mathcal{N}\left(\mu(t, x, \psi),\ \frac{\lambda}{|Q(x, \psi)|}\right), \tag{11}$$

where μ is a (deterministic) measurable function and $\lambda \geqslant 0$ is a scalar parameter. The particular form of the variance ties policy randomization to the geometry of the surrogate error: since $|Q|$ governs the Euler residual through (7), the variance $\lambda/|Q|$ suppresses policy-induced randomness in stiff regions (large $|Q|$) while allowing comparatively more randomness otherwise, while λ controls the overall level of this randomization without changing the mean. For analysis, we assume $|Q(x, \psi)| > 0$ almost surely on compact intervals of (x, ψ). In implementation we replace $|Q|$ by $|Q| \vee \varepsilon := \max(|Q|, \varepsilon)$ for a small $\varepsilon > 0$.

We now present the "exploratory formulation" of ART, following Wang et al. (2020). Denote by $\Pi^{(\lambda)}$ the collection of policies of the form (11). For a fixed policy $\pi^{(\lambda)} \in \Pi^{(\lambda)}$, the corresponding state $(x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))_{t\in[0,T]}$ satisfies the "exploratory" dynamics

$$\begin{cases} \dfrac{\mathrm{d}x^{\pi^{(\lambda)}}(t)}{\mathrm{d}t} = \displaystyle\int_{\mathbb{R}} \theta\, F\big(x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)\big) \\ \qquad\qquad \pi^{(\lambda)}\Big(\theta \mid t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)\Big)\, \mathrm{d}\theta, & x^{\pi^{(\lambda)}}(0) = x_0 \sim P_T, \end{cases} \tag{12a}$$

$$\begin{cases} \dfrac{\mathrm{d}\psi^{\pi^{(\lambda)}}(t)}{\mathrm{d}t} = \displaystyle\int_{\mathbb{R}} \theta\, \pi^{(\lambda)}\Big(\theta \mid t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)\Big)\, \mathrm{d}\theta, & \psi^{\pi^{(\lambda)}}(0) = 0, \quad \psi^{\pi^{(\lambda)}}(T) = T. \end{cases} \tag{12b}$$

Moreover, the performance criterion is

$$\begin{aligned} J^{\pi^{(\lambda)}}(s, y, \phi) = \mathbb{E}\Bigg[\int_s^T \int_{\mathbb{R}} \Big(-|Q\big(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)\big)|\, \theta^2 \ -\ \gamma\, \theta\Big)\, \pi^{(\lambda)}(\theta \mid t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))\, \mathrm{d}\theta\, \mathrm{d}t \\ +\ (\gamma + \lambda)\, T \ \Big|\ x^{\pi^{(\lambda)}}(s) = y,\ \psi^{\pi^{(\lambda)}}(s) = \phi \Bigg]. \end{aligned} \tag{13}$$

Here, the additional term λT in (13) is to compensate a constant bias induced by Gaussian randomization, so that the resulting criterion is comparable to the deterministic one under the same mean control. To see this, fix any mean function $\mu(t, x, \psi)$ and consider the Gaussian policy (11). Taking the policy expectation of the deterministic running cost yields the identity $\int_{\mathbb{R}}\big(-|Q|\,\theta^2 - \gamma\,\theta\big)\,\pi^{(\lambda)}(\theta \mid t, x, \psi)\,d\theta = -|Q|\,\mu^2 - \gamma\,\mu - \lambda$.

[1]The reason for choosing the Gaussian policies will be revealed in the subsequent theoretical analysis.

The associated optimal value function is

$$V^{(\lambda)}(s,y,\phi) \;=\; \max_{\pi^{(\lambda)}\in\Pi^{(\lambda)}} J^{\pi^{(\lambda)}}(s,y,\phi), \qquad (s,y,\phi)\in[0,T]\times\mathbb{R}^d\times\mathbb{R}. \tag{14}$$

4.2 Connecting the original and randomized formulations

We now establish that the solution to the original ART control problem (10) can be recovered from that to the randomized one (14). Indeed, the value function V of (10) satisfies the HJB equation

$$V_t + \sup_{\theta}\Big\{\big(V_x^\top F(x,\psi)+V_\psi-\gamma\big)\theta - |Q(x,\psi)|\,\theta^2\Big\} = 0, \tag{15}$$

together with the terminal condition $V(T,x,\psi)=\gamma T$. Meanwhile, the value function $V^{(\lambda)}$ of (14) satisfies

$$V_t^{(\lambda)} + \sup_{\mu}\Big\{\big(V_x^{(\lambda)\top} F(x,\psi)+V_\psi^{(\lambda)}-\gamma\big)\mu - |Q(x,\psi)|\Big(\mu^2+\frac{\lambda}{|Q(x,\psi)|}\Big)\Big\} = 0, \tag{16}$$

with terminal condition $V^{(\lambda)}(T,x,\psi)=(\gamma+\lambda)T$.

The following theorem discloses a precise relationship between them.

Theorem 1. *If V is a classical solution to the HJB equation* (15)*, then $V^{(\lambda)}$ is a classical solution to the HJB equation* (16) *where*

$$V^{(\lambda)}(t,x,\psi) = V(t,x,\psi)+\lambda t \tag{17}$$

is a classical solution to HJB equation (16)*. Moreover,*

$$\pi^{(\lambda)*}(\cdot|t,x,\psi) = \mathcal{N}\left(\mu^*(t,x,\psi),\frac{\lambda}{|Q(x,\psi)|}\right) \quad \textit{with } \mu^*(t,x,\psi)=\frac{V_x^\top F(x,\psi)+V_\psi-\gamma}{2|Q(x,\psi)|} \tag{18}$$

is the optimal policy for the auxiliary problem (14) *subject to the dynamics* (12)*. Finally, $\mu^*(t,x,\psi)$ is the optimal policy for the original problem* (10) *subject to the dynamics* (4)*.*

Proof. First of all, the "sup" in the two equations (15) and (16) are respectively achieved at

$$\theta^*(t,x,\psi)=\frac{V_x^\top F(x,\psi)+V_\psi-\gamma}{2|Q(x,\psi)|},\;\; \mu^*(t,x,\psi)=\frac{V_x^{(\lambda)\top} F(x,\psi)+V_\psi^{(\lambda)}-\gamma}{2|Q(x,\psi)|}.$$

So if V solves (15) and $V^{(\lambda)}$ is chosen to satisfy (17), then the above two maximizers are identical. Moreover, it is straightforward to check that $V^{(\lambda)}$ solves (16).

Next, we show $V^{(\lambda)}$ and $\pi^{(\lambda)}$ are respectively the optimal value function and optimal policy for the randomized problem (14) via a standard verification approach. Fix a policy $\pi^{(\lambda)}$. Applying Itô's lemma to

$V^{(\lambda)}(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))$, we have

$$
\begin{aligned}
&V^{(\lambda)}(T, x^{\pi^{(\lambda)}}(T), \psi^{\pi^{(\lambda)}}(T)) - V^{(\lambda)}(s, x^{\pi^{(\lambda)}}(s), \psi^{\pi^{(\lambda)}}(s)) \\
&+ \int_s^T \Big(-|Q(x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))|\mu(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))^2 - \lambda - \gamma\mu(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)) \Big) \mathrm{d}t \\
=&\int_s^T \Big(V_t^{(\lambda)} + (V_x^{(\lambda)\top} F(x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)) + V_\psi^{(\lambda)} - \gamma)\mu(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)) \\
&- |Q(x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))|\mu(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))^2 - \lambda \Big) \mathrm{d}t \\
\leqslant& 0,
\end{aligned}
$$

where the last inequality follows from the HJB equation (16). Thus, we have

$$
\begin{aligned}
&V^{(\lambda)}(s, y, \phi) \\
\geqslant&\mathbb{E}\Big[\int_s^T \Big(-|Q(x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))|\mu(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t))^2 - \lambda - \gamma\mu(t, x^{\pi^{(\lambda)}}(t), \psi^{\pi^{(\lambda)}}(t)) \Big) \mathrm{d}t \\
&+ V^{(\lambda)}(T, x^{\pi^{(\lambda)}}(T), \psi^{\pi^{(\lambda)}}(T)) | x^{\pi^{(\lambda)}}(s) = y, \psi^{\pi^{(\lambda)}}(s) = \phi \Big] \\
=&J^{\pi^{(\lambda)}}(s, y, \phi).
\end{aligned}
\tag{19}
$$

When the policy (18) is taken, the above inequality becomes equality because (18) achieves the supremum in the HJB equation (16). This establishes the optimality of the policy (18) along with $V^{(\lambda)}$ being the optimal value function. On the other hand, noticing the previous analysis applies to the case when $\lambda = 0$ and μ^* is independent of λ, we arrive at the final conclusion of theorem.

□

Theorem 1 implies that the ART solution can be recovered by solving the ART-RL problem (14) with Gaussian policies. The latter can indeed be solved using an actor–critic scheme that is *not* directly applicable to the former. We will carry this out in the next section.

5 ART-RL Actor–Critic: Theory and Algorithm

Building on Theorem 1, we now work within ART-RL to learn the ART optimizer. Our approach is premised upon the continuous-time actor–critic framework of Jia and Zhou (2022b), adapted to the ART-RL setting and in particular the Gaussian policies. We start with two theorems on policy evaluation and policy improvement, followed by development of the resulting algorithm.

5.1 Theoretical results: policy evaluation and improvement

In an actor–critic method, the critic estimates the value of a given policy, while the actor updates/improves the policy by moving in a favorable direction guided by this value information. Theorem 2 below formalizes this idea. For any Gaussian policy $\pi^{(\lambda)}$, it first characterizes the associated value function, and then constructs a new Gaussian policy $\tilde{\pi}^{(\lambda)}$, whose mean is obtained by a Hamiltonian-type maximization based on that value function, and shows that this updated policy improves the value for all states.

Theorem 2. *(i) The value function under a Gaussian policy $\pi^{(\lambda)}(\cdot|t,x,\psi) = \mathcal{N}(\mu(t,x,\psi), \frac{\lambda}{|Q(x,\psi)|})$ is given by*

$$J^{\pi^{(\lambda)}}(t,x,\psi) = \bar{v}(t,x,\psi) + \lambda t, \tag{20}$$

where $\bar{v}$ satisfies the linear PDE

$$\bar{v}_t + (\bar{v}_x^\top F(x,\psi) + \bar{v}_\psi - \gamma)\mu(t,x,\psi) - |Q(x,\psi)|\mu(t,x,\psi)^2 = 0, \tag{21}$$

with the terminal condition $\bar{v}(T,x,\psi) = \gamma T$.

