# Sample and Computationally Efficient Continuous-Time Reinforcement Learning with General Function Approximation

**Runze Zhao**[*1] **Yue Yu**[*2] **Adams Yiyue Zhu**[3] **Chen Yang**[1] **Dongruo Zhou**[†1]

[1]Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington, Bloomington, Indiana, USA
[2]Department of Statistics, Indiana University Bloomington, Bloomington, Indiana, USA
[3]Department of Electronic and Computer Engineering, University of Maryland, College Park, College Park, Maryland, USA

## Abstract

Continuous-time reinforcement learning (CTRL) provides a principled framework for sequential decision-making in environments where interactions evolve continuously over time. Despite its empirical success, the theoretical understanding of CTRL remains limited, especially in settings with general function approximation. In this work, we propose a model-based CTRL algorithm that achieves both sample and computational efficiency. Our approach leverages optimism-based confidence sets to establish the first sample complexity guarantee for CTRL with general function approximation, showing that a near-optimal policy can be learned with a suboptimality gap of $\tilde{O}(\sqrt{d_{\mathcal{R}} + d_{\mathcal{F}}}N^{-1/2})$ using $N$ measurements, where $d_{\mathcal{R}}$ and $d_{\mathcal{F}}$ denote the distributional Eluder dimensions of the reward and dynamic functions, respectively, capturing the complexity of general function approximation in reinforcement learning. Moreover, we introduce structured policy updates and an alternative measurement strategy that significantly reduce the number of policy updates and rollouts while maintaining competitive sample efficiency. We implemented experiments to backup our proposed algorithms on continuous control tasks and diffusion model fine-tuning, demonstrating comparable performance with significantly fewer policy updates and rollouts. The code is available at `https://github.com/MLIUB/PURE`.

## 1 INTRODUCTION

Continuous-time reinforcement learning (CTRL) is a fundamental problem in learning-based control, with numerous applications in robotics, finance, healthcare, and autonomous systems. Many real-world decision-making problems are more naturally modeled in continuous time rather than discrete time, as they involve continuous interaction with the environment. The goal of CTRL is to find an optimal policy that continuously interacts with and adapts to the environment to maximize long-term rewards. A growing body of work has demonstrated the empirical success of CTRL, leveraging approaches such as model-based continuous-time control [Greydanus et al., 2019, Yildiz et al., 2021, Lutter et al., 2021, Treven et al., 2024a] and fine-tuning in diffusion models [Yoon et al., 2024, Xie et al., 2023]. These studies have shown promising results in real-world tasks, indicating that continuous-time policies can outperform their discrete-time counterparts in various applications.

Despite these empirical advances, the theoretical understanding of CTRL remains limited. A fundamental question in CTRL is *sample efficiency*, which refers to the total number of measurements an agent must take from the environment to learn a near-optimal policy. Existing works have primarily focused on specific settings, such as linear quadratic regulators (LQR) [Cohen et al., 2018, Abeille and Lazaric, 2020, Simchowitz and Foster, 2020] or well-calibrated statistical models [Treven et al., 2024a], where strong structural assumptions facilitate theoretical analysis. However, this stands in contrast to empirical practice, where general function classes—such as neural networks—are widely used. These structured models often fail to capture the complexity of real-world environments, highlighting the need for a more general theoretical framework. Thus, we pose the following question:

*What is the sample complexity for CTRL with general function approximation to find a near-optimal policy?*

Beyond sample efficiency, another crucial aspect is *computational efficiency*, which is characterized by minimizing the number of policy updates and episode rollouts during the online learning phase. Unlike discrete-time RL, where

[*]These authors contributed equally to this work.
[†]Corresponding author.

*Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence* (UAI 2025), PMLR 244:5073–5100.

sample complexity is tightly coupled with the number of episode rollouts, CTRL allows for multiple—even an infinite number of—measurements within a single rollout. This flexibility enables practitioners to employ various measurement strategies, such as equidistant sampling or adaptive strategies, to enhance learning efficiency. While empirical studies have explored multiple measurement strategies [Treven et al., 2024a], their theoretical understanding remains limited, particularly regarding the tradeoff between computational efficiency and sample complexity. This leads to our second fundamental question:

*Can we develop provable new measurement strategies that enhance computational efficiency without significantly sacrificing sample efficiency?*

In this work, we answer the above questions affirmatively. Specifically, we study model-based CTRL in a general function approximation setting, where both the dynamic model and policies are approximated using a general function class. We propose an algorithm, *Policy Update and Rolling-out Efficient CTRL* (PURE), which achieves both sample and computational efficiency. Our main contributions are as follows:

1. We first introduce $\text{PURE}_{\text{base}}$, the foundational version of PURE, which focuses on sample efficiency using the optimism-in-the-face-of-uncertainty principle [Abbasi-Yadkori et al., 2011] and a confidence set construction for the underlying environment. Theoretically, we prove that with $N$ measurements, $\text{PURE}_{\text{base}}$ finds a near-optimal policy with a suboptimality gap of $\tilde{O}(\sqrt{d_{\mathcal{R}} + d_{\mathcal{F}}} N^{-1/2})$, where $d_{\mathcal{R}}$ and $d_{\mathcal{F}}$ denote the Eluder dimensions [Russo and Van Roy, 2013, Jin et al., 2021] of the reward and dynamic functions, respectively. This result provides the first known sample complexity guarantee for CTRL with general function approximation. Notably, unlike prior works such as Treven et al. [2024a], $\text{PURE}_{\text{base}}$ does not rely on an external calibration model, which often requires strong smoothness assumptions that are difficult to satisfy and verify in practice.
2. To improve computational efficiency, we propose $\text{PURE}_{\text{LowSwitch}}$, an extension of $\text{PURE}_{\text{base}}$ that incorporates a tailored policy update strategy, reducing the number of policy updates from $N$ to $O(\log N(d_{\mathcal{R}} + d_{\mathcal{F}}))$. This represents a significant reduction for many function classes. Furthermore, we introduce $\text{PURE}_{\text{LowRollout}}$, designed to minimize the number of policy rollouts. We prove that $\text{PURE}_{\text{LowRollout}}$ reduces the number of rollouts by a factor of $m$, achieving a suboptimality gap of $\tilde{O}(\sqrt{C_{\mathcal{T},m}} N^{-1/2} + m/N)$, where $C_{\mathcal{T},m}$ is the *independency coefficient* that used for quantifying the independency between each measurement. Our results suggest that one can further improve computational complexity for CTRL.
3. We empirically backed up our theoretical findings by implementing PURE in both the traditional continuous-time RL framework [Yildiz et al., 2021] and the diffusion-model fine-tuning framework [Uehara et al., 2024]. Our experimental results demonstrate the practical advantages of PURE, achieving comparable performance with fewer policy updates and rollouts.

## 2 RELATED WORKS

**Continuous-Time Reinforcement Learning** Our algorithms fall into Continuous-Time Reinforcement Learning (CTRL), which has been extensively studied by the control community for decades [Doya, 2000, Vrabie and Lewis, 2009], primarily focusing on planning or simplified models such as the linear quadratic regulator [Shirani Faradonbeh and Shirani Faradonbeh, 2023, Caines and Levanony, 2019, Huang et al., 2024, Basei et al., 2022, Szpruch et al., 2024]. A significant shift occurred with Chen et al. [2018], which introduced CTRL with nonlinear function approximation, enabling continuous-time representations to be learned using neural networks. Building on this foundation, Yildiz et al. [2021] proposed an episodic model-based approach that iteratively fits an ODE model to observed trajectories and solves an optimal control problem via a continuous-time actor-critic method. More recently, Holt et al. [2024] investigated CTRL under a costly observation model, demonstrating that uniform time sampling is not necessarily optimal and that more flexible sampling policies can yield higher returns. While these works primarily focus on empirical studies of CTRL with nonlinear function approximation, theoretical understanding remains limited. In this direction, Treven et al. [2024a] analyzed deterministic CTRL with nonlinear function approximation, introducing the concept of a *measurement selection strategy* (MSS) to adaptively determine when to observe the continuous state for optimal exploration. Extending this line of research, Treven et al. [2024b] studied stochastic CTRL under a cost model, aiming to minimize the number of environment observations. Our work builds upon Treven et al. [2024a] by considering a broader function approximation class and providing theoretical insights into the tradeoff between sample efficiency and computational efficiency, without relying on strong assumptions about the epistemic uncertainty estimator.

**Reinforcement Learning with Low Switching Cost** In many real-world RL applications, frequently updating the policy can be impractical or costly. This has motivated the study of low-switching RL, where the agent deliberately restricts how often its policy changes. Early works focus on the bandit setting, including multi-armed bandits [Auer, 2002, Cesa-Bianchi et al., 2013, Gao et al., 2021] and linear bandits [Abbasi-Yadkori et al., 2011, Ruan et al., 2021], among others. In the RL setting, Bai et al. [2019] and Zhang et al. [2021a] studied low-switching algorithms for tabular Markov Decision Processes (MDPs), while Wang et al.

[2021], He et al. [2023] and Huang et al. [2022] extended the study to linear function approximation. The most relevant works to ours consider low-switching RL with general function approximation. For example, Kong et al. [2021] proposed a low-switching RL approach for episodic MDPs using an online subsampling technique, Zhao et al. [2023] explored low-switching RL through the lens of a generalized Eluder dimension, and Xiong et al. [2023] studied a low-switching RL framework under a general Eluder condition class. Our work differs from these prior studies in two key aspects. From an algorithmic perspective, we develop a CTRL-based approach, which contrasts with existing methods designed for discrete episodic RL. From a theoretical standpoint, we introduce novel analytical tools and notions to handle the continuous-time nature of our dynamics.

## 3 PRELIMINARIES

**Diffusion SDE** In this work, we consider a general nonlinear continuous time dynamical system with a state $x(t) \in \mathcal{X} \subseteq \mathbb{R}^d$ and a control unit $u(t) = \pi(x(t)) \in \mathcal{U} \subseteq \mathbb{R}^m, t \in [T]$. We model the system dynamics using an Itô-form stochastic differential equation (SDE), which is a tuple $\mathcal{S} = \{f^*, g^*, b^*\}$. Given an initial distribution $q \in \mathcal{Q} : \Delta(\mathcal{X})$, let the initial state $x(0) \sim q$, then the flow $x(t), t \in [0, T]$ is evolved following:

$$dx(t) = f^*(x(t), u(t))dt + g^*(x(t), u(t))dw(t), \qquad (1)$$

where $f^* \in \mathcal{F} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^d$ is the drift term and $g^* : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the diffusion term, and $w(t) \in \mathbb{R}^d$ is a standard Wiener process. Our goal is to find a deterministic policy $\pi \in \Pi : \mathcal{X} \to \mathcal{U}$ and an initial distribution $q \in \mathcal{Q} : \Delta(\mathcal{X})$ which maximizes the following quantity:

$$R(\pi, q) := \mathbb{E}\left[\int_{t=0}^{T} b^*(x(t), \pi(x(t)))dt \middle| x(0) \sim q\right],$$

where $b^* \in \mathcal{R} : \mathcal{X} \times \mathcal{U} \to [0, 1]$ denotes the reward function. Note that we only assume $f^*, b^*$ are unknown, and we can access $g^*$ during the algorithm. We only consider time-homogeneous policy. For time-inhomogeneous policy, we augment the state $x'(t) = [x(t), t]$.

*Remark* 3.1. Our formulation of continuous time dynamical system in (1) is general enough to capture many popular applications w.r.t. CTRL. A concrete example is given by diffusion models [Song et al., 2020b], where one could formulate the backward process as follows:

$$dx(t) = f(t, x(t))\, dt + \sigma(t)dw(t), \quad x(0) \sim q, \quad (2)$$

where $f$ is the standard drift formulated by neueral networks, and $\sigma(t)$ is a predefined diffusion term. Notably $f$ can be trained by either score matching [Song et al., 2020a, Ho et al., 2020] or flow matching [Lipman et al., 2022, Shi et al., 2024, Tong et al., 2023, Somnath et al., 2023, Albergo et al., 2023, Liu et al., 2023, 2022a]. Comparing (1) and (2), it is straightforward to see (2) falls into the definition of our (5.4).

For simplicity, we use $X(t, \pi, q)$ to denote the random variable $x(t)$ following policy $\pi$ and the initial distribution $q$. We also denote $z = (x, u)$, and we use $Z(t, \pi, q)$ to denote the random variable $(x(t), \pi(x(t)))$ following policy $\pi$ and the initial distribution $q$.

**Measurement Model** Everytime for a policy $\pi$ and the initial distribution $q$, we can choose a time $t$ to observe the following $(x(t), u(t), y(t), r(t))$, where

$$x(t) = X(t, \pi, q), u(t) = \pi(x(t)),$$
$$y(t) \sim \mathcal{N}(f^*(x, u), \frac{g^*(x, u)^2}{\Delta} \cdot I), r(t) \sim \mathcal{N}(b^*(x, u), 1),$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution and $\Delta > 0$ denotes the measurement time step.

*Remark* 3.2. We assume known diffusion coefficient $g^*(x, u)$ to simplify the theoretical analysis. This assumption is common in related literature — for instance, in diffusion-model-based RL fine-tuning [Uehara et al., 2024, Song et al., 2020a], where $g^* = \sigma$ as discussed in Remark 3.1, and in deterministic dynamic control problems [Yildiz et al., 2021], where $g^* = 0$.

*Remark* 3.3. In practice, only the state $x(t)$, control $u(t)$, and reward $r(t)$ are directly observed, whereas the instantaneous drift $y(t)$ must be approximated. Following the gradient-measurement approach in Treven et al. [2024a] (Definition 1), we assume that both $x(t)$ and $x(t + \Delta)$ can be accessed jointly—effectively doubling the state observation cost, which is often reasonable when dense trajectory data are available. Then, for a sufficiently small time step $\Delta$, we apply the Euler–Maruyama method [Platen and Bruti-Liberati, 2010] to approximate

$$y(t) \approx \frac{x(t + \Delta) - x(t)}{\Delta}.$$

With $\Delta$ small enough, this yields a valid approximation of the true instantaneous drift $y(t)$.

In the following sections, we often omit $t$ and directly use $(x, u, y, r)$ to describe any measurements we will receive during a policy execution.

**Distributional Eluder Dimension** We introduce the notion of the $\ell_p$-distributional Eluder dimension [Jin et al., 2021], which extends the classical Eluder dimension [Russo and Van Roy, 2013] to a distributional setting. Given a domain $\mathcal{A}$, a function class $\mathcal{B} \subseteq \mathcal{A} \to \mathbb{R}$, a distribution class $\mathcal{C} \subseteq \Delta(\mathcal{A})$, and a threshold parameter $\epsilon > 0$, we define the $\ell_p$-distributional Eluder dimension as $\mathrm{DE}_p(\mathcal{A}, \mathcal{B}, \mathcal{C}, \epsilon)$, which is the largest integer $L$ such that there exists a sequence of distributions $p_1, \ldots, p_L \subseteq \mathcal{C}$ satisfying the following condition: there exists a threshold $\epsilon' \geq \epsilon$ such that for all $l \in [L]$, there exists a function $h \in \mathcal{B}$ for which

$$|\mathbb{E}_{p_l} h| > \epsilon \quad \text{and} \quad \sum_{i=1}^{l-1} |\mathbb{E}_{p_i} h|^p \leq \epsilon'^p.$$

Intuitively, the distributional Eluder dimension quantifies the complexity of function class $\mathcal{B}, \mathcal{C}$ by capturing the nonlinearity of the expectation operator $\mathbb{E}_{p_l} h$. In this work, we leverage this measure as the key complexity metric to characterize the nonlinearity in our continuous-time dynamical system.

# 4 PROVABLE CTRL WITH GENERAL FUNCTION APPROXIMATION

In this section, we introduce PURE$_{\text{base}}$, outlined in Algorithm 1. Broadly speaking, PURE$_{\text{base}}$ is a model-based CTRL algorithm that interacts with the environment online, receives feedback, and continuously updates its estimates of the dynamics $f^*$ and reward function $b^*$. During the $n$-th episode, PURE$_{\text{base}}$ maintains confidence sets for $f^*$ and $b^*$, denoted as $\mathcal{F}_n$ and $\mathcal{R}_n$, respectively. Formally, given a dataset $\mathcal{D} = \{(x, u, y, r)\}$, we define the empirical loss functions for the dynamics and reward as

$$L_{\mathcal{D}}(f) = \sum_{(x,u,y,r)\in\mathcal{D}} (f(x,u) - y)^2,$$
$$L_{\mathcal{D}}(b) = \sum_{(x,u,y,r)\in\mathcal{D}} (b(x,u) - r)^2.$$

Let $\mathcal{D}_n$ be the collection of all measurements $(x, u, y, r)$ collected up to episode $(n-1)$. The confidence sets $\mathcal{F}_n$ and $\mathcal{R}_n$ are then constructed as follows:

$$\mathcal{F}_n \leftarrow \left\{f \mid L_{\mathcal{D}_n}(f) \le \min_{f'\in\mathcal{F}} L_{\mathcal{D}_n}(f') + \beta_{\mathcal{F}}\right\}, \quad (3)$$
$$\mathcal{R}_n \leftarrow \left\{b \mid L_{\mathcal{D}_n}(b) \le \min_{b'\in\mathcal{R}} L_{\mathcal{D}_n}(b') + \beta_{\mathcal{R}}\right\}. \quad (4)$$

Following the classical optimism-in-the-face-of-uncertainty principle [Abbasi-Yadkori et al., 2011], PURE$_{\text{base}}$ jointly optimizes its policy, initial distribution, and estimates of the dynamics and reward functions to maximize the accumulated reward $R(\pi, q, f, b)$. At each episode, it uniformly samples a time step $t_n \in [T]$, executes the policy $\pi_n$ under the initial state distribution $q_n$, and receives the measurement $(x_n, u_n, y_n, r_n)$ at time $t_n$. After running for $N$ episodes, PURE$_{\text{base}}$ outputs the target policy and initial distribution by selecting uniformly at random from the existing ones.

*Remark* 4.1. We briefly compare PURE$_{\text{base}}$ with OCoRL [Treven et al., 2024a], which is the most closely related algorithm. A key difference is that OCoRL requires access to an external oracle that quantifies epistemic uncertainty in estimating $f^*$, whereas PURE$_{\text{base}}$ operates without explicitly maintaining such an oracle. Additionally, OCoRL selects $t_n$ deterministically based on complex strategies and assumes additional smoothness conditions on the epistemic uncertainty oracle. In contrast, PURE$_{\text{base}}$ employs a simple and randomized selection of $t_n$, making it more flexible and potentially more practical in real-world applications.

Next we provide theoretical analysis for PURE$_{\text{base}}$. We first make the following assumptions on $f, b, g, \pi$.

**Assumption 4.2.** We have that for any $\pi \in \Pi$, any $f \in \mathcal{F}$, $b \in \mathcal{R}$, any $x, x' \in \mathcal{X}$, $u, u' \in \mathcal{U}$,

- We have the following bounded assumptions: $|f(x,u)| \le 1$, $|b(x,u)| \le 1$ and $|g(x,u)| \le G/\sqrt{d\Delta}$.
- We have the following Lipschitz-continuous assumptions:

$$\|f(x,u) - f(x',u')\|_2 \le L_f(\|x - x'\|_2 + \|u - u'\|_2),$$
$$|b(x,u) - b(x',u')| \le L_b(\|x - x'\|_2 + \|u - u'\|_2),$$
$$|g(x,u) - g(x',u')| \le L_g(\|x - x'\|_2 + \|u - u'\|_2),$$
$$\|\pi(x) - \pi(x')\|_2 \le L_\pi \|x - x'\|_2.$$

Next we define the distributional Eluder dimension of the function class PURE$_{\text{base}}$ used in its algorithm design.

**Definition 4.3.** Let $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$ and $p \in \mathcal{P}$ denote the following distribution class over $\mathcal{Z}$: each distribution $p$ is associated with a policy $\pi \in \Pi$ and an initial distribution $q \in \mathcal{Q}$, such that

$$z = (x,u) \sim p : \begin{cases} x = X(t, \pi, q),\ t \sim \mathrm{Unif}[0, T], \\ u = \pi(x). \end{cases}$$

Furthermore, denote function class $\bar{\mathcal{F}}, \bar{\mathcal{R}}$ as

$$\bar{\mathcal{F}} = \{\|f - f^*\|_2^2 : f \in \mathcal{F}\}, \bar{\mathcal{R}} = \{(b - b^*)^2 : b \in \mathcal{R}\}.$$

Then we set

$$d_{\mathcal{F},\epsilon} = \mathrm{DE}_1(\mathcal{Z}, \bar{\mathcal{F}}, \mathcal{P}, \epsilon),\ d_{\mathcal{R},\epsilon} = \mathrm{DE}_1(\mathcal{Z}, \bar{\mathcal{R}}, \mathcal{P}, \epsilon).$$

Intuitively, $d_{\mathcal{F},\epsilon}$ and $d_{\mathcal{R},\epsilon}$ capture the nonlinearity of loss functions $\|f - f^*\|_2^2$ and $(b - b^*)^2$ with respect to the dynamical system induced by function class $\mathcal{F}$ and policy class $\Pi$. Next proposition shows that several common models, including linear dynamical systems, enjoy small distributional Eluder dimensions. The proof is deferred to Appendix A.1.

**Proposition 4.4.** *Let $\mathcal{F} = \{f_\theta(z) = \langle \Theta, \phi(z)\rangle : \|\Theta\|_F \le R\}$ and $\mathcal{R} = \{b_\theta(z) = \langle \theta, \phi(z)\rangle : \|\theta\| \le R\}$, where $\Theta \in \mathbb{R}^{d\times d}$ and $\theta \in \mathbb{R}^d$. These represent classes of linear functions on $\mathcal{Z}$ with a feature mapping $\phi$. Assume that $\|\phi(z)\| \le L$ for all $z \in \mathcal{Z}$. Then, for any family $\mathcal{P}$ of distributions on $\mathcal{Z}$ and for any $\epsilon > 0$, we have $d_{\mathcal{F},\epsilon} = O(d^2 \log(1 + R^2L^2/\epsilon^2))$, $d_{\mathcal{R},\epsilon} = O(d^2 \log(1 + R^2L^2/\epsilon^2))$.*

Next we show our main algorithm which characterizes the sample complexity of PURE$_{\text{base}}$.

**Theorem 4.5.** *Set confidence radius $\beta_{\mathcal{F}}, \beta_{\mathcal{R}}$ as*

$$\beta_{\mathcal{F}} = O(G^2 \log(N \cdot \log N |\mathcal{N}(\mathcal{F}, N^{-2})|/\delta)),$$
$$\beta_{\mathcal{R}} = O(\log(N \cdot \log N |\mathcal{N}(\mathcal{R}, N^{-2})|/\delta)),$$

*then with probability at least $1 - \delta$, Algorithm 1 satisfies:*

**Algorithm 1** $\text{PURE}_{\text{base}}$

**Require:** Total measurement number $N$, policy class $\Pi$, initial distribution class $\mathcal{Q}$, drift class $\mathcal{F}$, diffusion term $g^*$, reward class $\mathcal{R}$, confidence radius $\beta_{\mathcal{F}}$, $\beta_{\mathcal{R}}$, episode length $T$, initial measurement set $\mathcal{D}_1 = \emptyset$

1: Initialize confidence sets $\mathcal{F}_1 = \mathcal{F}, \mathcal{R}_1 = \mathcal{R}$.
2: **for** episode $n = 1, \ldots, N$ **do**
3: Set $\pi_n, q_n, f_n, b_n \leftarrow \arg\max_{\pi\in\Pi, q\in\mathcal{Q}, f\in\mathcal{F}_n, b\in\mathcal{R}_n} R(\pi, q, f, b)$.
4: Uniformly sample $t_n \in \text{Unif}[0, T]$. Execute $q_n, \pi_n$ and receives measurement $(x_n, u_n, y_n, r_n)$ at time $t_n$. Update $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup (x_n, u_n, y_n, r_n)$.
5: Update $\mathcal{F}_{n+1}$ following (3), update $\mathcal{R}_{n+1}$ following (4).
6: **end for**

**Ensure:** Randomly pick up $n \in [N]$ uniformly and outputs $(\hat{\pi}, \hat{q})$ as $(\pi_n, q_n)$.

- *For all* $n \in [N]$, $f^* \in \mathcal{F}_n, b^* \in \mathcal{R}_n$.
- *The suboptimality gap of* $(\hat{\pi}, \hat{q})$ *is bounded by*

$$O\Big(\frac{T\sqrt{d_{\mathcal{R},N^{-1}}\beta_{\mathcal{R}}} + LT^{3/2}\sqrt{\exp(KT)}\sqrt{d_{\mathcal{F},N^{-1}}\beta_{\mathcal{F}}}}{\sqrt{N/\log N}}\Big), \tag{5}$$

*where* $K = 1 + (1 + L_\pi)^2 \cdot L_g^2 + 2(1 + L_\pi)^2 \cdot L_f^2, L = L_b(1 + L_\pi)$.

From Theorem 4.5, we know that to find an $\epsilon$-optimal policy $\hat{\pi}$, it is sufficient to set the total measurement number

$$N = \tilde{O}\Big(\epsilon^{-2} \cdot \big(T^2 d_{\mathcal{R},\epsilon^{-2}} \log(|\mathcal{N}(\mathcal{R}, \epsilon^{-4})|/\delta) + L^2T^3 \exp(KT) d_{\mathcal{F},\epsilon^{-2}} G^2 \log(|\mathcal{N}(\mathcal{F}, \epsilon^{-4})|/\delta)\big)\Big),$$

which gives us an $\tilde{O}(\epsilon^{-2})$ sample complexity if we treat the Eluder dimensions and covering numbers as constants.

*Remark* 4.6. Note that our sample complexity has an exponential dependence on the time horizon $T$, which seems larger than the polynomial dependence of the planning horizon in discrete-time RLs [Jin et al., 2021]. However, we would like to highlight that CTRL and discrete-time RL are two different types of algorithms for different problem settings, thus they can not be compared directly. Meanwhile, our result also aligns with several recent works about CTRL [Treven et al., 2024a], which also established exponential dependence on $T$.

*Remark* 4.7. We assume a perfect approximation of $R$ to simplify the theoretical analysis. However, it is not difficult to extend our results to account for a $\ell$-approximation. Suppose that in each episode $n$, we can only access an estimate $\hat{R}_n$ such that $|\hat{R}_n - R| \le \ell$ for some $\ell > 0$. Then, it is not difficult to show that the suboptimality gap will be bounded by (5) with an additional $\ell$ factor. This ensures that the approximation error does not significantly impact the overall performance of the algorithm.

### 4.1 PROOF SKETCH

Below is the proof sketch of Theorem 4.5, with full proof deferred to Appendix B. We mainly demonstrate how to bound the regret $\sum_{n=1}^N R_n$, where

$$R_n = R(f^*, b^*, \pi^*, q^*) - R(f^*, b^*, \pi_n, q_n).$$

1. **Trajectory deviation.** Applying Itô's lemma, Fubini's theorem, Grönwall's inequality and standard analytic arguments, the mean–square gap between the true trajectory $x_n(t)$ and the optimistic trajectory $\hat{x}_n(t)$ is

$$\mathbb{E}\big\|\hat{x}_n(t) - x_n(t)\big\|^2 \le 2e^{Kt}\int_0^t \mathbb{E}\big\|f^*(x_n(s), \pi_n(x_n(s))) - f_n(x_n(s), \pi_n(x_n(s)))\big\|^2 ds.$$

2. **High-probability confidence sets.** Standard covering argument yields empirical–risk inequalities for any $b \in \mathcal{R}$ and $f \in \mathcal{F}$. Choosing confidence radii $\beta_{\mathcal{R}}, \beta_{\mathcal{F}} = \tilde{O}(\log(|\cdot|/\delta))$ guarantees $b^* \in \mathcal{R}_n$ and $f^* \in \mathcal{F}_n$ for all $n$ with probability $1 - \delta$.

3. **Per-episode regret decomposition.** By optimism,

$$R_n \le R(f_n, b_n, \pi_n, q_n) - R(f^*, b^*, \pi_n, q_n).$$

Using Lipschitz continuity and Cauchy–Schwarz inequality, one shows

$$R_n \le L\,\mathbb{E}\int_0^T \|\hat{x}_n(t) - x_n(t)\|dt + \mathbb{E}\int_0^T (b_n - b^*)(x_n(t), \pi_n(x_n(t)))\,dt,$$

which, together with the trajectory bound, gives

$$R_n \;\le\; LT\sqrt{2Te^{KT}A_n} \;+\; T\sqrt{B_n},$$

where $A_n = \mathbb{E}\|f_n - f^*\|^2$, $B_n = \mathbb{E}|b_n - b^*|^2$.

4. **Chaining via Eluder dimension.** Applying Theorem 5.3 from Wang et al. [2023] to the sequences $\{(b_n - b^*)^2\}$ and $\{\|f_n - f^*\|^2\}$ converts the confidence radii $\beta_{\mathcal{R}}, \beta_{\mathcal{F}}$ into $\sum_n B_n = O(d_{\mathcal{R}}\beta_{\mathcal{R}} \log N)$ and $\sum_n A_n = O(d_{\mathcal{F}}\beta_{\mathcal{F}} \log N)$.

5. **Cumulative regret.** Summing the regret decomposition from step 3 over $n = 1, \dots, N$ and applying the Cauchy–Schwarz inequality yield, with probability at least $1 - 2\delta \log N$,

$$\sum_{n=1}^{N} R_n = O\Big(T\sqrt{N\, d_{\mathcal{R}}(\log N + \log|\mathcal{R}_\epsilon|)} + LT\sqrt{TN\, e^{KT}\, d_{\mathcal{F}}(\log N + \log|\mathcal{F}_\epsilon|)}\Big).$$

Replacing $\delta$ by $\delta/(2\log N)$ in the confidence parameter leaves the asymptotic rate unchanged, and thus the theorem follows.

# 5 IMPROVED COMPUTATIONAL EFFICIENCY FOR PURE

$\text{PURE}_{\text{base}}$ suggests that to find an $\epsilon$-optimal policy, $\tilde{O}(\epsilon^{-2})$ measurements are required. This dependency aligns with the standard statistical error rate established in prior works. However, a key limitation of $\text{PURE}_{\text{base}}$ is that it updates its policy and initial distribution in every episode, which can be computationally expensive if such updates are costly. Additionally, $\text{PURE}_{\text{base}}$ collects only a single uniformly random measurement per episode. While this ensures sample efficiency, it also results in wasted rollouts, as the policy is executed at all times but only measured at one. In contrast, discrete-time RL evaluates the policy at every time step, making it more computationally efficient. To address these two challenges, we propose two improved versions of $\text{PURE}_{\text{base}}$, each designed to tackle a specific limitation.

## 5.1 POLICY UPDATE EFFICIENT PURE

We first introduce $\text{PURE}_{\text{LowSwitch}}$ in Algorithm 2, designed to reduce the frequency of policy and initial distribution updates. In essence, $\text{PURE}_{\text{LowSwitch}}$ follows the same setup as $\text{PURE}_{\text{base}}$ while actively monitoring how well the estimated model fits the collected measurements. Specifically, it updates the dynamic model $f_n$ and reward model $b_n$ only when either fails to fit the current dataset $\mathcal{D}_{n+1}$—that is, when the empirical loss $L_{\mathcal{D}_{n+1}}(f_n)$ exceeds a predefined threshold. When such a discrepancy is detected, $\text{PURE}_{\text{LowSwitch}}$ updates the corresponding model and adjusts the policy and initial distribution accordingly.

Next, we present the theoretical guarantees for $\text{PURE}_{\text{LowSwitch}}$.

**Theorem 5.1.** *Setting the confidence radii $\beta_{\mathcal{F}}, \beta_{\mathcal{R}}$ as in Theorem 4.5, with probability at least $1 - \delta$, Algorithm 2 satisfies:*

- *For all $n \in [N]$, $f^* \in \mathcal{F}_n$ and $b^* \in \mathcal{R}_n$.*
- *The suboptimality gap of $(\hat{\pi}, \hat{q})$ matches that in* (5)*.*
- *The total number of episodes where $\pi_n$ and $q_n$ are updated is at most $\log N \cdot O(d_{\mathcal{F},N^{-1}} + d_{\mathcal{R},N^{-1}})$.*

The proof is deferred to Appendix C. The result above implies that $\text{PURE}_{\text{LowSwitch}}$ significantly reduces the number of policy and initial distribution updates from $N$ to $\log N$, without degrading the final policy performance.

*Remark* 5.2. A similar strategy has been explored in prior works on discrete-time RL with general function approximation [Xiong et al., 2023, Zhao et al., 2023]. However, in discrete-time RL, these methods monitor *every* discrete time step $t$ to detect discrepancies in the estimated dynamics. In contrast, such an approach is infeasible in CTRL, as continuous-time monitoring is not possible. This fundamental difference makes our analysis of $\text{PURE}_{\text{LowSwitch}}$ significantly more challenging.