(ii) Consider the policy defined as

$$\tilde{\pi}^{(\lambda)}(\cdot|t,x,\psi) = \mathcal{N}\left(\tilde{\mu}(t,x,\psi), \frac{\lambda}{|Q(x,\psi)|}\right), \quad \tilde{\mu}(t,x,\psi) = \frac{\bar{v}_x(t,x,\psi)^\top F(x,\psi) + \bar{v}_\psi(t,x,\psi) - \gamma}{2|Q(x,\psi)|}. \tag{22}$$

Then $\tilde{\pi}^{(\lambda)}$ improves the original policy $\pi^{(\lambda)}$ in the sense that

$$J^{\tilde{\pi}^{(\lambda)}}(t,x,\psi) \geqslant J^{\pi^{(\lambda)}}(t,x,\psi) \quad \textit{for all} \quad (t,x,\psi).$$

Proof. (i) For a given policy $\pi^{(\lambda)}$, applying the Feynman-Kac formula leads to the following PDE for the value function $J^{\pi^{(\lambda)}}$:

$$J_t^{\pi^{(\lambda)}} + (J_x^{\pi^{(\lambda)}\top} F(x,\psi) + J_\psi^{\pi^{(\lambda)}} - \gamma)\mu(t,x,\psi) - |Q(x,\psi)|\mu(t,x,\psi)^2 - \lambda = 0,$$

with $J^{\pi^{(\lambda)}}(T,x,\psi) = (\gamma+\lambda)T$. Since the function $J^{\pi^{(\lambda)}}$ defined in (20) satisfies the above PDE, the result follows from the uniqueness of the solution to linear PDE.

(ii) From Part (i), $J^{\tilde{\pi}^{(\lambda)}}$ can also be expressed as $J^{\tilde{\pi}^{(\lambda)}}(t,x,\psi) = \tilde{v}(t,x,\psi) + \lambda t$, where $\tilde{v}$ satisfies

$$\tilde{v}_t + (\tilde{v}_x^\top F(x,\psi) + \tilde{v}_\psi - \gamma)\tilde{\mu}(t,x,\psi) - |Q(x,\psi)|\tilde{\mu}(t,x,\psi)^2 = 0, \tag{23}$$

with $\tilde{v}(T,x,\psi) = \gamma T$.

Take the left hand side of (21) as a quadratic function in $\mu(t,x,\psi)$, which clearly achieves the maximum

at $\tilde{\mu}(t,x,\psi)$. Hence

$$\begin{aligned}&\bar{v}_t+(\bar{v}_x^\top F(x,\psi)+\bar{v}_\psi-\gamma)\tilde{\mu}(t,x,\psi)-|Q(x,\psi)|\tilde{\mu}(t,x,\psi)^2\\ \geqslant&\bar{v}_t+(\bar{v}_x^\top F(x,\psi)+\bar{v}_\psi-\gamma)\mu(t,x,\psi)-|Q(x,\psi)|\mu(t,x,\psi)^2=0.\end{aligned}\tag{24}$$

It now follows from the comparison principle for PDEs applied to (23) and (24) that $\tilde{v}(t,x,\psi)\geqslant\bar{v}(t,x,\psi)$. Equivalently, $J^{\tilde{\pi}^{(\lambda)}}(t,x,\psi)\geqslant J^{\pi^{(\lambda)}}(t,x,\psi)$ for all (t,x,ψ). □

Theorem 2 is a pure theoretical result that cannot be used directly for computation, because it involves solving PDEs that are intractable in high dimensions. However, it provides the foundation for the next theorem that in turn underpins an implementable, data-driven actor–critic scheme.

Theorem 3. *(i) Let $\pi^{(\lambda)}$ be a Gaussian policy and $\hat{V}$ be a continuous function with $\hat{V}(T,x,\psi)=\gamma T+\lambda T$. Denote by $\theta^{\pi^{(\lambda)}}$ a control sampled from $\pi^{(\lambda)}$ and by $(x^{\theta^{\pi^{(\lambda)}}},\psi^{\theta^{\pi^{(\lambda)}}})$ the state process under $\theta^{\pi^{(\lambda)}}$ with the initial condition $x^{\theta^{\pi^{(\lambda)}}}(s)=y$ and $\psi^{\theta^{\pi^{(\lambda)}}}(s)=\phi$. If for any measurable function ξ and every $(s,y,\phi)\in([0,T]\times\mathbb{R}^d\times\mathbb{R})$,*

$$\begin{aligned}&\mathbb{E}\bigg[\int_s^T\xi(t,x^{\theta^{\pi^{(\lambda)}}}(t),\psi^{\theta^{\pi^{(\lambda)}}}(t))\\&\qquad\Big(\mathrm{d}\hat{V}(t,x^{\theta^{\pi^{(\lambda)}}}(t),\psi^{\theta^{\pi^{(\lambda)}}}(t))-(|Q(x^{\theta^{\pi^{(\lambda)}}}(t),\psi^{\theta^{\pi^{(\lambda)}}}(t))|{\theta^{\pi^{(\lambda)}}}^2(t)+\gamma\theta^{\pi^{(\lambda)}}(t))\mathrm{d}t\Big)\bigg]=0\end{aligned}$$

holds, then $\hat{V}\equiv J^{\pi^{(\lambda)}}$.

(ii) Let $\hat{\pi}(\cdot|t,x,\psi)=\mathcal{N}(\hat{\mu}(t,x,\psi),\frac{\lambda}{|Q(x,\psi)|})$ where $\hat{\mu}$ is a continuous function. Denote by $\theta^{\hat{\pi}^{(\lambda)}}$ a control sampled from $\hat{\pi}^{(\lambda)}$ and by $(x^{\theta^{\hat{\pi}^{(\lambda)}}},\psi^{\theta^{\hat{\pi}^{(\lambda)}}})$ the state process under $\theta^{\hat{\pi}^{(\lambda)}}$ with the initial condition $x^{\theta^{\hat{\pi}^{(\lambda)}}}(s)=y$ and $\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(s)=\phi$. If for any measurable function η and for every $(s,y,\phi)\in([0,T]\times\mathbb{R}^d\times\mathbb{R})$,

$$\begin{aligned}&\mathbb{E}\bigg[\int_s^T\eta(t,x^{\theta^{\hat{\pi}^{(\lambda)}}}(t),\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))\Big(\theta^{\hat{\pi}^{(\lambda)}}(t)-\hat{\mu}(t,x^{\theta^{\hat{\pi}^{(\lambda)}}}(t),\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))\Big)\\&\qquad\Big(\mathrm{d}J^{\pi^{(\lambda)}}(t,x^{\theta^{\hat{\pi}^{(\lambda)}}}(t),\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))-(|Q(x^{\theta^{\hat{\pi}^{(\lambda)}}}(t),\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))|{\theta^{\tilde{\pi}^{(\lambda)}}}^2(t)+\gamma\theta^{\hat{\pi}^{(\lambda)}}(t))\mathrm{d}t\Big)\bigg]=0\end{aligned}$$

holds, then $\hat{\mu}\equiv\tilde{\mu}$ as defined in Theorem 2-(ii).

Proof. (i) The equation presented in the statement is the martingale orthogonality condition for policy evaluation, developed in Jia and Zhou (2022a). By following the same reasoning as in the proof of Proposition 4 therein, we arrive at the following expression:

$$\hat{V}(s,y,\phi)=\mathbb{E}\bigg[\int_s^T\Big(-\gamma\theta^{\pi^{(\lambda)}}(t)-|Q(x^{\theta^{\pi^{(\lambda)}}}(t),\psi^{\theta^{\pi^{(\lambda)}}}(t))|{\theta^{\pi^{(\lambda)}}}^2(t)\Big)\mathrm{d}t+\gamma T+\lambda T\Big|x^{\theta^{\pi^{(\lambda)}}}(s)=y,\psi^{\theta^{\pi^{(\lambda)}}}(s)=\phi\bigg],$$

which is consistent with the definition of the value function $J^{\pi^{(\lambda)}}$.

(ii) To simplify the notation, we denote $\eta(t):=\eta(t,x^{\theta^{\hat{\pi}^{(\lambda)}}}(t),\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))$, $\bar{v}(t)=\bar{v}(t,x^{\theta^{\hat{\pi}^{(\lambda)}}}(t),\psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))$,

$F(t) := F(x^{\theta^{\pi^{(\lambda)}}}(t), \psi^{\theta^{\pi^{(\lambda)}}}(t))$, $Q(t) := Q(x^{\theta^{\pi^{(\lambda)}}}(t), \psi^{\theta^{\pi^{(\lambda)}}}(t))$, $\hat{\mu}(t) := \hat{\mu}(t, x^{\theta^{\hat{\pi}^{(\lambda)}}}(t), \psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))$, and $J(t) := J^{\pi^{(\lambda)}}(t, x^{\theta^{\hat{\pi}^{(\lambda)}}}(t), \psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t)) = \bar{v}(t) + \lambda t$.

Applying Ito's lemma to J, we have

$$\begin{aligned}
0 =& \mathbb{E}\left[\int_s^T \eta(t)(\theta^{\hat{\pi}^{(\lambda)}}(t) - \hat{\mu}(t))\left\{\mathrm{d}J(t) - (\gamma\theta^{\hat{\pi}^{(\lambda)}}(t) + |Q(t)|\theta^{\hat{\pi}^{(\lambda)}2})\mathrm{d}t\right\}\right] \\
=& \mathbb{E}\left[\int_s^T \eta(t)(\theta^{\hat{\pi}^{(\lambda)}}(t) - \hat{\mu}(t))\left\{\bar{v}_t(t) + \lambda + (\bar{v}_x(t)F(t) + \bar{v}_\psi(t) - \gamma)\theta^{\hat{\pi}^{(\lambda)}}(t) - |Q(t)|\theta^{\hat{\pi}^{(\lambda)}2}\right\}\mathrm{d}t\right] \\
=& \mathbb{E}\int_s^T \eta(t)\frac{\lambda}{|Q(t)|}\left\{\bar{v}_x(t)F(t) + \bar{v}_\psi(t) - \gamma - 2|Q(t)|\hat{\mu}(t)\right\}\mathrm{d}t.
\end{aligned}$$

Because this equations holds for any η, the integrand must be zero. Therefore, we obtain the condition

$$\bar{v}_x(t)F(t) + \bar{v}_\psi(t) - \gamma - 2|Q(t)|\hat{\mu}(t) = 0,$$

or

$$\hat{\mu}(t) = \frac{\bar{v}_x(t)F(t) + \bar{v}_\psi(t) - \gamma}{2|Q(t)|}.$$

This expression is identical with $\tilde{\mu}$ defined in (22) of Theorem 2-(ii). □

The term $\theta^{\hat{\pi}^{(\lambda)}}(t) - \hat{\mu}(t, x^{\theta^{\hat{\pi}^{(\lambda)}}}(t), \psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))$ in the equation of Theorem 3-(ii) reveals why invoking stochastic policies is vital for our approach to work: $\theta^{\hat{\pi}^{(\lambda)}}(t)$ is sampled from the Gaussian policy with the mean $\hat{\mu}(t, x^{\theta^{\hat{\pi}^{(\lambda)}}}(t), \psi^{\theta^{\hat{\pi}^{(\lambda)}}}(t))$ so the two terms are generally *different*. In this case the equation provides a genuine direction for policy improvements. If we consider only deterministic policies, then these terms are identical and the equation becomes trivial giving away no information at all about the direction for improvement. This observation highlights the necessity of recasting ART as ART-RL. In the following subsections, we develop an ART-RL algorithm based on the established results.