## 5.2 ROLLOUT EFFICIENT PURE

Next, we study how to reduce the number of rollouts required by $\text{PURE}_{\text{base}}$. To achieve this, we propose $\text{PURE}_{\text{LowRollout}}$, outlined in Algorithm 3, which performs multiple measurements within a single episode. Specifically, in the $n$-th episode, measurements are taken at times $t_{n,1}, \dots, t_{n,m}$, where $m$ is the *measurement frequency*. For simplicity, we analyze a fixed measurement strategy, assuming that for any $n \in [N/m]$, the measurement times $t_{n,1}, \dots, t_{n,m}$ are sampled from a predefined distribution $\mathcal{T}$. Here, $\mathcal{T}$ is allowed to be any joint distribution over $\text{Unif}[0,T]^m$. In each episode, the dataset $\mathcal{D}_n$ is updated in batches, incorporating $m$ measurements $\{(x_{n,i}, u_{n,i}, y_{n,i}, r_{n,i})\}_{i=1}^m$ collected during the episode. Other than this batched measurement update, $\text{PURE}_{\text{LowRollout}}$ follows the same procedure as $\text{PURE}_{\text{base}}$.

By introducing the measurement frequency $m$ and the sampler $\mathcal{T}$, $\text{PURE}_{\text{LowRollout}}$ reduces the number of policy rollouts from $N$ to $N/m$. To maintain a fair comparison and ensure consistency in sample complexity, we also set the total number of episodes to be $N/m$, ensuring that $\text{PURE}_{\text{LowRollout}}$ and $\text{PURE}_{\text{base}}$ use the same total number of measurements, differing only in the number of rollouts.

*Remark* 5.3. Our in-episode sampling strategy is similar to the Measurement-Selection-Strategy (MSS) introduced in Treven et al. [2024a]. However, a key distinction is that we do not impose determinism or any specific structure on the sampler $\mathcal{T}$, allowing for a more general and flexible measurement selection process.

Next, we analyze how the introduced measurement frequency affects the output policy and initial distribution. To quantify this effect, we define the *independency coefficient* of $\mathcal{T}$, which measures how well our sampler approximates i.i.d. samples.

**Definition 5.4.** Given a policy $\pi$, an initial distribution $q$, and a sampling strategy $\mathcal{T}$, let $\hat{Z}$ be the random variable defined as $\hat{Z} = Z(t, \pi, q)$, where $t \sim \text{Unif}[0,T]$. Let $\bar{Z}_1, \dots, \bar{Z}_m$ be random variables corresponding to measurement times $t_1, \dots, t_m \sim \mathcal{T}$, where $\bar{Z}(t)$ denotes a trajectory sampled according to $\pi, q$, and $\bar{Z}_i = \bar{Z}(t_i)$. We

**Algorithm 2** $\text{PURE}_{\text{LowSwitch}}$

**Require:** Total measurement number $N$, policy class $\Pi$, initial distribution class $\mathcal{Q}$, drift class $\mathcal{F}$, diffusion term $g^*$, reward class $\mathcal{R}$, confidence radius $\beta_{\mathcal{F}}$, $\beta_{\mathcal{R}}$, episode length $T$, initial measurement set $\mathcal{D}_1 = \emptyset$.

1: Initialize confidence sets $\mathcal{F}_1 = \mathcal{F}, \mathcal{R}_1 = \mathcal{R}$.
2: **for** episode $n = 1, \ldots, N$ **do**
3: Set $\pi_n, q_n, f_n, b_n \leftarrow \arg\max_{\pi\in\Pi, q\in\mathcal{Q}, f\in\mathcal{F}_n, b\in\mathcal{R}_n} R(\pi, q, f, b)$.
4: Uniformly sample $t_n \in \text{Unif}[0, T]$. Execute $q_n, \pi_n$ and receives measurement $(x_n, u_n, y_n, r_n)$ at time $t_n$. Update $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup (x_n, u_n, y_n, r_n)$.
5: Set $\mathcal{F}_{n+1} \leftarrow \mathcal{F}_n$, $\mathcal{R}_{n+1} \leftarrow \mathcal{R}_n$
6: **if** $L_{\mathcal{D}_{n+1}}(f_n) \geq \min_{f'\in\mathcal{F}} L_{\mathcal{D}_{n+1}}(f') + 5\beta_{\mathcal{F}}$ **then** Update $\mathcal{F}_{n+1}$ following (3)
7: **if** $L_{\mathcal{D}_{n+1}}(b_n) \geq \min_{b'\in\mathcal{R}} L_{\mathcal{D}_{n+1}}(b') + 5\beta_{\mathcal{R}}$ **then** Update $\mathcal{R}_{n+1}$ following (4).
8: **end for**

**Ensure:** Randomly pick up $n \in [N]$ uniformly and outputs $(\hat{\pi}, \hat{q})$ as $(\pi_n, q_n)$.

**Algorithm 3** $\text{PURE}_{\text{LowRollout}}$

**Require:** Total measurement number $N$, policy class $\Pi$, initial distribution class $\mathcal{Q}$, drift class $\mathcal{F}$, diffusion term $g^*$, reward class $\mathcal{R}$, confidence radius $\beta_{\mathcal{F}}$, $\beta_{\mathcal{R}}$, episode length $T$, measurement frequency $m$, sampler $\mathcal{T}$, initial measurement set $\mathcal{D}_1 = \emptyset$.

1: Initialize confidence sets $\mathcal{F}_1 = \mathcal{F}, \mathcal{R}_1 = \mathcal{R}$.
2: **for** episode $n = 1, \ldots, N/m$ **do**
3: Set $\pi_n, q_n, f_n, b_n \leftarrow \arg\max_{\pi\in\Pi, q\in\mathcal{Q}, f\in\mathcal{F}_n, b\in\mathcal{R}_n} R(\pi, q, f, b)$.
4: Sample $t_{n,1}, \ldots, t_{n,m} \sim \mathcal{T}$. Execute $q_n, \pi_n$ and receive measurement $(x_{n,i}, u_{n,i}, y_{n,i}, r_{n,i})$ at time $t_{n,i}$. Update $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup \{(x_{n,i}, u_{n,i}, y_{n,i}, r_{n,i})\}_{i=1}^m$.
5: Update $\mathcal{F}_{n+1}$ following (3), update $\mathcal{R}_{n+1}$ following (4).
6: **end for**

**Ensure:** Randomly pick up $n \in [N/m]$ uniformly and outputs $(\hat{\pi}, \hat{q})$ as $(\pi_n, q_n)$.

define the *independency coefficient* $C_{\mathcal{T},m}$ as $C_{\mathcal{T},m} := \sup_{i\in[m]} \max\{C_{\mathcal{T},m,\mathcal{F},i}, C_{\mathcal{T},m,\mathcal{R},i}\}$, where

$$C_{\mathcal{T},m,\mathcal{F},i} := \sup_{\bar{Z}_{i-1}\ldots,\bar{Z}_1,\pi,q} \frac{\mathbb{E}_{z_i\sim\mathbb{P}_{\bar{Z}}} \|f(z_i) - f^*(z_i)\|_2^2}{\mathbb{E}_{z_i\sim\mathbb{P}_{\bar{Z}_i|\bar{Z}_{i-1},\ldots,\bar{Z}_1}} \|f(z_i) - f^*(z_i)\|_2^2},$$

$$C_{\mathcal{T},m,\mathcal{R},i} := \sup_{\bar{Z}_{i-1}\ldots,\bar{Z}_1,\pi,q} \frac{\mathbb{E}_{z_i\sim\mathbb{P}_{\bar{Z}}} (b(z_i) - b^*(z_i))^2}{\mathbb{E}_{z_i\sim\mathbb{P}_{\bar{Z}_i|\bar{Z}_{i-1},\ldots,\bar{Z}_1}} (b(z_i) - b^*(z_i))^2},$$

Intuitively, $C_{\mathcal{T},m}$ quantifies how well the measurement times generated by $\mathcal{T}$ approximate those obtained from uniform sampling per rollout. The following proposition suggests that for certain continuous-time dynamical systems, $C_{\mathcal{T},m}$ can be upper bounded by a small constant. The proof is deferred to Appendix A.2.

**Proposition 5.5.** *There exists a one-dimensional continuous-time dynamical system with the control space $\mathcal{U}$ being one-dimensional and lower bounded by $u_{\min} > 0$, satisfying $C_{\mathcal{T},m} \leq 1 + \frac{m}{2Tu_{\min}}$. For this dynamical system, we have $C_{\mathcal{T},m} \leq 2$ when $m \leq 2Tu_{\min}$.*

Using $C_{\mathcal{T},m}$, we establish the following theoretical guarantee for $\text{PURE}_{\text{LowRollout}}$.

**Theorem 5.6.** *Setting the confidence radii $\beta_{\mathcal{F}}, \beta_{\mathcal{R}}$ as in Theorem 4.5, with probability at least $1 - \delta$, Algorithm 3 satisfies:*

- *For all $n \in [N/m]$, $f^* \in \mathcal{F}_n$ and $b^* \in \mathcal{R}_n$.*
- *The suboptimality gap of $(\hat{\pi}, \hat{q})$ is bounded by*

$$O\Bigg(\frac{T\sqrt{d_{\mathcal{R},N^{-1}}\beta_{\mathcal{R}}} + LT^{3/2}\sqrt{\exp(KT)}\sqrt{d_{\mathcal{F},N^{-1}}\beta_{\mathcal{F}}}}{\sqrt{N/\log N}} \cdot \sqrt{C_{\mathcal{T},m}} + \frac{mT(d_{\mathcal{F},N^{-1}} + d_{\mathcal{R},N^{-1}})}{N/\log N}\Bigg).$$

*where $K = 1 + (1 + L_\pi)^2 \cdot L_g^2 + 2(1 + L_\pi)^2 \cdot L_f^2$, $L = L_b(1 + L_\pi)$.*

The proof is deferred to Appendix D. By treating the Eluder dimension and covering numbers as constants, Theorem 5.6 suggests the following suboptimality gap:

$$\tilde{O}\Big(\frac{\sqrt{C_{\mathcal{T},m}}}{\sqrt{N}} + \frac{m}{N}\Big). \tag{6}$$

Comparing this result with the suboptimality gap in Theorem 4.5, we draw the following conclusions. First, the quality of the final output policy and initial distribution depends on the effectiveness of the sampler $\mathcal{T}$. The closer $\mathcal{T}$ is to generating i.i.d. samples, the more similar the performance of $\text{PURE}_{\text{LowRollout}}$ and $\text{PURE}_{\text{base}}$. Second, if the sampler $\mathcal{T}$ is sufficiently well-designed such that $C_{\mathcal{T},m} = O(1)$, then by (6), we can safely increase $m$ without significantly compromising the final policy performance.

# 6 EXPERIMENTS

In this section, we apply the principles of $\text{PURE}_{\text{base}}$, $\text{PURE}_{\text{LowSwitch}}$, and $\text{PURE}_{\text{LowRollout}}$ to several practical CTRL-based setups to evaluate their effectiveness. Specifically, we aim to answer the following question:

*Given the same number of measurements, can we reduce the total training time of CTRL by minimizing the number of policy updates and rollouts while maintaining comparable final performance to the original base algorithm?*

To investigate this, we conduct experiments across two distinct domains: (1) fine-tuning diffusion models and (2) classical continuous control tasks.

## 6.1 FINE-TUNING DIFFUSION MODELS

**Experiment Setup** We consider fine-tuning a diffusion model to generate images with enhanced aesthetic quality, as measured by aesthetic scores [Wightman, 2019, Radford et al., 2021, Liu et al., 2022b, Schuhmann et al., 2022, Black et al., 2023]. Our baseline fine-tuning approach is SEIKO [Uehara et al., 2024], a continuous-time reinforcement learning framework specifically tailored for optimizing diffusion models. SEIKO jointly refines the diffusion policy $\hat{\pi}$ and the initial distribution $\hat{q}$ by drawing samples from a pre-trained diffusion backbone and training a reward function $\hat{r}$ based on these samples, where the reward encodes the aesthetic score as its evaluation metric. This setup dovetails perfectly with our PURE framework under the assumption of fixed, known dynamics $f$. The measurement we receive is simplified to $(x, u, r)$ since $f$ does not need to be estimated. A key feature of SEIKO is its low-switching strategy, where episodes are divided into $\mathcal{K}$ batches, and the batch size $B_i$ is increased geometrically according to $B_{i+1} = \eta_{\text{base}} B_i$ for $i \in [\mathcal{K}]$. Due to space constraints, we defer details on SEIKO's backbone architecture and the training procedure for updating $\hat{\pi}, \hat{q}$, and $\hat{r}$ to Appendix F.

**Algorithm Implementation** We propose our algorithm, $\text{PURE}_{\text{SEIKO}}$, which builds upon SEIKO by incorporating an additional sampler, $\mathcal{T}$, from $\text{PURE}_{\text{LowRollout}}$ to reduce the number of rollouts and thereby lower the computational complexity of SEIKO. The full pseudo-code for $\text{PURE}_{\text{SEIKO}}$ is provided in Algorithm 4, with further details on SEIKO and our modifications available in Appendix F. Our sampler $\mathcal{T}$ operates based on a measurement frequency parameter $m$. At the $n$-th episode, it samples $\{t_{n,1}, \ldots, t_{n,m}\} \subseteq \{\frac{1}{m}T, \ldots, \frac{m-1}{m}T, T\}$. Each $t_{n,i}$ is drawn independently, with the probability of selecting $\frac{i}{m}T$ following a geometric distribution $\mathbb{P}\left(\frac{i}{m}T\right) \propto \lambda^i$, where $\lambda > 0$ is a tunable temperature parameter. In our experiments, we set $\lambda = 6$. Intuitively, geometric weighting aligns well with the exponential nature of information gain in reverse diffusion [Sohl-Dickstein et al., 2015], encouraging the agent to focus more on later time steps. We repeat the experiments across five random seeds.

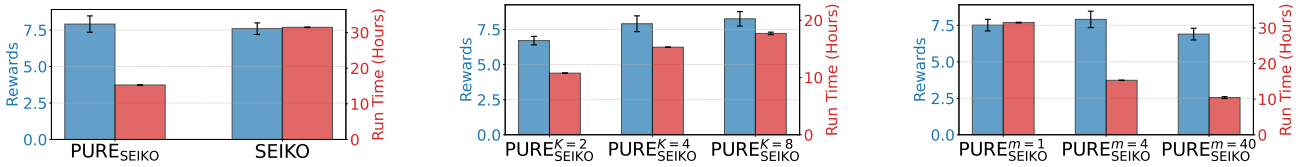
**Results** We compare $\text{PURE}_{\text{SEIKO}}$ with $m = 4$ and $\eta_{\text{base}} = 2$, where the diffusion model is updated $\mathcal{K} = 4$ times, generating only a single final image per trajectory. We evaluate the final aesthetic score of the generated image following our learned $\hat{\pi}, \hat{q}$ and compare it to the one produced by SEIKO. Additionally, we compare the total training time of $\text{PURE}_{\text{SEIKO}}$ and SEIKO. To ensure a fair comparison, both algorithms are run under the same update time $\mathcal{K}$ and with the same total number of measurements, $N = 19200$. Our results are summarized in Figure 1a. We report the mean and standard deviation of both reward and runtime over multiple seeds, demonstrating that $\text{PURE}_{\text{SEIKO}}$ achieves aesthetic scores comparable to SEIKO while requiring significantly fewer episodes. This reduction translates to approximately half the training time, demonstrating the efficiency of $\text{PURE}_{\text{SEIKO}}$.

**Ablation Study** To evaluate the impact of the number of policy updates $\mathcal{K}$ on both aesthetic reward and computational efficiency, we conduct an ablation study comparing $\text{PURE}_{\text{SEIKO}}$ with different $\eta_{\text{base}}$, setting $\mathcal{K} = 2, 4, 8$ under the same sample budget $N = 19200$. The results are summarized in Figure 1b. Our findings suggest that the number of policy updates exhibits a threshold at $\mathcal{K} = 4$. Specifically, for $\text{PURE}_{\text{SEIKO}}$ with $\mathcal{K} \geq 4$, the aesthetic score remains nearly unchanged even if $\mathcal{K}$ is reduced. However, when $\mathcal{K}$ is too small, e.g., $\mathcal{K} = 2$, the aesthetic score drops significantly. This empirical evidence supports our theoretical result in Theorem 5.1, which establishes a lower bound on the number of policy updates required.

Furthermore, we analyze the impact of the measurement frequency $m$ by comparing $\text{PURE}_{\text{SEIKO}}$ with $m = 1, 4, 40$. As illustrated in Figure 1c, increasing $m$ generally reduces the total training time by decreasing the total number of episodes. However, setting $m$ excessively high leads to performance degradation, suggesting that an optimal choice of $m$ is necessary to balance the quality of the generated images and training efficiency. This supports our claim in Theorem 5.6, which emphasizes the importance of selecting an appropriate $m$ for achieving the best trade-off.