5.2 Actor–critic parameterization and update rules

Guided by Theorems 2 and 3, we now turn the analytical results into concrete actor–critic update rules. The parametrization of the critic follows the structure (20) in Theorem 2(i), while the parametrization of the actor is chosen within the Gaussian class (11) designed for ART-RL. Specifically, For function approximations of the actor and critic, by Theorems 2 and 3, we parameterize the value function and policy using two separate neural network functions NN^{ϑ_c} and NN^{ϑ_a}:

$$\hat{V}^{\vartheta_c}(t, x, \psi) = NN^{\vartheta_c}(t, x, \psi) + \lambda t, \qquad \hat{\pi}^{\vartheta_a}(\cdot \mid t, x, \psi) = \mathcal{N}\left(NN^{\vartheta_a}(t, x, \psi), \frac{\lambda}{|Q(x, \psi)|}\right). \tag{25}$$

Applying Theorem 3 to the parametrization (25) yields, for suitable test processes ξ and η, the coupled

moment conditions

$$\begin{cases} \mathbb{E}\Big[\int_0^T \xi(t)\big(\mathrm{d}\hat{V}^{\vartheta_c}(t, x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t)) - (|Q(x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t))|\theta^{\hat{\pi}^{\vartheta_a}}{}^2(t) + \gamma\theta^{\hat{\pi}^{\vartheta_a}}(t))\mathrm{d}t\big)\Big] = 0, \\ \mathbb{E}\Big[\int_0^T \eta(t)\big(\theta^{\hat{\pi}^{\vartheta_a}}(t) - NN^{\vartheta_a}(t, x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t))\big)\big(\mathrm{d}\hat{V}^{\vartheta_c}(t, x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t)) \\ \qquad\qquad - (|Q(x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t))|\theta^{\hat{\pi}^{\vartheta_a}}{}^2(t) + \gamma\theta^{\hat{\pi}^{\vartheta_a}}(t))\mathrm{d}t\big)\Big] = 0, \end{cases} \tag{26}$$

where $(x^{\theta^{\hat{\pi}^{\vartheta_a}}}, \psi^{\theta^{\hat{\pi}^{\vartheta_a}}})$ denotes the state process under a control $\theta^{\hat{\pi}^{\vartheta_a}}$ generated from the Gaussian policy $\hat{\pi}^{\vartheta_a}$, which can be simulated and therefore observable as data.

In addition, we take the following test processes:

$$\xi(t) = \frac{\partial}{\partial \vartheta_c} NN^{\vartheta_c}(t, x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t)), \qquad \eta(t) = \frac{\partial}{\partial \vartheta_a} NN^{\vartheta_a}(t, x^{\theta^{\hat{\pi}^{\vartheta_a}}}(t), \psi^{\theta^{\hat{\pi}^{\vartheta_a}}}(t)),$$

which are consistent with the standard choices in the RL actor–critic literature (Sutton and Barto, 1998; Konda and Tsitsiklis, 1999; Jia and Zhou, 2022b; Huang et al., 2025b).

To derive update rules, we interpret the moment conditions (26) as equations in $(\vartheta_c, \vartheta_a)$ and solve them by stochastic approximation. We use subscript n to denote quantities at iteration n; for example, $\vartheta_{c,n}$ denotes the value of ϑ_c at the n-th iteration. Given the n-th observed trajectory (x_n, ψ_n, θ_n) generated/sampled from the current policy $\hat{\pi}^{\vartheta_{a,n}}$ and a learning rate $a_n > 0$, we update the critic and actor parameters by

$$\begin{aligned} \vartheta_{c,n+1} \leftarrow \vartheta_{c,n} + a_n \int_0^T & \frac{\partial NN^{\vartheta_c}}{\partial \vartheta_c}(t, x_n(t), \psi_n(t)) \\ & \cdot \big[\mathrm{d}\hat{V}^{\vartheta_c}(t, x_n(t), \psi_n(t)) - (|Q(x_n(t), \psi_n(t))|\theta_n(t)^2 + \gamma\theta_n(t))\mathrm{d}t\big], \end{aligned} \tag{27a}$$

$$\begin{aligned} \vartheta_{a,n+1} \leftarrow \vartheta_{a,n} + a_n \int_0^T & \frac{\partial NN^{\vartheta_a}}{\partial \vartheta_a}(t, x_n(t), \psi_n(t))\big(\theta_n(t) - NN^{\vartheta_a}(t, x_n(t), \psi_n(t))\big) \\ & \cdot \big[\mathrm{d}\hat{V}^{\vartheta_c}(t, x_n(t), \psi_n(t)) - (|Q(x_n(t), \psi_n(t))|\theta_n(t)^2 + \gamma\theta_n(t))\mathrm{d}t\big]. \end{aligned} \tag{27b}$$

Finally, the Lagrange multiplier γ, enforcing the terminal constraint on ψ, is updated along the same trajectory by

$$\gamma_{n+1} \leftarrow \gamma_n + a_n\big(\psi_n(T) - T\big). \tag{28}$$

Equations (27) and (28) constitute the theoretical update rules of all the learnable parameters in ART-RL.

5.3 Time-discretized ART-RL actor–critic algorithm

To have an implementable algorithm, the final step is to discretize the previous update rules. To this end, we work on a uniform time grid $0 = t_0 < t_1 < \cdots < t_K = T$ with step size $\Delta t = T/K$. For the n-th

iteration, write

$$\hat{V}_k^{\vartheta_{c,n}} := \hat{V}^{\vartheta_{c,n}}(t_k, x_n(t_k), \psi_n(t_k)) = NN^{\vartheta_{c,n}}(t_k, x_n(t_k), \psi_n(t_k)) + \lambda t_k, \qquad k = 0, \dots, K.$$

A simple Riemann approximation of the integrals in (27) leads to the time-discretized critic and actor updates

$$\begin{aligned}\vartheta_{c,n+1} \leftarrow \vartheta_{c,n} + a_n \sum_{k=0}^{K-1} & \frac{\partial NN^{\vartheta_{c,n}}}{\partial \vartheta_c}\big(t_k, x_n(t_k), \psi_n(t_k)\big) \\ & \times \Big[\hat{V}_{k+1}^{\vartheta_{c,n}} - \hat{V}_k^{\vartheta_{c,n}} - \big(|Q(x_n(t_k), \psi_n(t_k))|\, \theta_n(t_k)^2 + \gamma_n\, \theta_n(t_k)\big)\, \Delta t\Big], \end{aligned} \tag{29a}$$

$$\begin{aligned}\vartheta_{a,n+1} \leftarrow \vartheta_{a,n} + a_n \sum_{k=0}^{K-1} & \frac{\partial NN^{\vartheta_{a,n}}}{\partial \vartheta_a}\big(t_k, x_n(t_k), \psi_n(t_k)\big)\, \big(\theta_n(t_k) - NN^{\vartheta_{a,n}}(t_k, x_n(t_k), \psi_n(t_k))\big) \\ & \times \Big[\hat{V}_{k+1}^{\vartheta_{c,n}} - \hat{V}_k^{\vartheta_{c,n}} - \big(|Q(x_n(t_k), \psi_n(t_k))|\, \theta_n(t_k)^2 + \gamma_n\, \theta_n(t_k)\big)\, \Delta t\Big]. \end{aligned} \tag{29b}$$

The update for the Lagrange multiplier is unchanged:

$$\gamma_{n+1} \leftarrow \gamma_n + a_n\big(\psi_n(T) - T\big), \tag{30}$$

where $\psi_n(T) = \psi_n(t_K)$.

We summarize the resulting ART-RL actor-critic scheme in Algorithm 1. The inner loop generates one trajectory under the current Gaussian policy (25), and the outer loop then updates the actor, critic, and Lagrange multiplier via (29) and (30).

Algorithm 1 Time-discretized ART-RL Actor-Critic

for $n = 1$ to N **do**
  Set $k = 0$, $t = t_k = 0$, initialize $(x_n(t_0), \psi_n(t_0))$
  while $t < T$ **do**
    Compute policy mean $m_{n,k} = NN^{\vartheta_{a,n}}(t_k, x_n(t_k), \psi_n(t_k))$
    Sample control according to the Gaussian policy (25) $\theta_n(t_k) \sim \mathcal{N}\Big(m_{n,k}, \frac{\lambda}{|Q(x_n(t_k), \psi_n(t_k))|}\Big)$
    Update $(x_n(t_{k+1}), \psi_n(t_{k+1}))$ by one time step of the ART dynamics (4)
    Increment time: $t_{k+1} = t_k + \Delta t$, $k \leftarrow k + 1$
  end while
  Collect trajectory $\{(t_k, x_n(t_k), \psi_n(t_k), \theta_n(t_k))\}_{k=0}^{K-1}$
  Update critic parameters $\vartheta_{c,n+1}$ via (29a)
  Update actor parameters $\vartheta_{a,n+1}$ via (29b)
  Update multiplier γ_{n+1} via (30)
end for

6 Numerical Experiments

We now numerically evaluate ART-RL across several regimes that differ in dimensionality, numerical solver, model capacity, and experimental protocol. The central practical question is whether ART-RL can improve existing samplers by changing only the timestep grid, while leaving the pretrained model, solver, and other sampling components unchanged. In this section, the EDM- and EDM2-based experiments test this question in modern image-generation pipelines, while the one-dimensional analytical-score experiment isolates discretization effects and the MNIST experiment tests a less optimized score model.

6.1 Experimental setup and baselines

Datasets. We consider both synthetic and real-image settings. For the former, we conduct an experiment where a synthetic target distribution on $\mathbb{R}$ whose score function is known precisely and explicitly, allowing us to isolate the effect of time reparameterization from score-estimation errors. For the latter, which consists of several high-dimensional image generation tasks, we work on the following datasets: 1) CIFAR-10 (Krizhevsky and Hinton, 2009), a dataset of 32×32 natural images from ten classes; 2) AFHQv2, a variant of the AFHQ animal faces dataset (Choi et al., 2020) with 64×64 images of cats, dogs, and wildlife; 3) FFHQ (Karras et al., 2019), a collection of human face images that we downsample to 64×64 as in Karras et al. (2022); and 4) ImageNet (Russakovsky et al., 2015), which we evaluate at both 64×64 and 512×512 resolutions. The ImageNet–64 experiment follows the standard EDM setup of Karras et al. (2022), while the ImageNet–512 experiment adopts the EDM2 setting of Karras et al. (2024). The latter is important because it evaluates ART-RL on a more modern backbone and sampling pipeline, moves from pixel-space diffusion to latent-space diffusion, and tests substantially higher-resolution image generation. To study a small-model regime, we also consider MNIST (LeCun et al., 2002), a dataset of 28×28 grayscale handwritten digits, and train a compact score model directly on this dataset.

Timestep schedules and baselines. We compare four timestep schedules that differ in how they allocate a fixed number of function evaluations along the reverse trajectory.

Uniform is the simplest choice and serves as a reference baseline: it discretizes the physical time interval $\tau \in [0, T]$ using an equally spaced grid.

EDM is the hand-designed schedule by Karras et al. (2022), which is widely adopted in diffusion sampling and is known to perform strongly on standard image benchmarks. The discrete timesteps are calculated by

$$\tau_k = \left(\sigma_{\max}^{1/\rho} + \frac{k}{K}\left(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho}\right)\right)^{\rho}, \qquad k = 0, \ldots, K,$$

with hyperparameters $\sigma_{\min} > 0$, $\sigma_{\max} > \sigma_{\min}$, and $\rho > 0$. Following Karras et al. (2022) we use the recommended default $\rho = 7$. Equivalently, this construction corresponds to using a uniform grid in the transformed coordinate $\sigma^{1/\rho}$ between $\sigma_{\max}^{1/\rho}$ and $\sigma_{\min}^{1/\rho}$, and hence can be interpreted as a fixed, pre-specified

time reparameterization.

DPM-Solver (Lu et al., 2022), henceforth denoted as DPM, provides another strong hand-designed timestep grid that is widely used in fast diffusion sampling. We use its standard uniform log-SNR grid only as a timestep schedule, without changing the numerical integrator. In the variance-exploding (VE) setting, this gives the geometrically spaced noise levels

$$\tau_k = \sigma_{\max}\Big(\frac{\sigma_{\min}}{\sigma_{\max}}\Big)^{k/K}, \qquad k = 0, \dots, K,$$

where $\sigma_{\min} > 0$ and $\sigma_{\max} > \sigma_{\min}$.

ART-RL is our learned schedule, obtained from the ART objective and implemented by Algorithm 1. With ART, a control θ induces a time change ψ, and sampling is performed by placing a uniform grid on the reparameterized clock and mapping it back to physical time through ψ. This construction indeed includes the other three schedules as special cases: the identity map $\psi(t) = t$ recovers Uniform, selecting ψ to match the EDM coordinate $\sigma^{1/\rho}$ (up to a constant rescaling) reproduces EDM, and selecting ψ so that the induced grid is uniform in the DPM-Solver log-SNR coordinate leads to DPM. Importantly, ART-RL learns ψ from data, making the schedule adaptive and allowing timestep allocation to move beyond hand-crafted designs when this improves sampling accuracy under a fixed evaluation budget.

Evaluation metrics. We assess sampling performance using metrics appropriate to the dimensionality and application settings. In the one-dimensional synthetic experiment, sampling accuracy is quantified by the squared Wasserstein distance W_2 between the empirical distribution of generated samples and the (known) target distribution. We report this metric alongside the number of timesteps used by the Euler discretization. For the image-generation experiments, we adopt the standard evaluation protocol and measure sample quality using the Fréchet Inception Distance (FID) as a function of the number of function evaluations (NFE). For the MNIST small-model diagnostic, where Inception features are less natural for handwritten digits, we instead report LeNet-FID using a LeNet feature space. When experiments are conducted within the EDM pipeline (Karras et al., 2022), all components other than the timestep schedule are held fixed; so differences in log FID–NFE curves can be attributed solely to the choice of time discretization. We use FID for image-generation because it is the standard metric in the EDM and EDM2 evaluations and is applicable across the face, animal-face, and ImageNet experiments considered here. Metrics such as Inception Score can be useful as supplemental diagnostics on class-diverse ImageNet-style data, but they are less informative for narrow-domain datasets such as FFHQ or AFHQv2 and are therefore not used here.

Training cost and amortization. ART-RL requires an offline one-off training stage to learn the schedule, whereas Uniform, EDM, and DPM are hand-designed and therefore training-free. This training cost should be interpreted differently from inference or sampling cost: in our CIFAR–10 image experiment, learning the schedule for a given number of timesteps takes about 1–2 hours on a Colab T4 GPU, and this cost is paid

only *once*. After training, we find it justified to *distill* the learned policy into a fixed precomputed time grid; so deployment for sampling is identical to using a hand-designed schedule: the sampler simply reads a list of timesteps, and ART-RL introduces no additional inference-time overhead relative to EDM or DPM. No retraining or architectural modification of the score model is required, nor is any change made to the solver other than the locations at which it evaluates the same reverse dynamics. In this sense, ART-RL is not a competing diffusion backbone or a new sampler implementation; it is a learned schedule that can be dropped into an existing sampler. Moreover, we further experiment on amortizing the cost by *transfer learning*. Specifically, we reuse the *same* distilled schedule trained on CIFAR–10 with a certain number of timesteps across *different* timestep counts, target datasets, and the EDM2 latent-space pipeline without retraining, and find that the results still improve over those of the hand-designed grids. Thus, the relevant practical question is not only whether ART-RL improves a single trained configuration, but whether a learned time parametrization can serve as a reusable schedule across many sampling settings. The generalization experiments reported in Section 6.5 will test this point.

Presentation of quantitative curves. Numerical results will be reported in various tables throughout this section; here we first present a visual overview in Figure 3. To cater for different error scales, the figure collects the error curves using a logarithmic vertical axis and a linear horizontal axis in timestep count or NFE. This log-scale presentation avoids compressing the larger-budget regime, where FID, LeNet-FID, and Wasserstein errors are small but the differences between schedules remain important. Clearly, ART achieves the best results over all the experimented datasets and timestep budgets.

6.2 Experiment with known score function

The first experiment disentangles the effect of timestep allocation from score approximation, where a one–dimensional diffusion model for which all coefficients in the probability–flow ODE are available in closed forms. This setting allows sampling performance to be attributed solely to the choice of time discretization.

The forward diffusion begins at $p_0 = \mathcal{N}(0,1)$, follows the variance exploding (VE) dynamics $\mathrm{d}x(t) = \sqrt{2t}\,\mathrm{d}w(t)$ and terminates at $T = 3$. The marginal law admits the explicit form $x(t) \sim p_t = \mathcal{N}(0, 1+t^2)$, with the terminal distribution $p_T = \mathcal{N}(0, 10)$. The associated score function is then given by $S(t,x) = -x/(1+t^2)$.

Under this specialization, substituting the analytical score into the general definitions (5) and (8) produces explicit expressions for the reparameterized probability–flow field and the Euler error coefficient, namely $F(x,\psi) = -(T-\psi)x/\big(1+(T-\psi)^2\big)$ and $Q(x,\psi) = x/\big(1+(T-\psi)^2\big)^2$.

We next examine the time–warping control learned by ART–RL. We test different number K of timesteps, and here we discuss the case of $K = 100$. After training with $K = 100$, we collect the realized θ sequences from the final 10,000 backward trajectories. To remove incidental fluctuations in the terminal condition due to computational errors, each trajectory is rescaled so that the resulting time change integrates exactly to T. We then compute pointwise summary statistics across trajectories, reporting the empirical mean together

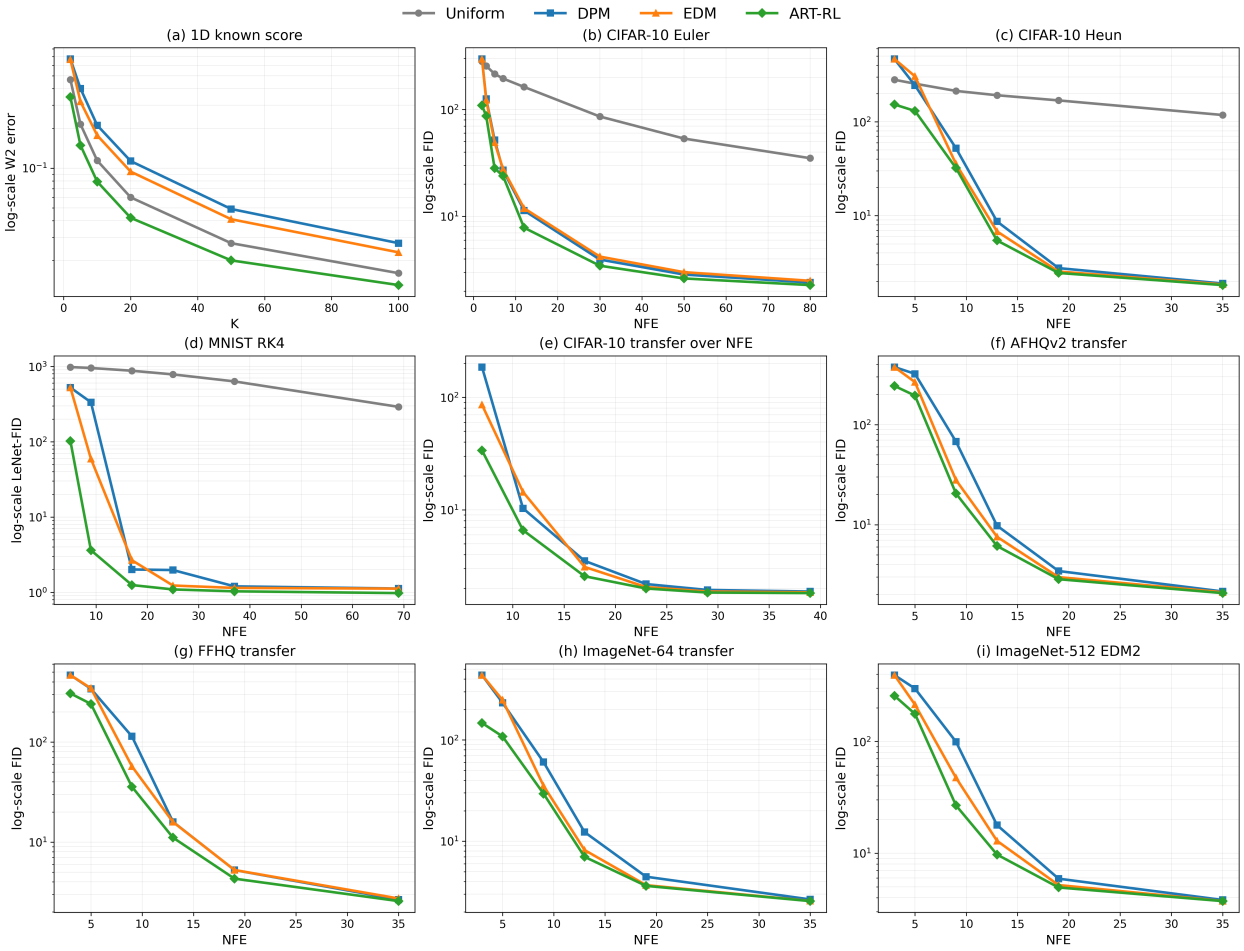


Figure 3: Visual overview of ART-RL across experiments. Each panel uses a logarithmic vertical axis and compares schedules at matched timestep counts or matched NFE. DPM denotes the DPM-Solver timestep grid.

with the interquartile (IQR) range (25–75 percentiles) at each timestep. These aggregated statistics are shown in Figure 4.