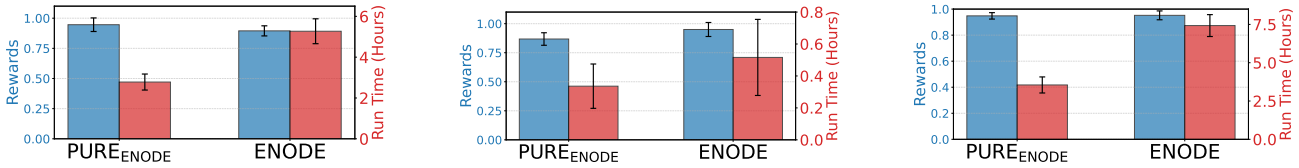
## 6.2 CONTINUOUS-TIME CONTROL

**Experiment Setup** We study continuous-time control tasks in the standard Gym benchmark [Brockman, 2016], focusing on three tasks: Acrobot, Pendulum, and CartPole. As our baseline model, the Ensemble Neural ODE (ENODE) [Yildiz et al., 2021] is used. The dynamical system is deterministic, and the reward function is known. ENODE employs a low-policy rollout strategy. The length of total observation time for a trajectory is $T = 50$(s), and the measurement frequency is $m = 250$. We defer details on ENODE's backbone architecture, training procedure and the details of the sampler $\mathcal{T}$ to Appendix G.2.




(a) $\text{PURE}_{\text{SEIKO}}$ vs. SEIKO (b) $\text{PURE}_{\text{SEIKO}}$ with varying $\mathcal{K}$ (c) $\text{PURE}_{\text{SEIKO}}$ with varying $m$

Figure 1: Summary of the experiment for fine-tuning Diffusion Models. 1a presents a comparison of aesthetic scores for denoised images generated by the fine-tuned Diffusion policy. 1b and 1c show ablation studies examining the effects of the number of policy updates and the value of $m$ on the final reward.




(a) Comparison on Acrobot (b) Comparison on Pendulum (c) Comparison on Cart Pole

Figure 2: Summary of continuous-time control experiments, $\text{PURE}_{\text{ENODE}}$ vs. ENODE, in three control environments.

**Algorithm Implementation** We implement $\text{PURE}_{\text{ENODE}}$ based on ENODE, incorporating the low-policy update strategy introduced in $\text{PURE}_{\text{LowSwitch}}$. Specifically, we adopt a batch-like strategy similar to SEIKO, as described in Section 6.1, to reduce the frequency of policy updates. In this approach, the batch size $B_i$ is doubled at each step, following $B_{i+1} = 2B_i$. To ensure the same sample budget $N$, we may slightly modify the doubling batch size strategy according to the environments. Further details on the algorithmic implementation, dataset collection, and policy updates for our experiments are provided in Appendix G.3.

**Results** Our main results are presented in Figure 2. We report both the mean and standard derivation for both reward and running time from 20 seeds. Empirically, $\text{PURE}_{\text{ENODE}}$ reaches the target state using only $1/2$ to $1/4$ of the policy update steps required by standard ENODE. This reduction also leads to nearly a $50\%$ decrease in training time. Such efficiency gains align with our theoretical predictions in the main theorems. To further evaluate the effectiveness of the policy update strategy, we conducted ablation study about the measurement frequency $m$ and different batch update scheduling strategy with varing policy updates. The corresponding results are deferred to Appendix G.4.

## 7 CONCLUSION AND LIMITATIONS

**Conclusion** In this work, we study CTRL with general function approximation under a finite number of measurements, aiming to reduce its computational cost. We develop a theoretical framework to propose our algorithm, PURE, along with a finite-sample analysis. Additionally, we introduce several variants of our algorithm with improved computational efficiency. Empirical results on continuous-time control tasks and fine-tuning of diffusion models backup our theoretical findings.

**Limitations** While our analysis yields useful insights for sample- and computationally efficient CTRL, it does come with several caveats. First, the regret bound grows on the order of $\exp(T)$, which may render it vacuous for long time horizons. Second, our theoretical guarantees hinge on the Euler–Maruyama discretization [Platen and Bruti-Liberati, 2010] of the underlying SDE, introducing unquantified bias from discretization error that could degrade performance in practice. Finally, we assume access to jointly measured observations $(x(t), x(t+\Delta))$ following Treven et al. [2024a], which may be infeasible in systems with asynchronous sensors or communication delays. We leave these challenges as directions for future work.

## References


Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Marc Abeille and Alessandro Lazaric. Efficient optimistic exploration in linear-quadratic regulators via lagrangian relaxation. In *International Conference on Machine Learning*, pages 23–31. PMLR, 2020.

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

P Auer. Finite-time analysis of the multiarmed bandit problem, 2002.

Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32, 2019.

Matteo Basei, Xin Guo, Anran Hu, and Yufei Zhang. Logarithmic regret for episodic continuous-time linear-quadratic reinforcement learning over a finite-time horizon. *Journal of Machine Learning Research*, 23(178): 1–34, 2022.

Richard Bellman. The stability of solutions of linear differential equations. *Duke Mathematical Journal*, 10(4):643 – 647, 1943.

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

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

G Brockman. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Peter E. Caines and David Levanony. Stochastic $\varepsilon$-optimal linear quadratic adaptation: An alternating controls policy. *SIAM Journal on Control and Optimization*, 57(2):1094–1126, 2019.

Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. *Advances in Neural Information Processing Systems*, 26, 2013.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *International Conference on Machine Learning*, pages 1029–1038. PMLR, 2018.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

Minbo Gao, Tianle Xie, Simon S Du, and Lin F Yang. A provably efficient algorithm for linear markov decision process with low switching cost. *arXiv preprint arXiv:2101.00494*, 2021.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Jiafan He, Heyang Zhao, Dongruo Zhou, and Quanquan Gu. Nearly minimax optimal reinforcement learning for linear markov decision processes. In *International Conference on Machine Learning*, pages 12790–12822. PMLR, 2023.

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

Samuel Holt, Alihan Hüyük, and Mihaela van der Schaar. Active observing in continuous-time control. *Advances in Neural Information Processing Systems*, 36, 2024.

Jiawei Huang, Jinglin Chen, Li Zhao, Tao Qin, Nan Jiang, and Tie-Yan Liu. Towards deployment-efficient reinforcement learning: Lower bound and optimality. *arXiv preprint arXiv:2202.06450*, 2022.

Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Sublinear regret for a class of continuous-time linear–quadratic reinforcement learning problems. *arXiv preprint arXiv:2407.17226*, 2024.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dingwen Kong, Ruslan Salakhutdinov, Ruosong Wang, and Lin F Yang. Online sub-sampling for reinforcement learning with general function approximation. *arXiv preprint arXiv:2106.07203*, 2021.

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

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos Theodorou, Weili Nie, and Anima Anandkumar. $I^2$ sb: Image-to-image schrödinger bridge. In *International Conference on Machine Learning*, pages 22042–22062. PMLR, 2023.

Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*, 2022a.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Michael Lutter, Shie Mannor, Jan Peters, Dieter Fox, and Animesh Garg. Value iteration in continuous actions, states and time. In *International Conference on Machine Learning*, pages 7224–7234. PMLR, 2021.

Eckhard Platen and Nicola Bruti-Liberati. *Numerical solution of stochastic differential equations with jumps in finance*, volume 64. Springer Science & Business Media, 2010.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

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

Yufei Ruan, Jiaqi Yang, and Yuan Zhou. Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–87, 2021.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

C. Schuhmann. Laion aesthetic predictor. `https://laion.ai/blog/laion-aesthetics/`, 2022. Accessed: 2024-09-29.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=M3Y74vmsMcY`.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

Mohamad Kazem Shirani Faradonbeh and Mohamad Sadegh Shirani Faradonbeh. Online reinforcement learning in stochastic continuous-time systems. In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 612–656. PMLR, 12–15 Jul 2023.

Max Simchowitz and Dylan Foster. Naive exploration is optimal for online lqr. In *International Conference on Machine Learning*, pages 8937–8948. PMLR, 2020.

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

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. In *Uncertainty in Artificial Intelligence*, pages 1985–1995. PMLR, 2023.

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

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

Lukasz Szpruch, Tanut Treetanthiploet, and Yufei Zhang. Optimal scheduling of entropy regularizer for continuous-time linear-quadratic reinforcement learning. *SIAM Journal on Control and Optimization*, 62(1):135–166, 2024.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Lenart Treven, Jonas Hübotter, Florian Dorfler, and Andreas Krause. Efficient exploration in continuous-time model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024a.

Lenart Treven, Bhavya Sukhija, Yarden As, Florian Dörfler, and Andreas Krause. When to sense and control? a time-adaptive approach for continuous-time rl. *arXiv preprint arXiv:2406.01163*, 2024b.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Sergey Levine, and Tommaso Biancalani. Feedback efficient online fine-tuning of diffusion models. *arXiv preprint arXiv:2402.16359*, 2024.

Draguna Vrabie and Frank Lewis. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Networks*, 22(3):237–246, 2009.

Kaiwen Wang, Kevin Zhou, Runzhe Wu, Nathan Kallus, and Wen Sun. The benefits of being distributional: Small-loss bounds for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Tianhao Wang, Dongruo Zhou, and Quanquan Gu. Provably efficient reinforcement learning with linear function approximation under adaptivity constraints. *Advances in Neural Information Processing Systems*, 34:13524–13536, 2021.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4230–4239, 2023.

Nuoya Xiong, Zhaoran Wang, and Zhuoran Yang. A general framework for sequential decision-making under adaptivity constraints. In *Forty-first International Conference on Machine Learning*, 2023.

Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In *International Conference on Machine Learning*, pages 12009–12018. PMLR, 2021.

TaeHo Yoon, Kibeom Myoung, Keon Lee, Jaewoong Cho, Albert No, and Ernest Ryu. Censored sampling of diffusion models using 3 minutes of human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Zihan Zhang, Xiangyang Ji, and Simon Du. Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. In *Conference on Learning Theory*, pages 4528–4531. PMLR, 2021a.

Zihan Zhang, Jiaqi Yang, Xiangyang Ji, and Simon S Du. Improved variance-aware confidence sets for linear bandits and linear mixture mdp. *Advances in Neural Information Processing Systems*, 34:4342–4355, 2021b.

Heyang Zhao, Jiafan He, and Quanquan Gu. A nearly optimal and low-switching algorithm for reinforcement learning with general function approximation. *arXiv preprint arXiv:2311.15238*, 2023.

Yaofeng Desmond Zhong and Naomi Leonard. Unsupervised learning of lagrangian dynamics from images for prediction and control. *Advances in Neural Information Processing Systems*, 33:10741–10752, 2020.

# A PROOF OF PROPOSITIONS

## A.1 PROOF OF PROPOSITION 4.4

*Proof.* Note that for any $p \in \mathcal{P}$, $f \in \mathcal{F}$, $b \in \mathcal{R}$, we have

$$\mathbb{E}_{z\sim p}\|f(z) - f^*(z)\|_2^2 = \mathbb{E}_{z\sim p}\phi(z)^\top(\Theta - \Theta^*)(\Theta - \Theta^*)^\top\phi(z) = \mathbb{E}_{z\sim p}\langle(\Theta - \Theta^*)(\Theta - \Theta^*)^\top, \phi(z)\phi(z)^\top\rangle,$$

and similarily,

$$\mathbb{E}_{z\sim p}(b(z) - b^*(z))^2 = \mathbb{E}_{z\sim p}\langle(\theta - \theta^*)(\theta - \theta^*)^\top, \phi(z)\phi(z)^\top\rangle.$$

Therefore, $d_{\mathcal{F},\epsilon}$ can be bounded by $\mathrm{DE}_1(\mathcal{Z}, \bar{\mathcal{F}}, \mathcal{P}, \epsilon)$, where $\bar{\mathcal{F}} = \{f_\theta(z) = \langle\bar{\Theta}, \bar{\phi}(z)\rangle : \|\bar{\Theta}\|_F \le R^2, \|\bar{\phi}\|_2 \le L^2, \phi \in \mathbb{R}^{d^2}\}$. Therefore, first through Lemma 5.4 in Wang et al. [2023], we have $\mathrm{DE}_1(\mathcal{Z}, \bar{\mathcal{F}}, \mathcal{P}, \epsilon) \le \mathrm{DE}_2(\mathcal{Z}, \bar{\mathcal{F}}, \mathcal{P}, \epsilon)$. Then according to Proposition 29 in Jin et al. [2021], we have $\mathrm{DE}_2(\mathcal{Z}, \bar{\mathcal{F}}, \mathcal{P}, \epsilon) = O(d^2\log(1 + R^2L^2/\epsilon^2))$ for any $\mathcal{P}$. Similar arguments hold for $\mathcal{R}$ as well. □

## A.2 PROOF OF PROPOSITION 5.5

*Proof.* Consider a sampler $\mathcal{T}$ that selects time points $t_1, \ldots, t_m$ uniformly in $[0, T]$, i.e., $t_i = \frac{i}{m} \cdot T$. We consider an one-dimensional Ornstein–Uhlenbeck (OU) process follows

$$dx(t) = -u \cdot x(t)dt + \sqrt{2}dw(t), \quad x(0) = 0,$$

where $u_{\min} \le u \le u_{\max}$. Our dynamic class $\mathcal{F}$ is a singleton, and the reward class is defined as $\mathcal{R} = \{b(x, u) = \alpha \cdot x \mid 0 \le \alpha \le 1\}$ with $b^*(x, u) = x$. We have $C_{\mathcal{T},m,\mathcal{F},i} = 1$. To bound $C_{\mathcal{T},m,\mathcal{R},i}$, note that $(b - b^*)^2 = (1 - \alpha)^2x^2$. For simplicity, we use $z$ to denote $x$ and omit $u$ since it is a constant control unit. Then the Gaussian marginal distribution of the OU process gives

$$\mathbb{P}(\hat{Z} = z) = \frac{1}{T}\int_0^T \sqrt{\frac{u}{2\pi(1 - e^{-2ut})}}\exp\left(-\frac{uz^2}{2(1 - e^{-2ut})}\right)dt.$$

Since the OU process is a Markov process, we have

$$\mathbb{P}(Z(t_i) = z \mid Z(t_{i-1}) = z_{i-1}, \ldots, Z(t_1) = z_1) = \mathbb{P}(Z(t_i) = z \mid Z(t_{i-1}) = z_{i-1}).$$

The transition density is given by

$$\mathbb{P}(Z(t_i) = z \mid Z(t_{i-1}) = z_{i-1}) = \sqrt{\frac{u}{2\pi(1 - e^{-2uT/m})}}\exp\left(-\frac{u(z - z_{i-1}e^{-uT/m})^2}{2(1 - e^{-2uT/m})}\right).$$

We compute the expectations over $(b - b^*)^2$. For $\hat{Z}$,

$$(1 - \alpha)^{-2}\mathbb{E}[(b - b^*)^2(\hat{Z})] = \mathbb{E}[\hat{Z}^2] = \mathbb{E}[\mathbb{E}[Z_t^2 \mid t]] = \int_0^T \frac{1}{u}(1 - e^{-2ut})\frac{1}{T}dt = \frac{1}{u} - \frac{1}{2u^2T}(1 - e^{-2uT}).$$

For $Z(t_i)$,

$$\begin{aligned}(1 - \alpha)^{-2}\mathbb{E}[(b(Z(t_i)) - b^*(Z(t_i)))^2 \mid Z(t_{i-1}) = z_{i-1}] &= \mathbb{E}[Z(t_i)^2 \mid Z(t_{i-1}) = z_{i-1}] \\ &= (z_{i-1}e^{-uT/m})^2 + \frac{1}{u}(1 - e^{-2uT/m}).\end{aligned}$$

Then the ratio is bounded by

$$\begin{aligned}\frac{\mathbb{E}[(b - b^*)^2(\hat{Z})]}{\mathbb{E}[(b(Z(t_i)) - b^*(Z(t_i)))^2 \mid Z(t_{i-1}) = z_{i-1}]} &\le \frac{\frac{1}{u} - \frac{1}{2u^2T}(1 - e^{-2uT})}{\frac{1}{u}(1 - e^{-2uT/m})} \\ &\le \frac{1}{1 - e^{-2uT/m}} \\ &\le 1 + \frac{m}{2Tu_{\min}},\end{aligned}$$

which implies $C_{\mathcal{T},m,\mathcal{R},i} \le 1 + \frac{m}{2Tu_{\min}}$ for all $i$. Thus, we obtain the upper bound for $C_{\mathcal{T},m\,\mathcal{R}}$. □

## B PROOF OF THEOREM 4.5

In this section we prove Theorem 4.5. To make our presentation more clear, we separate Theorem 4.5 into two theorems Theorem B.4 and Theorem B.5 and prove them separately. To begin with, we have the following lemma to bound the flow first.