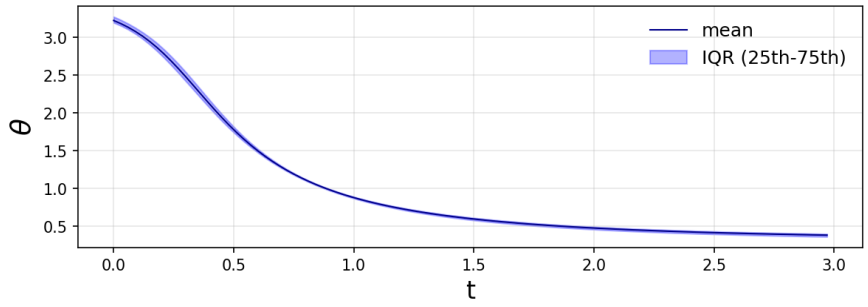


Figure 4: Empirical mean (solid line) and 25–75 percent IQR range (shaded region) of the executed control θ across the last 10,000 training trajectories in the one–dimensional experiment with $K = 100$ timesteps. Each trajectory is normalized so that the resulting terminal time satisfies $\psi(T) = T$.

Figure 4 shows that the mean curve of θ is very smooth and the IQR band is extremely narrow. Moreover, when we plot the 99 percent empirical confidence band (see Appendix A.1, Figure 8), the shaded region is visually indistinguishable from the mean curve. This observation is prevalent with other values of K, which indicates that, in this one–dimensional example, the learned control θ depends only weakly on the state and can be effectively regarded as a deterministic function of time. In other words, the policy has collapsed to an almost *time-only* schedule.

Motivated by this observation, we perform a simple *distillation* step for this one-dimensional example: for each given K we discard the neural network parameterization of the actor (which in general is a feedback policy function of (t, x, ψ)) and replace it with the empirical mean curve of θ as a *fixed* function of t. This distilled approach has two important advantages.

First, it removes entirely the cost of evaluating a neural network to obtain θ at each step. Although the actor network is indeed not large (e.g. much smaller than the score model), repeatedly evaluating it along every sampling trajectory still incurs a nontrivial computational overhead. After distillation, sampling under the ART-RL schedule requires no additional computation beyond that of standard schemes such as Uniform or EDM. In fact, the timestep sequence is precomputed once and then reused.

Second, it eliminates residual mismatch in the terminal time. While the learned actor attempts to enforce $\psi(T) = T$, individual trajectories may slightly overshoot or undershoot T when θ is produced by a neural network at every timestep. This discrepancy is negligible when the number of timesteps K is small but becomes significant as K grows due to the need of a finer time grid. By distilling to a deterministic schedule whose increments are explicitly normalized to sum to T, we guarantee that the induced time grid hits T exactly at the end, thereby improving the numerical fidelity of the discretized probability flow ODE.

We now present the results with different timestep budgets, where the comparison is remarkably consistent. As shown in Figure 3(a) and Table 1, DPM performs the worst among all compared schedules for every value of K, while EDM also underperforms the Uniform grid throughout. By contrast, ART-RL achieves the

best Wasserstein–2 error consistently, with a clear margin over all baselines. These results also show that the hand-designed schedules DPM and EDM are designed *specifically* for image benchmarks and may fail in other domains even in simple toy examples.

Table 1: Wasserstein–2 error versus number of timesteps K in the one–dimensional experiment.

K	2	5	10	20	50	100
Uniform	.468	.215	.114	.060	.027	.016
DPM	.670	.401	.211	.113	.049	.027
EDM	.664	.319	.177	.094	.041	.023
ART-RL	**.345**	**.149**	**.079**	**.042**	**.020**	**.013**

The one–dimensional study in this subsection isolates timestep effects using an analytical score model and shows that ART-RL can learn an effective schedule in a principled way. Starting from the next subsection, we move to image benchmarks and ask whether similar distillation and other generalization techniques still work empirically for more complex tasks.

6.3 CIFAR–10 under EDM pipeline

We next conduct CIFAR–10 experiments within the official EDM pipeline (Karras et al., 2022), keeping the score network, noise conditioning, hyperparameters, and all implementation details fixed across methods under comparison, except the timestep schedule.

First of all, for ART-RL trained on CIFAR–10 with $K = 18$, Figure 5 shows that the empirical 99 percent confidence band remains narrow around a smooth positive mean curve. Similar concentration is observed for other time step counts. This supports distilling the CIFAR–10 trained policies into deterministic time-only grids as in the one-dimensional example. In the experiments below, ART-RL is trained and distilled separately for each K.

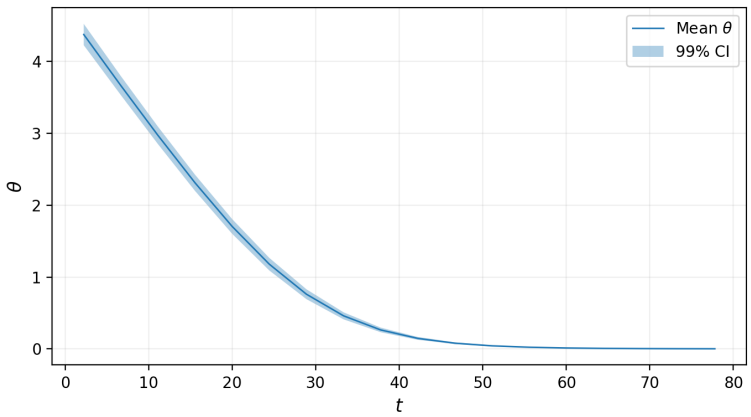


Figure 5: Empirical mean of the executed control θ and its 99 percent confidence interval for ART-RL trained on CIFAR–10 with $K = 18$.

6.3.1 Heun sampling

We first evaluate the distilled ART-RL schedules under the sampling configuration that is most relevant in modern image-generation pipelines. In particular, the default EDM pipeline uses Heun, instead of Euler, and higher-order solvers of this type are widely adopted in practice because they provide improved accuracy per function evaluation. Accordingly, we now evaluate Uniform, DPM, EDM, and ART-RL schedules on CIFAR–10 under the EDM pipeline with the Heun-based sampler, keeping all other components fixed so that differences in FID reflect only the effect of the timestep schedule.

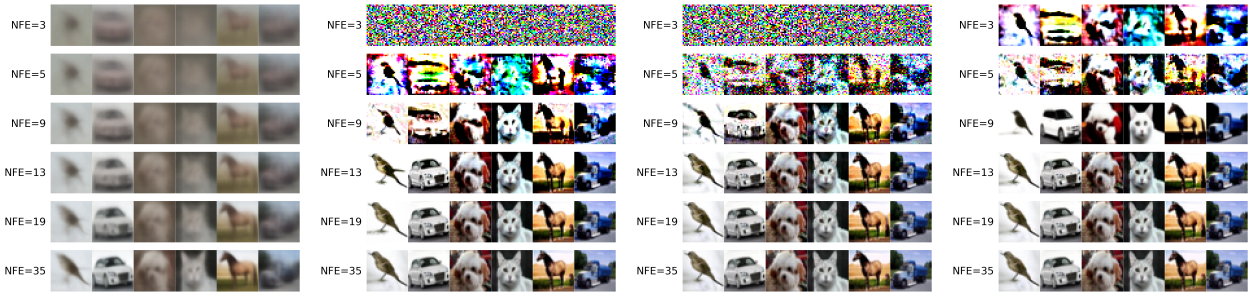
We consider step counts $K \in \{2, 3, 5, 7, 10, 18\}$ and follow the EDM implementation choice of using an Euler step for the final update. With Heun updates, each intermediate step requires two score evaluations; so the overall cost is NFE $= 2K - 1$. In particular, $K = 18$ corresponds to the strongest configuration reported by EDM for CIFAR–10 and is included as a budget-matched comparison.

Figure 3(c) and Table 2 show that ART-RL consistently achieves the best FID across all tested budgets. The improvement is especially pronounced at small to moderate NFEs, where ART-RL outperforms all hand-designed baselines by clear margins. Among the latter, DPM is slightly better than EDM at NFE = 5, while EDM becomes better from NFE = 9 onward. Both, however, remain consistently worse than ART-RL throughout. The outperformance of ART-RL persists even at the largest budget, which is also the strongest configuration reported by Karras et al. (2022): at NFE = 35, ART-RL achieves 1.82 versus 1.85 for EDM. The robustness of the result at NFE = 35 is further supported by three additional matched runs (with 50,000 samples each), which give FIDs $1.82, 1.79, 1.82$ for ART-RL versus $1.85, 1.83, 1.85$ for EDM respectively.

Table 2: FID versus number of function evaluations (NFE) on CIFAR–10 under Heun updates in EDM pipeline.

NFE	3	5	9	13	19	35
Uniform	280.29	254.47	213.13	191.69	168.87	118.02
DPM	465.83	244.50	52.29	8.67	2.76	1.89
EDM	465.83	305.15	35.54	6.79	2.54	1.85
ART-RL	**152.86**	**130.48**	**32.13**	**5.44**	**2.45**	**1.82**

Visual samples provided in Figure 6 show consistent results. Uniform schedules produce visibly blurrier images even at larger budgets, while EDM and ART-RL generate sharp samples once sufficient numbers of evaluations are available. At the smallest budgets (NFE= $3, 5$), ART-RL already produces recognizable images, whereas EDM outputs remain closer to noise.



(a) Uniform (b) DPM (c) EDM (d) ART-RL

Figure 6: CIFAR–10 samples across timesteps for the four schedules (Uniform, DPM, EDM, ART-RL). Each panel shows a 6×6 grid where rows correspond to increasing NFE.

Under Heun sampling, these results show that ART-RL can be deployed as a drop-in replacement for the EDM time grid within a competitive image-sampling pipeline. It improves the sampler substantially in low- and mid-computation regimes while preserving strong performance at larger budgets.

6.3.2 Euler sampling

To complement the Heun results, we also evaluate the distilled ART-RL schedules under Euler updates within the same EDM pipeline, thereby matching the Euler discretization used during ART-RL training. As in the one-dimensional experiment, sampling proceeds with K discrete Euler updates. In this case each update requires one score evaluation; so NFE $= K$. We evaluate representative step budgets $K \in \{2, 3, 5, 7, 12, 30, 50, 80\}$.

Figure 3(b) and Table 3 show that ART-RL consistently achieves the best FID across all tested budgets, with clear margins over all the baselines. Under Euler updates, EDM is slightly better than DPM at smaller budgets, while DPM becomes slightly better at larger budgets. However, both are consistently worse than ART-RL. This shows that the benefit of learning the time grid persists in a high-dimensional image model where the score is learned from data, even when ART-RL is used only by replacing the grid at which the same EDM reverse dynamics are evaluated. Additional 6×6 visual results for the Euler ablation are provided in Appendix A.3.1, Figure 9.

Table 3: FID versus number of function evaluations (NFE) on CIFAR–10 under Euler updates in EDM pipeline.

NFE	2	3	5	7	12	30	50	80
Uniform	280.50	255.02	214.60	194.40	162.14	85.83	53.40	34.99
DPM	295.65	125.67	51.73	27.07	11.35	3.95	2.86	2.41
EDM	295.65	122.56	49.10	27.73	11.91	4.21	3.01	2.50
ART-RL	**109.11**	**86.84**	**28.16**	**23.88**	**7.84**	**3.46**	**2.63**	**2.28**

6.4 MNIST with small score model

We next consider MNIST – a deliberately simple setting in which the score network is lightweight (approximately 4.5 MB), trained from scratch, and much less optimized than the pretrained image-generation models used in the EDM experiments. This experiment is not part of the EDM pipeline and is not meant as a cross-dataset transfer test. Instead, it asks whether schedule learning remains useful when the score model itself is small and less accurate instead of large and optimally pretrained. Such compact score models are common in latency- or memory-limited deployments; so it is important to understand whether timestep adaptation continues to improve sampling quality in this regime.

Again, to isolate the effect of the time grid, all the methods use exactly the same small score network and the same numerical integrator. Moreover, in this experiment we use the RK4 setting (see, e.g., Wu et al. 2024) to test whether the learned schedule remains effective under an ODE solver that has an order higher than Heun. Following the EDM/Heun sampling convention, we use RK4 for the non-terminal updates and retain an Euler step for the final update, which gives NFE $= 4K - 3$. We then compare Uniform, DPM, EDM, and ART-RL schedules under identical training and sampling configurations. In short, the timestep schedule is the only component that differs across the methods under comparison.

Figure 3(d) and Table 4 show that ART-RL achieves the lowest LeNet-FID at every reported evaluation budget. The advantage is consistent across the full NFE range, indicating that the learned timestep allocation remains effective when the score model is compact. Figure 11 in Appendix A.4 further illustrates that ART-RL produces coherent digit samples earlier than the hand-crafted schedules.

Table 4: LeNet-FID versus number of function evaluations (NFE) on MNIST.

NFE	5	9	17	25	37	69
Uniform	981.13	953.74	876.58	783.65	632.26	290.46
DPM	523.60	334.44	2.01	1.97	1.20	1.12
EDM	523.60	59.36	2.66	1.23	1.14	1.12
ART-RL	**102.13**	**3.62**	**1.25**	**1.09**	**1.03**	**0.98**

The MNIST experiment indicates that the advantage of ART-RL extends beyond modern large-model pipelines. Indeed, here the gain over EDM and DPM is even larger, suggesting that learned timestep allocation remains effective even with simple, less optimized score-models.

6.5 Transfer and amortization of ART-RL

We next experiment on the transferability of the ART-RL time schedule beyond the exact configuration in which it is learned. This question is central to the *practical* value of ART-RL: if the learned schedule had to be retrained for every step count, dataset, or sampling pipeline, the offline training cost would be harder to justify. So we study whether the same CIFAR–10 schedule can be amortized across a broad family of settings, turning the learned policy into a fixed plug-in timestep grid. In each transfer experiment, ART-RL

leaves the pretrained model, backbone, solver, and implementation pipeline unchanged and replaces only the hand-designed timestep schedule. Thus, this section tests the strongest practical form of the drop-in claim: a schedule learned once in the CIFAR–10 EDM setting with a given time step budget is inserted directly into other budgets, datasets and even the EDM2 pipeline, and still outperforms.

The study has three parts. The first part focuses on intra-dataset flexibility: starting from the CIFAR–10 schedule learned at $K = 18$ in Section 6.3, we construct schedules for other step counts via interpolation and extrapolation. The second part tests cross-dataset transfer within the pixel-space EDM pipeline: we reuse the same CIFAR–10 schedule on AFHQv2, FFHQ, and ImageNet–64 without any additional training. The third part further tests transfer to the EDM2 Karras et al. (2024) pipeline on ImageNet–512. This setting changes three aspects at once: it uses a more modern EDM2 backbone and sampling pipeline, operates in latent space rather than pixel space, and evaluates high-resolution image generation. Together, these experiments assess whether the learned time parametrization captures structure that persists not only across time budgets and datasets, but also across pipelines, resolutions, and representation spaces.

In this subsection, we restrict attention to DPM, EDM, and ART-RL. The Uniform grid is substantially worse in the corresponding image settings and is therefore omitted.

6.5.1 Transfer across timestep counts on CIFAR–10

We first examine whether the ART-RL schedule learned at $K = 18$ can be reused at other step counts. All the experiments follow the same CIFAR–10 EDM-pipeline configuration as in Section 6.3. For ART-RL, we take the learned $K = 18$ sampling grid and generate new grids for $K' \in \{4, 6, 9, 12, 15, 20\}$ by log-linear resampling of the remaining-time values $T-\psi$. Specifically, let $0 = \psi_0^{(K)} < \cdots < \psi_K^{(K)} = T$ denote the learned K-step grid, where the superscript indicates the step count. For $j = 0, \ldots, K' - 1$, set $r_j = j(K-1)/K'$, $i_j = \lfloor r_j \rfloor$, and $\alpha_j = r_j - i_j$, and define $\psi_j^{(K')} = T - \exp\{(1-\alpha_j)\log(T - \psi_{i_j}^{(K)}) + \alpha_j \log(T - \psi_{i_j+1}^{(K)})\}$, with $\psi_{K'}^{(K')} = T$. This is log-linear in $T - \psi$, and the same rule is used for interpolation to smaller step counts $K' < K$ and extrapolation to larger step counts $K' > K$. For EDM, the timestep sequence at each K is computed directly from its analytic rule.

Figure 3(e) and Table 5 show that the $K = 18$ ART-RL schedule transfers smoothly across different timestep counts. ART-RL still achieves the best FID for all the reported NFEs, outperforming both EDM and DPM after interpolation and extrapolation of the learned schedule. This suggests that the learned time parametrization captures a stable allocation pattern that remains effective under changes in grid resolution.

Table 5: FID versus number of function evaluations (NFE) on CIFAR–10 for interpolated and extrapolated timestep counts.

NFE	7	11	17	23	29	39
DPM	185.63	10.31	3.52	2.19	1.94	1.88
EDM	85.80	14.42	3.11	2.06	1.88	1.85
ART-RL	**33.73**	**6.59**	**2.57**	**2.00**	**1.84**	**1.82**

Additional 6×6 image grids for these interpolated and extrapolated schedules are provided in Appendix A.3.2, Figure 10.

6.5.2 Cross-dataset transfer to AFHQv2, FFHQ, and ImageNet–64

We next test cross-dataset transfer without retraining. For each target dataset and time step budget, we keep the corresponding EDM pipeline unchanged, including the score network, solver configuration, and all hyperparameters, and replace only the timestep grid by the ART-RL schedule learned on CIFAR–10 in Section 6.3. The hand-designed baselines use their corresponding schedules at the same step counts, and the NFE accounting follows the same convention as in the CIFAR–10 experiments.

Figure 3(f)–(h), together with Table 6, show that the learned CIFAR–10 schedule transfers successfully to all the three datasets. ART-RL achieves the lowest FID in every reported setting, suggesting that the learned time parametrization remains effective as a drop-in grid replacement across different image distributions under the same EDM pipeline.

Table 6: FID versus number of function evaluations (NFE) for cross-dataset transfer. The ART-RL schedule is learned on CIFAR–10 and reused without retraining.

Dataset	Method	3	5	9	13	19	35
AFHQv2	DPM	375.76	321.59	67.64	9.77	3.44	2.15
	EDM	375.76	266.02	27.88	7.56	2.99	2.11
	ART-RL	**243.48**	**194.79**	**20.48**	**6.12**	**2.85**	**2.07**
FFHQ	DPM	466.76	340.51	113.87	15.94	5.25	2.66
	EDM	466.76	344.76	57.13	15.87	5.26	2.73
	ART-RL	**305.97**	**240.38**	**35.73**	**11.08**	**4.31**	**2.57**
ImageNet–64	DPM	437.42	233.35	60.48	12.31	4.46	2.66
	EDM	437.42	248.32	35.32	8.18	3.68	2.57
	ART-RL	**147.21**	**108.47**	**29.49**	**7.01**	**3.62**	**2.56**

6.5.3 Transfer to EDM2 on ImageNet–512

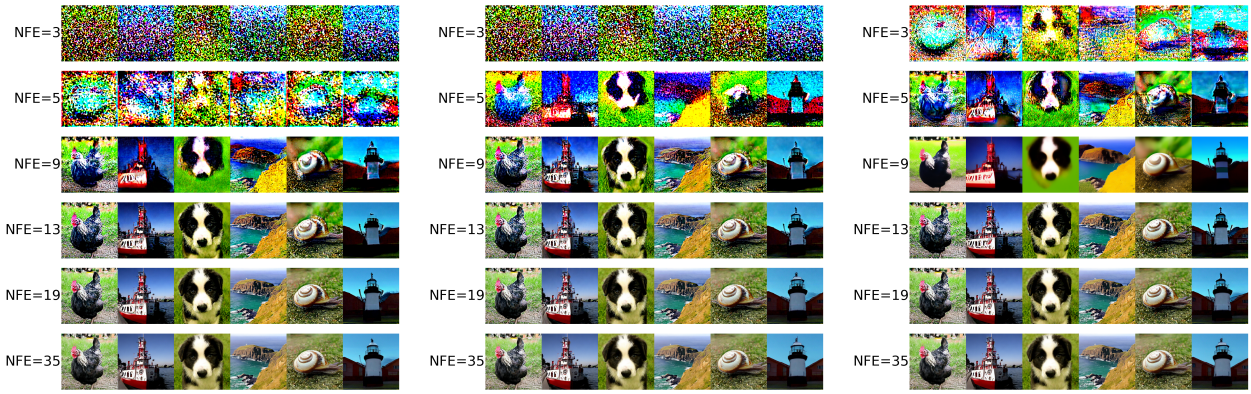
We further evaluate whether the learned ART-RL schedule transfers beyond the pixel-space EDM pipeline. Specifically, we test ImageNet–512 under the EDM2 pipeline, which uses a latent-space diffusion model rather than directly operating in pixel space. We use the extra-small (XS) EDM2 ImageNet–512 model. As before, we keep the score model, solver configuration, and all hyperparameters fixed, and compare only the timestep schedules.