**Lemma B.1.** *Denote $\hat{x}(t)$ to be the state flow that following $f_n, \pi_n, q_n$, and let $x(t)$ denote the state flow following $f^*, \pi_n, q_n$. Then we have*

$$\mathbb{E}\|\hat{x}_n(t) - x_n(t)\|_2^2 \leq 2e^{Kt} \cdot \int_{s=0}^{t} \mathbb{E}[\|f^*(x_n(s), \pi_n(x_n(s))) - f_n(x_n(s), \pi_n(x_n(s)))\|_2^2] ds.$$

*where $K = 1 + d \cdot (1 + L_\pi)^2 \cdot L_g^2 + 2(1 + L_\pi)^2 \cdot L_f^2$.*

*Proof.* Define $\delta(t) := \hat{x}_n(t) - x_n(t)$. The dynamics of $\delta(t)$ are governed by:

$$\begin{aligned} d\delta(t) = &[f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))]\, dt \\ &+ [g^*(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - g^*(x_n(t), \pi_n(x_n(t)))]\, dw(t). \end{aligned} \tag{7}$$

Applying Itô's lemma to $\|\delta(t)\|_2^2$ yields:

$$d\|\delta(t)\|_2^2 = 2\delta(t)^\top d\delta(t) + d \cdot |g^*(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - g^*(x_n(t), \pi_n(x_n(t)))|^2 dt.$$

Taking integration on both sides:

$$\|\delta(t)\|_2^2 = \|\delta(0)\|_2^2 + \int_0^t 2\delta(t)^\top d\delta(t) + d \cdot \int_0^t |g^*(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - g^*(x_n(t), \pi_n(x_n(t)))|^2 dt.$$

Taking expectations, the martingale term corresponding to (7) vanishes, then apply Fubini's theorem for other terms leading to:

$$\begin{aligned} \frac{d}{dt}\mathbb{E}\|\delta(t)\|_2^2 = &2\mathbb{E}\left[\delta(t)^\top \left(f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\right)\right] \\ &+ d \cdot \mathbb{E}|g^*(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - g^*(x_n(t), \pi_n(x_n(t)))|^2. \end{aligned}$$

Bounding the first term using Cauchy-Schwarz and $2ab \leq a^2 + b^2$:

$$\begin{aligned} &2\mathbb{E}\left[\delta(t)^\top \left(f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\right)\right] \\ &\leq 2\mathbb{E}\left[\|\delta(t)\|_2 \cdot \|f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\|_2\right] \\ &\leq \mathbb{E}[\|\delta(t)\|_2^2] + \mathbb{E}[\|f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\|_2^2]. \end{aligned}$$

To leverage the $L_f$-Lipschitzness of $f_n, f^*$ (Assumption 4.2), we bound the term:

$$\begin{aligned} &\mathbb{E}\left\|f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\right\|_2^2 \\ &\leq 2\mathbb{E}[\|f_n(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - f_n(x_n(t), \pi_n(x_n(t)))\|_2^2] + 2\mathbb{E}[\|f_n(x_n(t), \pi_n(x_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\|_2^2] \\ &\leq 2L_f^2(1 + L_\pi)^2\mathbb{E}[\|\delta(t)\|_2^2] + 2\mathbb{E}[\|f_n(x_n(t), \pi_n(x_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\|_2^2], \end{aligned}$$

where the first inequality is due to the fact that $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, the second inequality is due to Assumption 4.2. Similarly for the diffusion term, by $g^*$ is $L_g$-Lipschitz:

$$\mathbb{E}|g^*(\hat{x}_n(t), \pi_n(\hat{x}_n(t))) - g^*(x_n(t), \pi_n(x_n(t)))|^2 \leq (1 + L_\pi)^2 \cdot L_g^2 \mathbb{E}\|\delta(t)\|_2^2.$$

Combining above:

$$\frac{d}{dt}\mathbb{E}\|\delta(t)\|_2^2 \leq (1 + d \cdot (1 + L_\pi)^2 \cdot L_g^2 + 2(1 + L_\pi)^2 \cdot L_f^2)\mathbb{E}\|\delta(t)\|_2^2 + 2\mathbb{E}\left\|f_n(x_n(t), \pi_n(x_n(t))) - f^*(x_n(t), \pi_n(x_n(t)))\right\|_2^2.$$

Applying Grönwall's inequality with $K = 1 + (1 + L_\pi)^2 \cdot L_g^2 + 2(1 + L_\pi)^2 \cdot L_f^2$:

$$\mathbb{E}\|\delta(t)\|_2^2 \leq 2e^{Kt} \int_0^t \mathbb{E}\left\|f_n(x_n(s), \pi_n(x_n(s))) - f^*(x_n(s), \pi_n(x_n(s)))\right\|_2^2 ds.$$

□

We also have the following lemma to show the difference between $b - b^*$ with their empirical one:

**Lemma B.2.** *[Lemma 1,5, Russo and Van Roy 2013] For any $\delta > 0$ and $\epsilon > 0$, let $\Phi_\epsilon$ be the $\epsilon$-covering net of $\Phi$ (for any $\phi \in \Phi$, there exists a $\phi_\epsilon \in \Phi_\epsilon$ such that $\|\phi - \phi_\epsilon\|_\infty \le \epsilon$). Then with probability at least $1 - \delta$, we have*

- *For all $b \in \mathcal{R}$, recall that $|b| \le 1$ and $r$ is 1-Gaussian, then we have for all $n$,*

$$\sum_{z,r\in\mathcal{D}_n} (b(z)-r)^2 - (b^*(z)-r)^2 \ge \frac{1}{2}\sum_{z\in\mathcal{D}_n}(b(z)-b^*(z))^2 - 4\log(|\mathcal{R}_\epsilon|/\delta) - \epsilon n(8+\sqrt{8\log(4n^2|\mathcal{R}_\epsilon|/\delta)}),$$
$$\sum_{z,r\in\mathcal{D}_n} (b(z)-r)^2 - (b^*(z)-r)^2 \le \frac{3}{2}\sum_{z\in\mathcal{D}_n}(b(z)-b^*(z))^2 + 4\log(|\mathcal{R}_\epsilon|/\delta) + \epsilon n(\epsilon+8+\sqrt{8\log(4n^2|\mathcal{R}_\epsilon|/\delta)}),$$

- *For all $f \in \mathcal{F}$, recall that $\|f\|_2 \le 1$ and $y$ is $g(z)\cdot$ **1**-Gaussian vector, $|g| \le G/\sqrt{d}$, then we have for all $n$,*

$$\sum_{z\in\mathcal{D}_n}\|f(z)-f^*(z)\|^2 \le 2\Bigg(\sum_{z,y\in\mathcal{D}_n}\|f(z)-y\|^2 - \sum_{z,y\in\mathcal{D}_n}\|f^*(z)-y\|^2\Bigg) + 8G^2\log(|\mathcal{F}_\epsilon|/\delta) + 2\epsilon n(8+\sqrt{8G^2\log(4n^2|\mathcal{F}_\epsilon|/\delta)}).$$

$$\sum_{z,y\in\mathcal{D}_n}\|f(z)-y\|^2 - \|f^*(z)-y\|^2 \le \frac{3}{2}\sum_{z\in\mathcal{D}_n}\|f(z)-f^*(z)\|^2 + 4G^2\log(|\mathcal{F}_\epsilon|/\delta) + \epsilon n(\epsilon+8+\sqrt{8G^2\log(4n^2|\mathcal{F}_\epsilon|/\delta)}).$$

*Proof.* To prove the first part, let $\mathcal{R}_\epsilon \subset \mathcal{R}$ be a finite set such that for every $b \in \mathcal{R}$ there exists some $b_\epsilon \in \mathcal{R}_\epsilon$ with $\|b - b_\epsilon\|_\infty \le \epsilon$.
Recall the notations $L_{2,n}(b) := \sum_{z,r\in\mathcal{D}_n}(b(z)-r)^2$ and $\|b-b^*\|^2_{2,\mathcal{D}_n} := \sum_{z\in\mathcal{D}_n}(b(z)-b^*(z))^2$ from Russo and Van Roy 2013. Following the proof of Lemma 1 of Russo and Van Roy 2013, for each $b_\epsilon \in \mathcal{R}_\epsilon$, with $\eta^2 = 1$ as our 1-Gaussian assumption, with probability at least $1 - \frac{\delta}{|\mathcal{R}_\epsilon|}$:

$$\|b_\epsilon - b^*\|^2_{2,\mathcal{D}_n} \le 2\left(L_{2,n}(b_\epsilon) - L_{2,n}(b^*)\right) + 8\ln\left(\frac{|\mathcal{R}_\epsilon|}{\delta}\right), \tag{8}$$

taking union bound over $\mathcal{R}_\epsilon$ we then have with probability at least $1-\delta$, every $b_\epsilon \in \mathcal{R}_\epsilon$ satisfies above inequality.
Now let $b \in \mathcal{R}$ be arbitrary. By the definition of the $\epsilon$-net, pick $b_\epsilon \in \mathcal{R}_\epsilon$ with $\|b - b_\epsilon\|_\infty \le \epsilon$.
Since $\|b - b_\epsilon\|_\infty \le \epsilon$ and $\|b\|_\infty \le 1$ and the reward noise variance is $\eta^2 = 1$, apply Lemma 5 of Russo and Van Roy 2013 and taking union bound we have with probability at least $1-\delta$ for all $n \in \mathbb{N}$:

$$\left|\frac{1}{2}\|b_\epsilon - b^*\|^2_{2,\mathcal{D}_n} - \frac{1}{2}\|b-b^*\|^2_{2,\mathcal{D}_n} + L_{2,n}(b) - L_{2,n}(b_\epsilon)\right| \le \epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right], \tag{9}$$

We now connect $\|b-b^*\|^2$ to $\|b_\epsilon - b^*\|^2$. By (9):

$$\begin{aligned}\frac{1}{2}\|b-b^*\|^2_{2,\mathcal{D}_n} &= \frac{1}{2}\|b_\epsilon-b^*\|^2_{2,\mathcal{D}_n} - (L_{2,n}(b_\epsilon) - L_{2,n}(b)) + \left[\frac{1}{2}\|b-b^*\|^2 - \frac{1}{2}\|b_\epsilon-b^*\|^2 + L_{2,n}(b_\epsilon) - L_{2,n}(b)\right] \\ &\le \frac{1}{2}\|b_\epsilon-b^*\|^2_{2,\mathcal{D}_n} - (L_{2,n}(b_\epsilon) - L_{2,n}(b)) + \epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right].\end{aligned}$$

Multiply both sides by 2:

$$\|b-b^*\|^2_{2,\mathcal{D}_n} \le \|b_\epsilon-b^*\|^2_{2,\mathcal{D}_n} - 2(L_{2,n}(b_\epsilon) - L_{2,n}(b)) + 2\epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right].$$

Then apply (8) for $\|b_\epsilon - b^*\|^2_{2,\mathcal{D}_n}$:

$$\|b-b^*\|_{2,\mathcal{D}_n}^2 \le 2(L_{2,n}(b_\epsilon)-L_{2,n}(b^*))+8\ln\left(\tfrac{|\mathcal{R}_\epsilon|}{\delta}\right)-2(L_{2,n}(b_\epsilon)-L_{2,n}(b))+2\epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right]$$
$$=2(L_{2,n}(b)-L_{2,n}(b^*))+8\ln\left(\tfrac{|\mathcal{R}_\epsilon|}{\delta}\right)+2\epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right]$$
$$=2\Big(\sum_{z,r}(b-r)^2-\sum_{z,r}(b^*(z)-r)^2\Big)\ +\ 8\ \ln\Big(\tfrac{|\mathcal{R}_\epsilon|}{\delta}\Big)\ +\ 2\epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right].$$

This proves the first inequality. To prove the second inequality, first note that by modifying the proof of Lemma 1 in Russo and Van Roy 2013, namely setting $Z_t := f(A_t)-R_t)^2-(f_\theta(A_t)-R_t)^2$ which is the negative of original $Z_t$, we can go through the same arguments as the proof in Russo and Van Roy 2013 and arrive at the conclusion that with probability at least $1-\delta$, for every $b_\epsilon \in \mathcal{R}_\epsilon$ we have:

$$L_{2,n}(b_\epsilon)-L_{2,n}(b^*) \le \|b_\epsilon-b^*\|_{2,\mathcal{D}_n}^2+4\eta^2\ln\left(\frac{|\mathcal{R}_\epsilon|}{\delta}\right).$$

To combine this with Lemma 5 of Russo and Van Roy 2013, we decompose as:

$$\begin{aligned}L_{2,n}(b)-L_{2,n}(b^*) &= (L_{2,n}(b)-L_{2,n}(b_\epsilon))+(L_{2,n}(b_\epsilon)-L_{2,n}(b^*))\\ &= \frac{1}{2}\|b-b^*\|_{2,\mathcal{D}_n}^2-\frac{1}{2}\|b_\epsilon-b^*\|_{2,\mathcal{D}_n}^2+\left[\frac{1}{2}\|b_\epsilon-b^*\|^2-\frac{1}{2}\|b-b^*\|^2+L_{2,n}(b)-L_{2,n}(b_\epsilon)\right]\\ &\quad+(L_{2,n}(b_\epsilon)-L_{2,n}(b^*))\\ &\le \frac{1}{2}\|b-b^*\|_{2,\mathcal{D}_n}^2+\epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right]+\frac{1}{2}\|b_\epsilon-b^*\|_{2,\mathcal{D}_n}^2+4\ln\left(\frac{|\mathcal{R}_\epsilon|}{\delta}\right).\end{aligned}$$

To bound $\|b_\epsilon-b^*\|_{2,\mathcal{D}_n}^2$, note that $(b_\epsilon-b^*)^2 \le 2(b_\epsilon-b)^2+2(b-b^*)^2$ and thus we can bound as:

$$\frac{1}{2}\|b_\epsilon-b^*\|_{2,\mathcal{D}_n}^2 \le n\epsilon^2+\|b-b^*\|_{2,\mathcal{D}_n}^2.$$

Plug in and we obtain the bound:

$$L_{2,n}(b)-L_{2,n}(b^*) \le \frac{3}{2}\|b-b^*\|_{2,\mathcal{D}_n}^2+\epsilon n\left[8+\sqrt{8\ln\left(\frac{4n^2|\mathcal{R}_\epsilon|}{\delta}\right)}\right]+n\epsilon^2+4\ln\left(\frac{|\mathcal{R}_\epsilon|}{\delta}\right).$$

Thus we prove the first part of the lemma. The proof for the second part of the lemma is by using the same arguments, except replacing the noise variance with $\eta^2=G^2$, which is simply by adding the extra $G^2$ coefficients to the bound. □

**Lemma B.3** (Theorem 5.3, Wang et al. 2023)**.** *Given a function class $\Phi$ defined on $\mathcal{Z}$ with $|\phi(x)| \le 1$ for all $(\phi,z)\in\Phi\times\mathcal{Z}$, and a family of probability measures $\mathcal{P}$ over $\mathcal{Z}$. Suppose sequence $\{\phi_k\}_{k=1}^K \subset \Phi$ and $\{p_k\}_{k=1}^K \subset \mathcal{P}$ satisfy that for all $k\in[K]$, $\sum_{i=1}^{k-1}|\mathbb{E}_{p_i}[\phi_k]| \le \beta$. Then for all $k\in[K]$,*

$$\sum_{i=1}^{k}|\mathbb{E}_{p_i}[\phi_i]| \le O(DE_1(\mathcal{Z},\Phi,\mathcal{P},1/k)\beta\log k)$$

Next we are going to prove our theorems.