Figure 3(i) and Table 7 show that ART-RL continues to outperform both EDM and DPM under the EDM2 pipeline. The same conclusion holds also under Inception Score: ART-RL attains higher scores than both EDM and DPM at every reported budget; see Appendix A.2, Table 8. This provides the strongest transfer test in our study: the schedule learned from CIFAR–10 under the EDM pipeline remains effective when moved to a different backbone, a different sampling pipeline, a latent representation, and a substantially higher image

resolution. Importantly, this EDM2 experiment still changes only the timestep grid; the pretrained EDM2 model and the rest of the sampling pipeline are left intact. Thus, this ImageNet–512 experiment answers affirmatively whether ART-RL remains useful in a modern high-resolution diffusion setting.

Table 7: FID versus number of function evaluations (NFE) on ImageNet–512 under the EDM2 pipeline using the XS model.

NFE	3	5	9	13	19	35
DPM	392.19	297.26	99.38	17.86	5.92	3.82
EDM	392.19	213.45	47.33	12.91	5.19	3.74
ART-RL	**256.13**	**176.50**	**26.78**	**9.73**	**4.94**	**3.73**



(a) DPM (b) EDM (c) ART-RL

Figure 7: ImageNet–512 samples under the EDM2 pipeline for the three schedules (DPM, EDM, ART-RL). Each panel shows samples at increasing NFEs.

7 Conclusion

This paper introduces and develops ART, a control-theoretic framework, for timestep allocation in score-based diffusion sampling. While ART features a deterministic optimal control problem, solving it via the conventional HJB equations is insurmountable due to the typical (ultra) high dimensions of the state space with most generative AI tasks. We remedy the problem by introducing ART-RL, which is an auxiliary problem with a particular class of Gaussian policies, and establishing a precise relationship between the two problems. The ART-RL problem can be solved algorithmically *à la* the recently developed continuous-time reinforcement learning theory including policy evaluation and policy improvement. It is noteworthy that here ART-RL serves as a technical device to solve ART, rather than a prescription for exploration due to model uncertainty.

Existing time allocation schedules are mostly hand-crafted and *ad hoc* to the underlying tasks. As a result, they do not necessarily work across different types of tasks. For instance, it is shown in this paper that EDM, the state-of-the-art pipeline for image generation, works poorly for a very simple one-

dimensional example with a known score. By contrast, ART is a principled approach premised upon a rigorous theory encompassing general diffusion generative jobs. This is validated by our empirical study: the learned time schedules improve sample quality at matched evaluation budgets across different tasks and numerical solvers, including Euler, Heun, and RK4. They also generalize across timestep counts, datasets, sampling pipelines, and representation spaces. In particular, the experimentally demonstrated transferability of ART has important practical implications. Once a schedule is learned and distilled in image generation, ART has the same inference-time form as EDM or DPM; hence its offline training cost is amortized across many downstream sampling settings.

Remarks on possible future directions are in order. Our analysis is restricted to probability flow ODE sampling, and extending the formulation to SDE samplers may lead to different allocation behaviors and new theoretical questions. The current objective is motivated by an Euler local error surrogate; it would be interesting to investigate alternative criteria, including surrogates aligned with higher-order integrators, to better connect the control principle to practical solvers. Finally, while distillation to time-only schedules is effective in our experiments, it removes state dependence as a consequence, and it is not yet clear if/when richer state-conditioned schedules may provide additional benefits. Overall, ART and ART-RL offer a first step toward a systematic, theory-grounded design of timestep schedules for diffusion-based generative modeling.

Acknowledgments

Yilie Huang acknowledges financial support from the Start-up Fund of The Hong Kong Polytechnic University (Project ID: P0063874). Wenpin Tang is supported by NSF CAREER Award DMS-2538791 and the Tang Family Assistant Professorship. Xun Yu Zhou is supported by the Nie Center for Intelligent Asset Management at Columbia University. Wenpin Tang and Xun Yu Zhou are also part of a Columbia-CityU/HK collaborative project that is supported by the InnoHK Initiative, The Government of the HKSAR, and the AIFT Lab.

References

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *ICLR*, 2024.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *ICML*, pages 4735–4763, 2023a.

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. In *ICLR*, 2023b.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Min Dai, Yuchao Dong, Yanwei Jia, and Xun Yu Zhou. Data-driven merton's strategies via policy randomization. *arXiv preprint arXiv:2312.11797*, 2023.

Xuefeng Gao, Jiale Zha, and Xun Yu Zhou. Reward-directed score-based diffusion models via q-learning. *Journal of Machine Learning Research*, 26(302):1–46, 2025.

Google. State-of-the-art video and image generation with veo 2 and imagen 3. `https://blog.google/technology/google-labs/video-image-generation-update-december-2024/`, 2024. Accessed: 2025-09-17.

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

Daniel Zhengyu Huang, Jiaoyang Huang, and Zhengjiang Lin. Convergence analysis of probability flow ode for score-based generative models. 2025a. arXiv:2404.09730. To appear in IEEE Trans. Inf. Theory.

Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Mean–variance portfolio selection by continuous-time reinforcement learning: Algorithms, regret analysis, and empirical study. 2024. arXiv:2412.16175.

Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Sublinear regret for a class of continuous-time linear-quadratic reinforcement learning problems. *SIAM Journal on Control and Optimization*, 63(5):3452–3474, 2025b.

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(154):1–55, 2022b.

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

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Neurips*, volume 35, pages 26565–26577, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24174–24184, 2024.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, and Stefano Ermon. Mercury: Ultra-fast language models based on diffusion. 2025. arXiv:2506.17298.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. In *Neurips*, volume 35, pages 22870–22882, 2022.

Gen Li and Yuling Yan. Adapting to unknown low-dimensional structures in score-based diffusion models. In *Neurips*, volume 37, pages 126297–126331, 2024.

Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. In *ICLR*, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. 2025. arXiv:2502.09992.

OpenAI. Sora: Creating video from text. `https://openai.com/sora`, 2024. Accessed: 2025-09-17.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. 2022. arXiv:2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, and Oran Gafni. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neurips*, volume 32, page 11918–11930, 2019.

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.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Wenpin Tang and Hanyang Zhao. Score-based diffusion models via stochastic differential equations. *Statistics Surveys*, 19:28–64, 2025.

Wenpin Tang and Xun Yu Zhou. Regret of exploratory policy improvement and q-learning. 2024. arXiv:2411.01302.

Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34, 2020.

Yuchen Wu, Yuxin Chen, and Yuting Wei. Stochastic Runge-Kutta methods: Provable acceleration of diffusion models. 2024. arXiv:2410.04760.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *ICLR*, 2023.

Qinsheng Zhang, Jiaming Song, and Yongxin Chen. Improved order analysis and design of exponential integrator for diffusion models sampling. 2023. arXiv:2308.02157.

Hanyang Zhao, Wenpin Tang, and David D Yao. Policy optimization for continuous reinforcement learning. In *Neurips*, volume 36, 2023.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David Yao, and Wenpin Tang. Scores as Actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. 2024. arXiv:2409.08400.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David Yao, and Wenpin Tang. Score as Action: Fine tuning diffusion generative models by continuous-time reinforcement learning. In *ICML*, 2025.

A Additional Numerical Results

A.1 Results for One–Dimensional Study

Figure 8 shows the empirical mean of the executed control θ together with the 99 percent confidence band computed from the last 10,000 trajectories in the one–dimensional experiment reported in Subsection 6.2. As in the main text, each trajectory is normalized so that the induced terminal time satisfies $\psi(T) = T$. The confidence band is extremely narrow and visually indistinguishable from the mean curve, confirming that in this setting the learned control exhibits negligible variability across trajectories and can be treated as an effectively deterministic function of time.

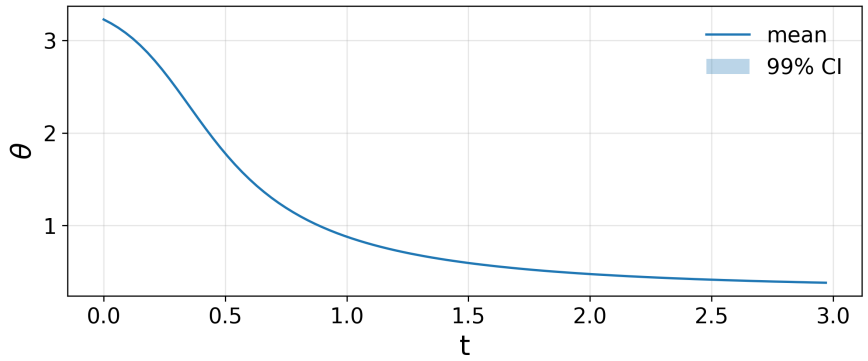


Figure 8: Empirical mean of the executed control θ and its 99 percent confidence interval, based on the last 10,000 trajectories in the one–dimensional experiment.

A.2 Additional ImageNet–512 Inception Score Results

Table 8 reports the Inception Score for the ImageNet–512 EDM2 experiment.

Table 8: Inception Score versus NFE on ImageNet–512 under the EDM2 pipeline using the XS model.