**Theorem B.4.** *Let*

$$\beta_{\mathcal{R}} := 8\log(|\mathcal{R}_\epsilon|/\delta)+2\epsilon N(8+\sqrt{8\log(4N^2|\mathcal{R}_\epsilon|/\delta)})$$
$$\beta_{\mathcal{F}} := 8G^2\log(|\mathcal{F}_\epsilon|/\delta)+2\epsilon N(8+\sqrt{8G^2\log(4N^2|\mathcal{F}_\epsilon|/\delta)}),$$

*then we have $f^*\in\mathcal{F}_n$ and $b^*\in\mathcal{R}_n$ for all $n$ w.h.p.*

*Proof.* Recall the definition of $\mathcal{R}_n$ and $\mathcal{F}_n$:

$$\mathcal{R}_n \leftarrow \left\{ b : \sum_{z,y,r \in \mathcal{D}_n} (b(z) - r)^2 \leq \min_{b' \in \mathcal{R}} \sum_{z,y,r \in \mathcal{D}_n} (b'(z) - r)^2 + \beta_{\mathcal{R}} \right\}$$
$$\mathcal{F}_n \leftarrow \left\{ f : \sum_{z,y,r \in \mathcal{D}_n} \|f(z) - y\|^2 \leq \min_{f' \in \mathcal{F}} \sum_{z,y,r \in \mathcal{D}_n} \|f'(z) - y\|^2 + \beta_{\mathcal{F}} \right\}.$$

Following Lemma B.2, we have that for any $b \in \mathcal{R}_n$, we have

$$\begin{aligned}
&\sum_{z,y,r \in \mathcal{D}_n} (b^*(z) - r)^2 - \sum_{z,y,r \in \mathcal{D}_n} (b(z) - r)^2 \\
&\leq -\frac{1}{2} \sum_{z \in \mathcal{D}_n} (b(z) - b^*(z))^2 + 8 \log(|\mathcal{R}_\epsilon|/\delta) + 2\epsilon n(8 + \sqrt{8 \log(4n^2 |\mathcal{R}_\epsilon|/\delta)}) \\
&\leq 8 \log(|\mathcal{R}_\epsilon|/\delta) + 2\epsilon n(8 + \sqrt{8 \log(4n^2 |\mathcal{R}_\epsilon|/\delta)}) \\
&\leq \beta_{\mathcal{R}}
\end{aligned}$$

following the definition of $\beta_{\mathcal{R}}$. Therefore, we have $b^* \in \mathcal{R}_n$. Similarily, we have $f^* \in \mathcal{F}_n$. □

We have the following theorem.

**Theorem B.5.** *With probability at least* $1 - 2\delta \log N$*, we have*

$$\begin{aligned}
\sum_{n=1}^{N} R_n = O(&T \sqrt{N d_{\mathcal{R}} (\log(4N/\delta) + \log(|\mathcal{R}_\epsilon|/\delta)) \log N} \\
&+ LT \cdot \sqrt{TN \cdot 2 \exp(KT)} \sqrt{d_{\mathcal{F}} G^2 (\log(4N/\delta) + \log(|\mathcal{F}_\epsilon|/\delta)) \log N}),
\end{aligned}$$

*where* $\epsilon = \frac{1}{N^2}, K = 1 + d \cdot (1 + L_\pi)^2 \cdot L_g^2 + 2(1 + L_\pi)^2 \cdot L_f^2, L = L_b(1 + L_\pi)$.

*Proof.* For simplicity, denote $\hat{x}(t)$ to be the state flow that following $f_n, \pi_n, q_n$, and let $x(t)$ denote the state flow following $f^*, \pi_n, q_n$. We introduce the notion of $R(f, b, \pi, q)$ to denote

$$R(f, b, \pi, q) := \mathbb{E}\left[ \int_{t=0}^{T} b(x(t), \pi(x(t)))dt \middle| x(0) \sim q, dx(t) = f(x(t), \pi(x(t))dt + g^*(x(t), \pi(x(t))dw(t) \right],$$

Then we have

$$\begin{aligned}
R_n :&= R(f^*, r^*, \pi^*, q^*) - R(f^*, r^*, \pi_n, q_n) \\
&\leq R(f_n, b_n, \pi_n, q_n) - R(f^*, r^*, \pi_n, q_n) \\
&= \mathbb{E}\left[ \int_{t=0}^{T} b_n(\hat{x}(t), \pi_n(\hat{x}(t)))dt \right] - \mathbb{E}\left[ \int_{t=0}^{T} b^*(x(t), \pi_n(x(t)))dt \right] \\
&= \mathbb{E}\left[ \int_{t=0}^{T} b_n(\hat{x}(t), \pi_n(\hat{x}(t)))dt \right] - \mathbb{E}\left[ \int_{t=0}^{T} b_n(x(t), \pi_n(x(t)))dt \right] + \mathbb{E}\left[ \int_{t=0}^{T} (b_n - b^*)(x(t), \pi_n(x(t)))dt \right] \\
&\leq L_b(1 + L_\pi) \cdot \mathbb{E}\left[ \int_{t=0}^{T} \|\hat{x}(t) - x(t)\|_2 dt \right] + \mathbb{E}\left[ \int_{t=0}^{T} (b_n - b^*)(x(t), \pi_n(x(t)))dt \right],
\end{aligned}$$

where the first inequality holds since $f^* \in \mathcal{F}_n, b^* \in \mathcal{R}_n$ and the optimism principle, and the last one holds due to Assumption 4.2. By Lemma B.1, we have

$$\mathbb{E}\|\hat{x}_n(t) - x_n(t)\|_2^2 \leq 2 \exp(Kt) \cdot \int_{s=0}^{t} \mathbb{E}[\|f^*(x_n(s), \pi_n(x_n(s))) - f_n(x_n(s), \pi_n(x_n(s)))\|_2^2] ds.$$

Therefore, we have

$$
\begin{aligned}
R_n &\le L \cdot \Bigg[ \int_{t=0}^{T} \sqrt{2\exp(Kt) \cdot \int_{s=0}^{t} \mathbb{E}[\|f^*(x_n(s), \pi_n(x_n(s))) - f_n(x_n(s), \pi_n(x_n(s)))\|_2^2] ds}\, dt \Bigg] \\
&\quad + \mathbb{E}\Bigg[ \int_{t=0}^{T} (b_n - b^*)(x(t), \pi_n(x(t))) dt \Bigg] \\
&\le LT \Bigg[ \sqrt{2\exp(KT) \cdot \int_{s=0}^{T} \mathbb{E}[\|f^*(x_n(s), \pi_n(x_n(s))) - f_n(x_n(s), \pi_n(x_n(s)))\|_2^2 ds}} \Bigg] + T\sqrt{B_n} \\
&= LT\sqrt{2T\exp(KT)}\sqrt{A_n} + T\sqrt{B_n}, \qquad (10)
\end{aligned}
$$

where

$$
\begin{aligned}
A_n &:= \mathbb{E}_{x_n,t} \|f^*(x_n(t), \pi_n(x_n(t))) - f_n(x_n(t), \pi_n(x_n(t)))\|_2^2, \\
B_n &:= \mathbb{E}_{x_n,t} |b_n(x_n(t), \pi_n(x_n(t))) - b^*(x_n(t), \pi_n(x_n(t)))|^2.
\end{aligned}
$$

Here, the second inequality holds due to the basic inequality $\mathbb{E}[x] \le \sqrt{\mathbb{E}[x^2]}$. Taking summation from $n = 1$ to $N$, we have

$$
\begin{aligned}
\sum_{n=1}^{N} R_n &\le \sum_{n=1}^{N} LT\sqrt{2T\exp(KT)}\sqrt{A_n} + T\sqrt{B_n} \\
&\le T\sqrt{N \sum_{n=1}^{N} B_n} + LT \cdot \sqrt{TN \cdot 2\exp(KT)} \sqrt{\sum_{n=1}^{N} A_n}, \qquad (11)
\end{aligned}
$$

Let $p_n$ denote the distribution of $z_n$, where $z_n = (x_n(t), \pi_n(x_n(t)))$ with the following joint distribution:

$$
t \sim \text{Unif}[0, T], x \sim X(t, \pi_n, q_n).
$$

With a slight abuse of notation, we use $f(z_n), b(z_n)$ to denote $f(x_n(t), \pi_n(x_n(t))), b(x_n(t), \pi_n(x_n(t)))$. Next we just need to make sure that for both $f$ and $b$, they satisfy the Eluder dimension. First, note that to train $b_n$ and $f_n$, we obtain it from the following one:

$$
\begin{aligned}
b_n \in \mathcal{R}_n, \mathcal{R}_n \leftarrow \Bigg\{ b : \sum_{z,y,r \in \mathcal{D}_n} (b(z) - r)^2 \le \min_{b' \in \mathcal{R}} \sum_{z,y,r \in \mathcal{D}_n} (b'(z) - r)^2 + \beta_{\mathcal{R}} \Bigg\} \\
f_n \in \mathcal{F}_n, \mathcal{F}_n \leftarrow \Bigg\{ f : \sum_{z,y,r \in \mathcal{D}_n} \|f(z) - y\|^2 \le \min_{f' \in \mathcal{F}} \sum_{z,y,r \in \mathcal{D}_n} \|f'(z) - y\|^2 + \beta_{\mathcal{F}} \Bigg\}.
\end{aligned}
$$

First, we have for any $b \in \mathcal{R}$ and $f \in \mathcal{F}$, by Lemma E.3 and taking union bound over $n$, for all $n$ we have with probability at least $1 - 2\delta \log N$:

$$
\sum_{i=1}^{n-1} \mathbb{E}_{z' \sim p_i} (b(z') - b^*(z'))^2 \le 8 \sum_{z \in \mathcal{D}_n} (b(z) - b^*(z))^2 + 4\log(4N/\delta), \qquad (12)
$$

$$
\sum_{i=1}^{n-1} \mathbb{E}_{z' \sim p_i} \|f(z') - f^*(z')\|^2 \le 8 \sum_{z \in \mathcal{D}_n} \|f(z) - f^*(z)\|^2 + 4\log(4N/\delta), \qquad (13)
$$

Then taking $b = b_n, f = f_n$ and using Lemma B.2, we have

$$
\begin{aligned}
\sum_{i=1}^{n-1} \mathbb{E}_{z' \sim p_i} (b_n(z') - b^*(z'))^2 &\le O\Bigg( \sum_{z,r \in \mathcal{D}_n} (b_n(z) - r)^2 - \inf_{b \in \mathcal{R}} \sum_{z,r \in \mathcal{D}_n} (b(z) - r)^2 + \log(4N/\delta) + \log(|\mathcal{R}_\epsilon|/\delta) \Bigg) \\
&\le O(\beta_{\mathcal{R}} + \log(4N/\delta) + \log(|\mathcal{R}_\epsilon|/\delta)), \qquad (14)
\end{aligned}
$$

and

$$\sum_{i=1}^{n-1}\mathbb{E}_{z'\sim p_i}\|f(z')-f^*(z')\|^2 \le O\Big(\sum_{z,y\in\mathcal{D}_n}\|f_n(z)-y\|^2-\inf_{f\in\mathcal{F}}\sum_{z,y\in\mathcal{D}_n}\|f_n(z)-y\|^2+\log(4N/\delta)+\log(|\mathcal{F}_\epsilon|/\delta)\Big)$$
$$\le O(\beta_{\mathcal{F}}+\log(4N/\delta)+G^2\log(|\mathcal{F}_\epsilon|/\delta)). \tag{15}$$

Therefore, taking $\phi_i(z)=(b_i(z)-b^*(z))^2$ and $\phi_i(z)=\|f_i(z)-f^*(z)\|^2$ separately, we can use Lemma B.3 for both cases and obtain that

$$\sum_{n=1}^{N}B_n \le O(d_{\mathcal{R}}(\beta_{\mathcal{R}}+\log(4N/\delta)+\log(|\mathcal{R}_\epsilon|/\delta))\log N)=O(d_{\mathcal{R}}\beta_{\mathcal{R}}\log N),$$
$$\sum_{n=1}^{N}A_n \le O(d_{\mathcal{F}}(\beta_{\mathcal{F}}+\log(4N/\delta)+G^2\log(|\mathcal{F}_\epsilon|/\delta))\log N)=O(d_{\mathcal{F}}\beta_{\mathcal{F}}\log N).$$

Substituting them into (11) finalizes our proof.

□

# C PROOF OF THEOREM 5.1

**Theorem C.1.** *With high probability, $f^*\in\mathcal{F}_n, b^*\in\mathcal{R}_n$. Meanwhile, the regret of Algorithm 2 is in the same order of Algorithm 1.*

*Proof.* Define the following confidence sets:

$$\hat{\mathcal{F}}_{n+1}\leftarrow\Big\{f:\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(f(x,u)-y)^2\le\min_{f'\in\mathcal{F}}\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(f'(x,u)-y)^2+5\beta_{\mathcal{F}}\Big\}.$$
$$\hat{\mathcal{R}}_{n+1}\leftarrow\Big\{b:\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(b_n(x,u)-r)^2\le\min_{b'\in\mathcal{R}}\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(b'(x,u)-r)^2+5\beta_{\mathcal{R}}\Big\}.$$

First, it is easy to see that Theorem B.4 still holds, therefore $b^*\in\mathcal{R}_n\subset\hat{\mathcal{R}}_n$ and $f^*\in\mathcal{F}_n\subset\hat{\mathcal{F}}_n$. Next, by our updating rule, we have for all $n$,

$$\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(f_n(x,u)-y)^2\le\min_{f'\in\mathcal{F}}\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(f'(x,u)-y)^2+5\beta_{\mathcal{F}}$$
$$\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(b_n(x,u)-r)^2\le\min_{b'\in\mathcal{R}}\sum_{x,u,y,r\in\mathcal{D}_{n+1}}(b'(x,u)-r)^2+5\beta_{\mathcal{R}}.$$

Therefore, we can follow the proof of Theorem B.5 by changing $\beta_{\mathcal{R}}$ and $\beta_{\mathcal{F}}$ with $5\beta_{\mathcal{R}}$ and $5\beta_{\mathcal{F}}$ in (14) and (15), the regret still holds. □

**Theorem C.2.** *The total switching number is $O(d_{\mathcal{F}}\log N+d_{\mathcal{R}}\log N)$.*

*Proof.* To begin with, note that for all $n$, we have

$$\sum_{x,u,y,r\in\mathcal{D}_n}(f_n(x,u)-y)^2\le\min_{f'\in\mathcal{F}}\sum_{x,u,y,r\in\mathcal{D}_n}(f'(x,u)-y)^2+5\beta_{\mathcal{F}}$$
$$\sum_{x,u,y,r\in\mathcal{D}_n}(b_n(x,u)-r)^2\le\min_{b'\in\mathcal{R}}\sum_{x,u,y,r\in\mathcal{D}_n}(b'(x,u)-r)^2+5\beta_{\mathcal{R}}.$$

Therefore, by Lemma B.2 and (12), (13), we have

$$\sum_{i=1}^{n-1}\mathbb{E}_{z'\sim p_i}(b_n(z')-b^*(z'))^2 \le 8\sum_{z\in\mathcal{D}_n}(b_n(z)-b^*(z))^2+4\log(4N/\delta),$$
$$\le 16\sum_{z\in\mathcal{D}_n}(b_n(z)-r)^2-\min_{b'\in\mathcal{R}}(b'(z)-r)^2+O(\beta_{\mathcal{R}})$$
$$\le O(\beta_{\mathcal{R}}),$$

where the second and third line hold due to the selection of $\beta_{\mathcal{R}}$. Similarily, we have

$$\sum_{i=1}^{n-1} \mathbb{E}_{z'\sim p_i} \|f_n(z') - f^*(z')\|^2 \le O(\beta_{\mathcal{F}}). \tag{16}$$

Next we derive the following bound. Consider $n_1 < n_2 < \cdots < n_l$ to be some $n \in [N]$ where $\mathcal{F}_n$ gets updated. Then at some $n_i$, we have

$$\sum_{x,u,y,r\in\mathcal{D}_{n_{i+1}}} (f_{n_i}(x,u) - y)^2 \ge \min_{f'\in\mathcal{F}} \sum_{x,u,y,r\in\mathcal{D}_{n_{i+1}}} (f'(x,u) - y)^2 + 5\beta_{\mathcal{F}} \ge \sum_{x,u,y,r\in\mathcal{D}_{n_i+1}} (f^*(x,u) - y)^2 + 4\beta_{\mathcal{F}}.$$

where the second inequality holds since $f^* \in \mathcal{F}_{n_i+1}$ due to Theorem C.1. Meanwhile, since $f_{n_i}$ is updated at $n_i$-th step, then $f_{n_i} \in \mathcal{F}_{n_i}$, which is

$$\sum_{x,u,y,r\in\mathcal{D}_{n_i}} (f_{n_i}(x,u) - y)^2 \le \min_{f'\in\mathcal{F}} \sum_{x,u,y,r\in\mathcal{D}_{n_i}} (f'(x,u) - y)^2 + \beta_{\mathcal{F}} \le \sum_{x,u,y,r\in\mathcal{D}_{n_i}} (f^*(x,u) - y)^2 + \beta_{\mathcal{F}}.$$

Therefore, we have

$$\sum_{x,u,y,r\in\mathcal{D}_{n_{i+1}}\setminus\mathcal{D}_{n_i}} [(f_{n_i}(x,u) - y)^2 - (f^*(x,u) - y)^2] \ge 3\beta_{\mathcal{F}}.$$