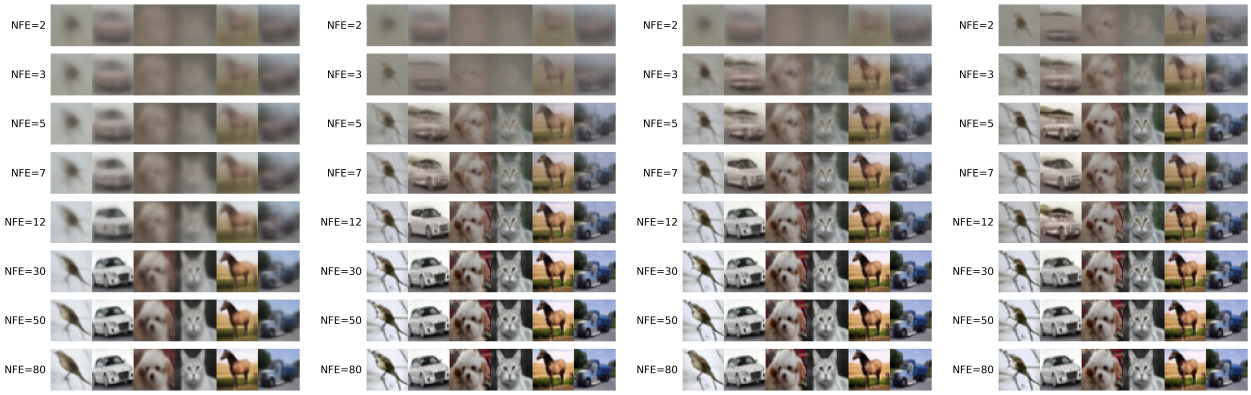
NFE	3	5	9	13	19	35
DPM	1.58	1.82	20.17	106.78	175.94	205.48
EDM	1.58	4.19	53.10	129.75	186.40	206.49
ART-RL	**4.15**	**7.50**	**79.68**	**147.40**	**188.28**	**207.37**

A.3 Additional qualitative results for CIFAR–10

This appendix collects qualitative grids for CIFAR–10 that complement the quantitative results in Section 6.3.2 (Euler ablation) and Section 6.5.1 (interpolation and extrapolation across timestep counts).

A.3.1 Euler ablation grids

Figure 9 shows qualitative results for the Euler ablation in Section 6.3.2, comparing Uniform, EDM, and ART-RL under the same Euler-based sampler. The ART-RL schedule tends to produce recognizable structure earlier at small budgets, consistent with the FID values reported in the main text.



(a) Uniform (b) DPM (c) EDM (d) ART-RL

Figure 9: CIFAR–10 samples across evaluation budgets under Euler updates. Each panel shows a 8×6 grid where rows correspond to increasing NFE.

A.3.2 Interpolation and extrapolation grids

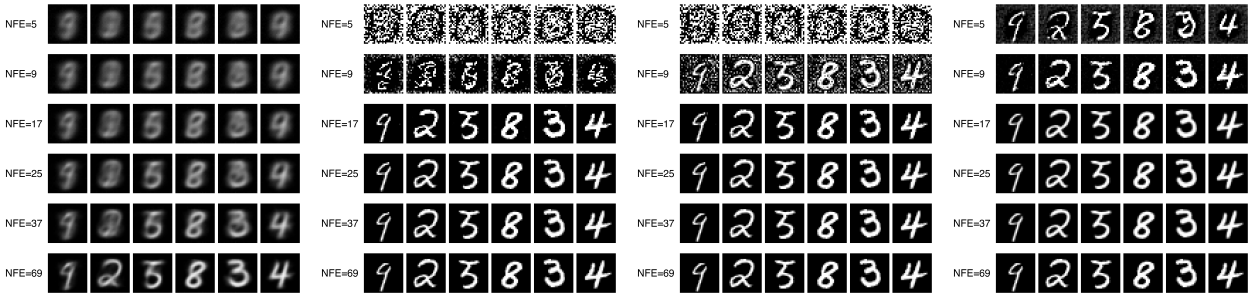
Figure 10 reports qualitative results for the interpolation and extrapolation study in Section 6.5.1. We reuse the schedule learned at $K = 18$ and construct grids for other timestep counts via log-linear interpolation and extrapolation, while DPM and EDM are computed from their analytic rules at each K.

(a) DPM (b) EDM (c) ART-RL

Figure 10: CIFAR–10 samples across evaluation budgets for interpolated and extrapolated timestep counts. Each panel shows a 6×6 grid where rows correspond to increasing NFE.

A.4 Additional qualitative results for MNIST

Figure 11 provides qualitative grids for the MNIST small-model experiment in Section 6.4. The DPM grid is included together with Uniform, EDM, and ART-RL so that the visual comparison matches the quantitative table in the main text.

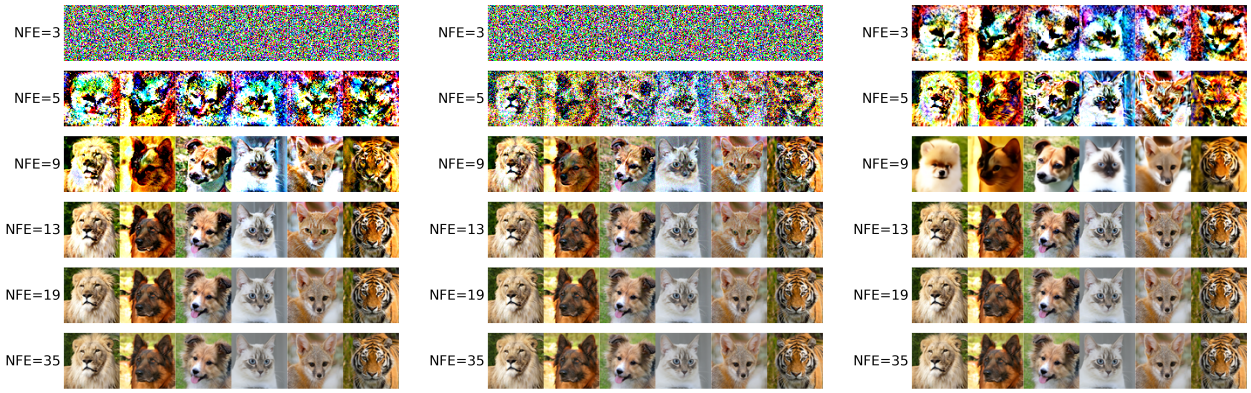


(a) Uniform (b) DPM (c) EDM (d) ART-RL

Figure 11: MNIST samples across timesteps for the four schedules (Uniform, DPM, EDM, ART-RL). Each panel shows a 6×6 grid where rows correspond to increasing NFE.

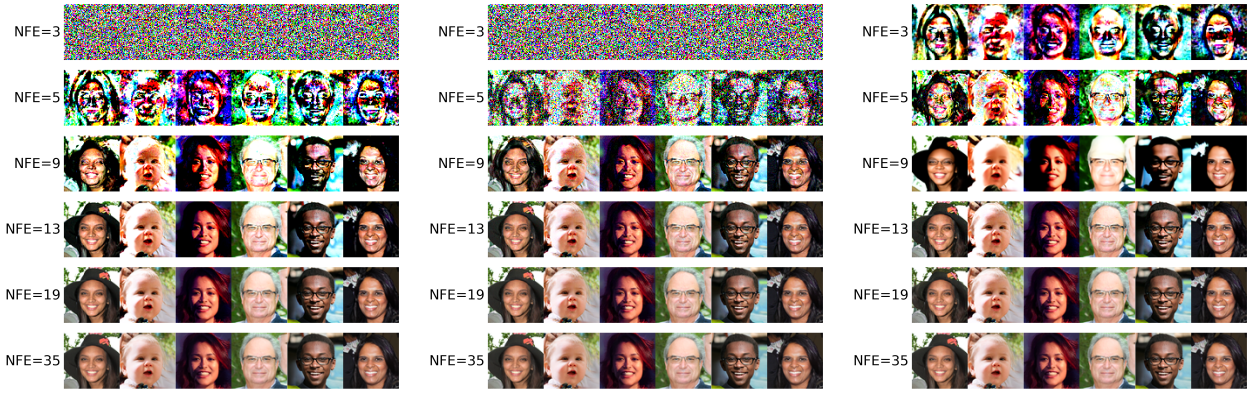
A.5 Additional qualitative transfer results

Figures 12–14 provide qualitative grids for the cross-dataset transfer experiments in Section 6.5.2.



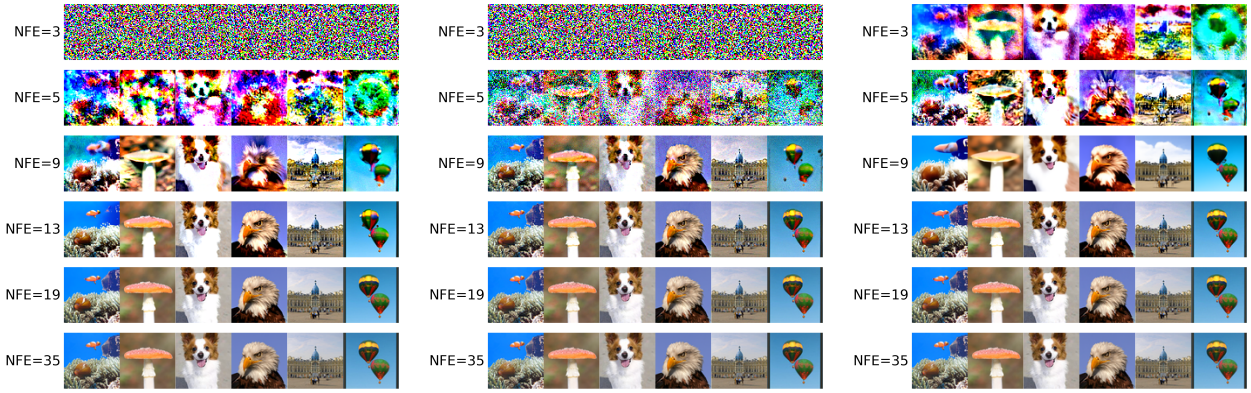
(a) DPM (b) EDM (c) ART-RL

Figure 12: AFHQv2 samples across timesteps for the three schedules (DPM, EDM, ART-RL). Each panel shows a 6×6 grid where rows correspond to increasing NFE.



(a) DPM (b) EDM (c) ART-RL

Figure 13: FFHQ samples across timesteps for the three schedules (DPM, EDM, ART-RL). Each panel shows a 6×6 grid where rows correspond to increasing NFE.



(a) DPM (b) EDM (c) ART-RL

Figure 14: ImageNet–64 samples across timesteps for the three schedules (DPM, EDM, ART-RL). Each panel shows a 6×6 grid where rows correspond to increasing NFE.