By Lemma B.2, we have

$$3\beta_{\mathcal{F}} \le \sum_{x,u,y,r\in\mathcal{D}_{n_{i+1}}\setminus\mathcal{D}_{n_i}} [(f_{n_i}(x,u) - y)^2 - (f^*(x,u) - y)^2] \le \sum_{x,u,y,r\in\mathcal{D}_{n_{i+1}}\setminus\mathcal{D}_{n_i}} \frac{3}{2}(f_{n_i}(x,u) - f^*(x,u))^2 + \beta_{\mathcal{F}}, \tag{17}$$

which suggests that

$$\sum_{n=n_i}^{n_{i+1}-1} (f_n(z_n) - f^*(z_n))^2 \ge \beta_{\mathcal{F}},$$

where we use the fact that $f_n = f_{n_i}$ when $n_i \le n < n_{i+1}$. Therefore, taking summation from $i = 1, \ldots, l$, we have

$$l \cdot \beta_{\mathcal{F}} \le \sum_{n=1}^{N} (f_n(z_n) - f^*(z_n))^2 = O\bigg(\sum_{n=1}^{N} \mathbb{E}_{z\sim p_n} (f_n(z) - f^*(z))^2 + \beta_{\mathcal{F}}\bigg) \le O(d_{\mathcal{F}}\beta_{\mathcal{F}} \log N),$$

where the first equality holds due to Lemma E.3, the second inequality holds due to (16) and Lemma B.3. It suggests the switching number $l = O(d_{\mathcal{F}} \log N)$. Similarily, the switching number of $\mathcal{R}_n$ is also bounded by $O(d_{\mathcal{R}} \log N)$. Combining them obtains the final result. □

# D PROOF OF THEOREM 5.6

The main idea of this proof originates from Xiong et al. [2023]. For the ease of presentation, we denote $p_{n,1}, \ldots, p_{n,m} = p_n$. We divide episodes $n = 1, \ldots, N/m$ into disjoint sets $E_j, j = 0, 1, \ldots, J$, where

$$\begin{aligned} j = 0, n \in E_0 : &\qquad \sum_{i=1}^{m} \mathbb{E}_{z_{n,i}\sim p_{n,i}} \|f_n(z_{n,i}) - f^*(z_{n,i})\|^2 < 100C_{\mathcal{T}} \cdot \beta_{\mathcal{F}}, \\ j \ge 1, n \in E_j : &\qquad 100C_{\mathcal{T}} \cdot 2^{j-1}\beta_{\mathcal{F}} \le \sum_{i=1}^{m} \mathbb{E}_{z_{n,i}\sim p_{n,i}} \|f_n(z_{n,i}) - f^*(z_{n,i})\|^2 < 100C_{\mathcal{T}} \cdot 2^j \beta_{\mathcal{F}}. \end{aligned} \tag{18}$$

Apparently, we have $J = O(\log N)$ since $f \leq 1$ and $m \leq N$. Meanwhile, note that by the definition of $f_n$, which is updated on $n$-th episode, then we have

$$\begin{aligned}
&\sum_{n'=1}^{n-1}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim p_{n',i}}\|f_n(z_{n',i}) - f^*(z_{n',i})\|^2 \\
&\leq C_{\mathcal{T}}\sum_{n'=1}^{n-1}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim\mathbb{P}_{\mathcal{T},\pi_{n'},q_{n'}}(\cdot|z_{n',i-1},\ldots,z_{n',1})}\Big[\|f_n(z_{n',i}) - f^*(z_{n',i})\|^2\Big] \\
&\leq C_{\mathcal{T}}\Big[4\sum_{n'=1}^{n-1}\sum_{i=1}^{m}\|f_n(z_{n',t_{n',i}}) - f^*(z_{n',t_{n',i}})\|^2 + \beta_{\mathcal{F}}\Big] \\
&\leq 100C_{\mathcal{T}}\beta_{\mathcal{F}}, \qquad (19)
\end{aligned}$$

where the first inequality due to the definition of $C_{\mathcal{T}}$, the second one holds due to Lemma E.2 and the selection of $\beta_{\mathcal{F}}$, the last one holds due to Lemma B.2 and the selection of $\beta_{\mathcal{F}}$. Combining the upper bound in (18) and (19), we have

$$\forall j \geq 0, \forall n \in E_j, \sum_{i=1}^{m}\mathbb{E}_{z_{n,i}\sim p_{n,i}}\|f_n(z_{n,i}) - f^*(z_{n,i})\|^2 + \sum_{n'=1}^{n-1}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim p_{n',i}}\|f_n(z_{n',i}) - f^*(z_{n',i})\|^2 \leq 200C_{\mathcal{T}}\cdot 2^j\beta_{\mathcal{F}},$$

therefore, by Lemma B.3, we have

$$\sum_{i=1}^{m}\mathbb{E}_{z_{n,i}\sim p_{n,i}}\|f_n(z_{n,i}) - f^*(z_{n,i})\|^2 + \sum_{n'=1}^{n-1}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim p_{n',i}}\|f_{n'}(z_{n',i}) - f^*(z_{n',i})\|^2 \leq O(d_{\mathcal{F}}C_{\mathcal{T}}\cdot 2^j\beta_{\mathcal{F}}\log N), \quad (20)$$

Next, for $j \geq 1$, we bound (20) from another direction. We have

$$\begin{aligned}
&\sum_{i=1}^{m}\mathbb{E}_{z_{n,i}\sim p_{n,i}}\|f_n(z_{n,i}) - f^*(z_{n,i})\|^2 + \sum_{n'=1}^{n-1}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim p_{n',i}}\|f_{n'}(z_{n',i}) - f^*(z_{n',i})\|^2 \\
&\geq \sum_{n'\in E_j, n'<n}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim p_{n',i}}\|f_{n'}(z_{n',i}) - f^*(z_{n',i})\|^2 \\
&\geq |\{n' \in E_j, n' < n\}|\cdot 100C_{\mathcal{T}}\cdot 2^{j-1}\beta_{\mathcal{F}}, \qquad (21)
\end{aligned}$$

where the second inequality holds due to the definition of $E_j$. Therefore, combining (20) and (21) and setting $n$ to be the max element in $E_j$, we have $|E_j| \leq O(d_{\mathcal{F}}\log N)$ for all $j \geq 1$. Similarily, for the reward function $b$, we define $F_j$ similar to $E_j$, we can also obtain that

$$\begin{aligned}
&j = 0, \forall n \in F_0: \quad \sum_{i=1}^{m}\mathbb{E}_{z_{n,i}\sim p_{n,i}}(b_n(z_{n,i}) - b^*(z_{n,i}))^2 + \sum_{n'=1}^{n-1}\sum_{i=1}^{m}\mathbb{E}_{z_{n',i}\sim p_{n',i}}(b_{n'}(z_{n',i}) - b^*(z_{n',i}))^2 \leq O(d_{\mathcal{R}}C_{\mathcal{T}}\beta_{\mathcal{R}}\log N) \\
&j \geq 1, \forall n \in F_j: \qquad |F_j| \leq O(d_{\mathcal{R}}\log N). \qquad (22)
\end{aligned}$$

Finally we bound the final regret. We look at the bound of the suboptimality gap $R_{n,i}$ from (10), where

$$R_{n,i} \leq LT\sqrt{2T\exp(KT)}\sqrt{A_{n,i}} + T\sqrt{B_{n,i}},\ A_{n,i} := \mathbb{E}_{z\sim p_{n,i}}\|f^*(z) - f_{n,i}(z)\|_2^2, B_{n,i} := \mathbb{E}_{z\sim p_{n,i}}|b_{n,i}(z) - b^*(z)|^2. \quad (23)$$

Then for the total regret, we have

$$\begin{aligned}
\sum_{i=1}^{m}\sum_{n=1}^{N/m}R_{n,i} &= \sum_{i=1}^{m}\Big(\sum_{n\in E_0\cap F_0}R_{n,i} + \sum_{n\notin E_0\cap F_0}R_{n,i}\Big) \\
&\leq \sum_{i=1}^{m}\sum_{n\in E_0\cap F_0}LT\sqrt{2T\exp(KT)}\sqrt{A_{n,i}} + T\sqrt{B_{n,i}} + (|E_1| + \cdots + |F_1| + \ldots)\cdot T \\
&\leq LT\sqrt{2T\exp(KT)}\sqrt{|E_0\cap F_0|\sum A_{n,i}} + T\sqrt{|E_0\cap F_0|\sum B_{n,i}} + mT\log N\cdot O(d_{\mathcal{F}} + d_{\mathcal{R}}) \\
&\leq O\big(LT\sqrt{2T\exp(KT)}\sqrt{Nd_{\mathcal{F}}C_{\mathcal{T}}\beta_{\mathcal{F}}\log N} + T\sqrt{Nd_{\mathcal{R}}C_{\mathcal{T}}\beta_{\mathcal{R}}\log N} + mT(d_{\mathcal{F}} + d_{\mathcal{R}})\log N\big),
\end{aligned}$$

where the first inequality holds due to (23) and the fact $R_{n,i} \leq T$, the second one holds due to Cauchy-Schwarz inequality, the last one holds due to conditions in (20), (22) and applying them to Lemma B.3. Therefore, our proof holds.

# E TECHNICAL LEMMAS

**Lemma E.1** (Gronwall's Inequality [Bellman, 1943]). *Let $u(t)$ be a non-negative, continuous function on the interval $[a, b]$. Suppose that*

$$u(t) \leq K + \int_a^t \gamma(s)u(s)\,ds$$

*for all $t \in [a, b]$, where $K$ is a non-negative constant and $\gamma(s)$ is a non-negative, continuous function on $[a, b]$. Then,*

$$u(t) \leq K \exp\left(\int_a^t \gamma(s)\,ds\right)$$

*for all $t \in [a, b]$.*

**Lemma E.2** (Zhang et al. 2021b). *Let $(\mathcal{F}_i)_{i\geq 0}$ be a filtration. Let $(X_i)_{i\geq 1}$ be a sequence of random variables such that $|X_i| \leq 1$ almost surely, and $X_i$ is $\mathcal{F}_i$-measurable. For every $\delta \in (0, 1)$, we have*

$$\Pr\left(\sum_{i=1}^n \mathbb{E}\left[X_i^2 \mid \mathcal{F}_{i-1}\right] \geq 8\sum_{i=1}^n X_i^2 + 4\ln\left(\frac{4}{\delta}\right)\right) \leq (\log(n) + 1)\delta.$$

**Lemma E.3** (Zhang et al. 2021b). *Let $(\mathcal{F}_i)_{i\geq 0}$ be a filtration. Let $(X_i)_{i\geq 1}$ be a sequence of random variables such that $|X_i| \leq 1$ almost surely, and $X_i$ is $\mathcal{F}_i$-measurable. For every $\delta \in (0, 1)$, we have*

$$\Pr\left(\sum_{i=1}^n X_i^2 \geq 8\sum_{i=1}^n \mathbb{E}\left[X_i^2 \mid \mathcal{F}_{i-1}\right] + 4\ln\left(\frac{4}{\delta}\right)\right) \leq (\log(n) + 1)\delta.$$

# F ADDITIONAL DETAILS OF EXPERIMENTS FOR DIFFUSION MODEL FINE-TUNING

In this section we introduce additional experiment details in Section 6.1.

## F.1 FROM THEORY TO PRACTICE

$\text{PURE}_{\text{SEIKO}}$ offers the first concrete realization of the general update schemes in Algorithms 2 and 3.

**How the Theoretical Insights Inform the Design of $\text{PURE}_{\text{SEIKO}}$** Intuitively, Theorem 5.1 suggests that by updating the policy and initial distribution less frequently, as prescribed by Algorithm 2, we can still maintain a high-probability confidence set for both the dynamics and the reward; Theorem 5.6 indicates that following Algorithm 3, performing multiple measurements within each episode–while keeping the *total* number of measurements unchanged–can yield comparable results to more rollout baselines.

As mentioned in the main context, SEIKO already adopts a low-switching schedule: training is divided into $\mathcal{K}$ batches with geometrically increasing sizes, $B_{i+1} = \eta_{\text{base}} B_i$ for $i \in [\mathcal{K}]$. Nevertheless, diffusion-model fine-tuning under SEIKO remains slow. Guided by Theorem 5.1 and Theorem 5.6, we insert extra mid-episode measurements to speed up data collection without enlarging the sample budget, producing Algorithm 4, the $\text{PURE}_{\text{SEIKO}}$ variant.

**How the Experiment Result Backup the Theoretical Results** Figure 1a demonstrates that adding in-trajectory measurements markedly shortens sampling time while achieving aesthetic scores comparable to the original SEIKO. This empirical behavior substantiates the prediction of Theorem 5.6.

## F.2 DETAILS OF SEIKO

Progress in SEIKO is primarily evaluated using a pre-trained aesthetic scorer, specifically the LAION Aesthetics Predictor V2 [Schuhmann, 2022]. Following Uehara et al. [2024], we fix the total number of scorer evaluations (i.e., measurements) at $N = 19200$. To address uncertainty in reward estimation, Uehara et al. [2024] propose two versions of the uncertainty oracle: *Bootstrap* (bootstrapped neural networks) and *UCB* (an uncertainty estimate derived from the network's last layer). We adopt the UCB variant, as it generally produces superior aesthetic scores. For the backbone diffusion model, SEIKO employs Stable Diffusion V1.5 [Rombach et al., 2022], which we also adopt as our pre-trained model. While Stable Diffusion V1.5 was originally trained with $1000$ discretized denoising steps, we follow SEIKO and reduce it to $50$ steps at inference time for improved sampling efficiency. For notational simplicity, we define the denoising time from $0$ (fully denoised) to $T$ (initial noise), inverting the conventional $T$-to-$0$ timeline.

### F.3 PURE$_{\text{SEIKO}}$ ALGORITHM

Building on SEIKO, we introduce a more flexible framework, PURE, which incorporates multiple in-trajectory measurements and allows control over the frequency of policy updates. We refer to this specialized version as PURE$_{\text{SEIKO}}$, whose pseudo-code is presented in Algorithm 4. In brief, at episode $n$, we begin from an initial state $x(t_0) \sim q_n$ and simulate the trajectory using the following update rule with time step $\Delta t$:

$$x(t_k) = x(t_{k-1}) + f_n(t_{k-1}, x(t_{k-1}))\,\Delta t + g^*(x(t_k))\,(\Delta w(t_k)),$$

where $\Delta w(t_k) \sim \mathcal{N}(0, (\Delta t)^2 \cdot I)$, $t_k = t_{k-1} + \Delta t$. The trajectory is then used to compute a Riemann sum over intermediate values of $b_n$—a learned reward function—to approximate the cumulative reward $R$. The dataset $\mathcal{D}_n$ for training $b_n$ is updated continuously across episodes, which ensures that $b_n$ converges toward the true reward function $b^*$ over time. This approach is commonly used for approximating integrals in diffusion models [Uehara et al., 2024].

To optimize $R = R(\pi, q, f, b)$ over the confidence sets, we construct upper confidence bounds (UCBs) for $f$ and $b$ based on their respective confidence sets $\mathcal{F}_n$ and $\mathcal{R}_n$. Then, we jointly optimize $R$ over $\pi$ and $q$ with UCBs of $f_n, b_n$ described above. In this framework, $B_1$ denotes the batch size in the first outer loop, while the hyperparameter $\eta_{\text{base}}$ determines the growth factor for the number of samples in subsequent outer loops, following the relation $B_{i+1} = \eta_{\text{base}} \cdot B_i$.

**Algorithm 4** **P**olicy **U**pdate and **R**olling **E**fficient CTRL for Optimi**S**tic fin**E**-tuning of d**I**ffusion with **K**L c**O**nstraint (PURE$_{\text{SEIKO}}$)

**Require:** Total measurement number $N$, initial distribution class $q \in \mathcal{Q}$, pre-trained drift class $f^{\text{pre}} \in \mathcal{F}$, diffusion term $g^*$, ground-truth reward $r \in \mathcal{R}$, reward approximation $\hat{r}$, episode length $T$, sampler $\mathcal{T}$, diffusion hyperparameter $\alpha, \{\beta_n\} \in \mathbb{R}^+$, counter $\kappa$, measurement frequency $m$, initial batch size $B_1 \in \mathbb{Z}^+$, hyperparameter $\eta_{\text{base}} \in \mathbb{R}^+$.

1: Initialize $f_1 = f^{\text{pre}}, q_1 = q, \kappa = 0$.
2: **for** episode $n = 1, \ldots, \lfloor N/m \rfloor$ **do**
3: Sample $t_{n,1}, \cdots, t_{n,m} \sim \mathcal{T}, t_{n,0} = 0$
4: Execute $dx(t) = f_{n-1}(t, x(t))dt + g^*(x(t))dw(t), x(0) \sim q_{n-1}$, receive feedback $y_{n,i} = r(x(t_{n,i})) + \epsilon$
5: Update $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup (\{x(t_{n,i}), y_{n,i}\}_{i=1}^m)$.
6: Set $\hat{r}_{n+1} \leftarrow \hat{r}_n$, $f_{n+1} \leftarrow f_n$
`// If collected enough samples, update the reward and diffusion once.`
7: **if** $\frac{\eta_{\text{base}}^{\kappa} \cdot B_1}{m} \leq n$ **then**
8: Train $\hat{r}_{n+1}$ on $\mathcal{D}_{n+1}$, and update $f_{n+1}$, $q_{n+1}$ by solving

$$f_{n+1}, q_{n+1} = \underset{f \in \mathcal{F}, q \in \mathcal{Q}}{\arg\max}\ \mathbb{E}_{\mathbb{P}^{f,q}}[\hat{r}(x(T))] \ - \ \alpha\,\mathrm{KL}(\mathbb{P}^{f,q} | \mathbb{P}^{f_1,q_1}) \ - \ \beta_n\,\mathrm{KL}(\mathbb{P}^{f,q} \| \mathbb{P}^{f_n,q_n}),$$

using the DDIM optimizer [Song et al., 2020a], where $\mathbb{P}^{f,q} \in \Delta(\mathcal{X})$ refers to the marginal distribution at $T$. Then set $\kappa \leftarrow \kappa + 1$.
9: **end for**
10: **Output:** $f_{\lfloor \frac{N}{m} \rfloor + 1}, q_{\lfloor \frac{N}{m} \rfloor + 1}$

### F.4 ADDITIONAL EXPERIMENT RESULTS

Figure 3 presents a qualitative comparison between samples generated by the diffusion model fine-tuned with SEIKO and our proposed PURE$_{\text{SEIKO}}$. Notably, PURE$_{\text{SEIKO}}$ achieves a comparable output image quality to SEIKO while requiring fewer computational resources.

### F.5 PURE$_{\text{SEIKO}}$ EXPERIMENT PARAMETERS

**Prompts** For a fair comparison with the SEIKO algorithm, we follow the prompt settings from Uehara et al. [2024]'s image task for both training and evaluation. Specifically, the training phase utilizes prompts from a predefined list of 50 animals [Black et al., 2023, Prabhudesai et al., 2023], while the evaluation phase employs the following unseen animal prompts: snail, hippopotamus, cheetah, crocodile, lobster, and octopus.

**Hyperparameters** Table 1 summarizes the key hyperparameters for fine-tuning. We use ADAM [Kingma, 2014] as the optimizer.

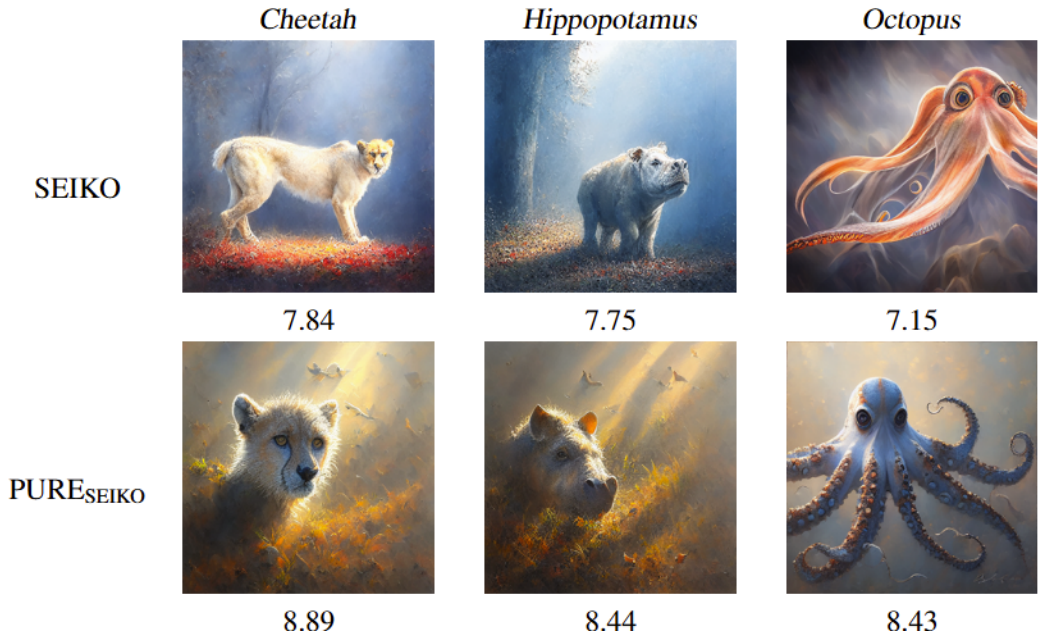


Figure 3: Qualitative comparison between SEIKO and our PURE$_{\text{SEIKO}}$ approach, with aesthetic scores listed below each image.

Table 1: Important hyperparameters for fine-tuning.

| **Method** | **Type** | |
|---|---|---|
| SEIKO | Batch size | 128 |
| | KL parameter $\beta$ | 0.01 |
| | UCB parameter $C_1$ | 0.002 |
| | Sampling to neural SDE | Euler |
| | Step size (fine-tuning) | 50 |
| | Epochs (fine-tuning) | 100 |
| PURE$_{\text{SEIKO}}$ | $\lambda$ (temperature parameter in 6.1) | 6 |
| | $N$ (total measurement number) | 19200 |
| | $m$ (measurement frequency) | 4 |
| | $B_1$ (number of samples in the first outer loop) | 1280 |
| | $\eta_{\text{base}}$ (growth factor for subsequent outer loop) | 2 |

# G CONTINUOUS-TIME CONTROL EXPERIMENTS

In this section we introduce additional experiment details about continuous-time control tasks.

## G.1 FROM THEORY TO PRACTICE

PURE$_{\text{ENODE}}$ offers the second realization of the general update schemes in Algorithms 2 and 3.

**How the Theoretical Insights Inform the Design of PURE$_{\text{ENODE}}$** Building on ENODE, we introduce a more flexible framework, PURE, which enables control over the frequency of policy updates. A specialized instance of this framework, referred to as PURE$_{\text{ENODE}}$, is detailed in Algorithm 5. As noted in the main text, ENODE already adopts a low-rollout strategy. In PURE$_{\text{ENODE}}$, we further incorporate a batch-style update scheme inspired by Algorithm 2. Specifically, following a scheme similar to SEIKO, the batch size $B_i$ doubles at each step according to $B_{i+1} = 2B_i$, while keeping the total sample budget $N$ fixed.

**How the Experiment Result Backup the Theoretical Results** Figure 2 demonstrates that reducing the number of policy updates can significantly shorten training time while maintaining comparable rewards. This empirical observation supports the insight of Theorem 5.1.

### G.2 ADDITIONAL DETAILS OF EXPERIMENTS FOR CONTINUOUS-TIME CONTROL TASKS

We conduct experiments on the Acrobot, Pendulum, and Cart Pole tasks using the OpenAI Gym simulator [Brockman, 2016]. In all tasks, the system begins in a hanging-down state, and the objective is to swing up and stabilize the pole(s) in an upright position [Yildiz et al., 2021]. We put related parameters in Table 2.

Table 2: Environment specifications

| **Environment** | $c_p$ | $c_a$ | $\alpha_{\max}$ | $\mathbf{s}^{\text{box}}$ | $\mathbf{s}^{\text{goal}}$ |
|---|---|---|---|---|---|
| **Acrobot** | 1e-4 | 1e-2 | 4 | $[0.1, 0.1, 0.1, 0.1]$ | $[0, 2\ell]$ |
| **Pendulum** | 1e-2 | 1e-2 | 2 | $[\pi, 3]$ | $[0, \ell]$ |
| **Cart Pole** | 1e-2 | 1e-2 | 3 | $[0.05, 0.05, 0.05, 0.05]$ | $[0, 0, \ell]$ |

**Acrobot** The Acrobot system consists of two links connected in series, forming a chain with one end fixed. The joint between the two links is actuated, and the goal is to apply torques to this joint to swing the free end above a target height, starting from the initial hanging-down state. We use the fully actuated version of the Acrobot environment, as no method has successfully solved the underactuated balancing problem, consistent with Zhong and Leonard [2020]. The control space is discrete and deterministic, representing the torque applied to the actuated joint. The state space consists of the two rotational joint angles and their angular velocities.

**Pendulum** The inverted pendulum swing-up problem is a fundamental challenge in control theory. The system consists of a pendulum attached at one end to a fixed pivot, with the other end free to move. Starting from a hanging-down position, the goal is to apply torque to swing the pendulum into an upright position, aligning its center of gravity directly above the pivot. The control space represents the torque applied to the free end, while the state space includes the pendulum's x-y coordinates and angular velocity.

**Cart Pole** The Cart Pole system comprises a pole attached via an unactuated joint to a cart that moves along a frictionless track. Initially, the pole is in an upright position, and the objective is to maintain balance by applying forces to the cart in either the left or right direction. The control space determines the direction of the fixed force applied to the cart. The state space includes the cart's position and velocity, as well as the pole's angle and angular velocity.

**Initial State** In all environments, the initial position $\mathbf{q}(0)$ is uniformly distributed as:

$$\mathbf{q}(0) \sim \text{Unif}\left(-\mathbf{s}^{\text{box}}, \mathbf{s}^{\text{box}}\right),$$

where $\mathbf{s}^{\text{box}}$ is the position parameter.

**Reward Functions** For all three tasks, we denote the state by $x = (\mathbf{q}, \mathbf{p})$, where $\mathbf{q}$ denotes the position and $\mathbf{p}$ denotes the velocity (momentum). Given a state $x = (\mathbf{q}, \mathbf{p})$ and a control unit $u$, the differentiable reward function is defined as:

$$b(x, u) = \exp\left(-\|\mathbf{q} - \mathbf{s}^{\text{goal}}\|_2^2 - c_p\|\mathbf{p}\|_2^2 - c_a\|u\|_2^2\right),$$

where $\mathbf{s}^{\text{goal}}$ denotes the goal position, $c_p$ and $c_a$ denote environment-specific constants. The exponential formulation ensures that the reward remains within $[0, 1]$, while penalizing deviations from the target state and excessive control effort. The environment-specific parameters are set following the exact configurations in Yildiz et al. [2021].

**Baseline** We highlight several unique components of $\text{PURE}_{\text{ENODE}}$. First, ENODE trains dynamics following an evidence lower bound (ELBO) [Blei et al., 2017] setup, which aims to minimize the negative log-likelihood function between the true state and the imagined state generated by the dynamics function. This process can be regarded as an approximation of our introduced measurement oracle. $\text{PURE}_{\text{ENODE}}$ employs a sampler that generates time steps consisting of several mini-batches, where each mini-batch comprises consecutive time steps with a randomly selected initial time step. To train the optimal policy, ENODE adopts the standard actor-critic framework based on the learned dynamics. Further details can be found in Yildiz et al. [2021].

**Neural Network Architectures** We adopt the same neural network architecture as described in Yildiz et al. [2021]. The dynamics, actor, and critic functions are approximated using multi-layer perceptrons (MLPs). The same neural network architectures were employed across all methods and environments, as detailed below:

- **Dynamics:** The dynamics function is modeled with three hidden layers, each containing 200 neurons, utilizing Exponential Linear Unit (ELU) activations. Experimental observations suggest that ELU activations enhance extrapolation on test sequences.

- **Actor:** The actor network consists of two hidden layers, each with 200 neurons, using ReLU activations. This design is motivated by the observation that optimal policies can often be approximated as a collection of piecewise linear functions. The final output of the network is passed through a tanh activation function and scaled by $\alpha_{\max}$.
- **Critic:** The critic network also consists of two hidden layers, each with 200 neurons, but employs tanh activations. Since state-value functions must exhibit smoothness, tanh activations are more suitable compared to other activation functions. Empirical results indicate that critic networks with ReLU activations tend to overfit to training data, leading to instability and degraded performance when extrapolating beyond the training distribution.

**Algorithm 5** **P**olicy **U**pdate and **R**olling **E**fficient CTRL with **E**nsemble **N**eural **ODE**s ($\text{PURE}_{\text{ENODE}}$)

**Require:** Total measurement number $N$, measurement frequency $m$, episode length $T = 50$, sub-episode length $T' = 5$, true reward $r \in \mathcal{R}$, dynamic class $\mathcal{F}$, policy class $\Pi$, initial batch size $B_1$, number of initial trajectories to collect $\eta_{\text{init}} \in \mathbb{Z}^+$, counter $\kappa$, sampler $\mathcal{T}$, mini-batch size $N_d = 5$, time grid $\Delta t \in \mathbb{R}^+$, hyperparameter $\eta_{\text{base}} \in \mathbb{R}^+$

Initialize dynamic $f$, policy $\pi$ as untrained Neural Network. Initialize an initial measurement dataset $\mathcal{D}_0 = \{\{x(t_{i,j}), u(t_{i,j})\}_{j=1}^{m}\}_{i=1}^{\eta_{\text{init}}}$, collecting $\eta_{\text{init}}$ trajectories with smooth random policies defined in Yildiz et al. [2021]; $\kappa = 0$
**for** episode $n = \eta_{\text{init}} + 1, \ldots, \lfloor N/m \rfloor$ **do**
  Run sampler $\mathcal{T}$ and receive $t_{n,i}, i \in [m = N_d T'/\Delta t]$, where $\{t_{n,i}\}$ consists of $N_d$ number of independent mini-batches, each mini-batch consists of $t^0, t^0 + \Delta t, \ldots, t^0 + T'$ consequent time steps with grid $\Delta t$.
  Execute policy $\pi$ and observe at $t_{n,i}$. Update dataset $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup \{x(t_{n,i}), u(t_{n,i})\}_{i=1}^{m}$
  `// If collected enough samples, update the dynamic and actor-critic once.`
  **if** $\frac{\eta_{\text{base}}^{\kappa} \cdot B_1}{m} \leq n - \eta_{\text{init}}$ **then**
    Train $f$ by using the ELBO on $\mathcal{D}_{n+1}$
    Train $\pi$ following the actor-critic schedule based on the dynamic $f$ following Algorithm 1 in Yildiz et al. [2021]
    Set $\kappa \leftarrow \kappa + 1$.
  **end if**
**end for**
**Output:** Policy $\pi$

### G.3 CONTINUOUS-TIME CONTROL EXPERIMENT DETAILS

**Additional Details** We include the batch size information in Table 3. In all experiments, we use DOPRI5 (RK45) as the adaptive ODE solver, as suggested by Yildiz et al. [2021]. We use the ADAM optimizer [Kingma, 2014] to train all model components, with the learning rate varying by environment.

Table 3: Data and Policy Updates

| **Environment** | Model | $N/m$ | $\eta_{\text{init}}$ | Number of Batches | Batch Sizes |
|---|---|---|---|---|---|
| **Acrobot** | ENODE | 87 | 7 | 20 | $[4, \ldots, 4]$ with length 20 |
| | $\text{PURE}_{\text{ENODE}}$ | | | 6 | $[2, 4, 8, 16, 18, 32]$ |
| **Pendulum** | ENODE | 9 | 3 | 6 | $[1, 1, 1, 1, 1, 1]$ |
| | $\text{PURE}_{\text{ENODE}}$ | | | 3 | $[1, 2, 3]$ |
| **Cart Pole** | ENODE | 80 | 5 | 25 | $[3, \ldots, 3]$ with length 25 |
| | $\text{PURE}_{\text{ENODE}}$ | | | 6 | $[2, 4, 8, 13, 16, 32]$ |

### G.4 ABLATION STUDY

For simplicity of presentation, we use $N_{\text{pu}}$ to denote the number of batches where the dynamics and policy are only updated at the beginning of each batch. For all experiments in the ablation study of continuous-time control, we select the Acrobot environment, as it requires a moderate amount of time to reach success.

#### G.4.1 Varying Number of Batches $N_{\text{pu}}$

First, we investigate the impact of different batch update scheduling strategies, namely, the policy update times $N_{\text{pu}}$. In addition to the doubling strategy $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=6}$ introduced in Section 6.2, we implement two alternative variations of $\text{PURE}_{\text{ENODE}}$: **(a)** $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=10}$, which maintains a constant batch size $B_i$ at each step (equaling strategy) but reduces the policy update frequency to half of the original ENODE, and **(b)** a more aggressive tripling approach $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=4}$, where the batch size $B_i$ triples at each step, following the rule $B_{i+1} = 3B_i$. In all cases, we ensured that the total sample budget $N$ remained consistent, and the total episode number is $N/m = 87$, with each data trajectory containing $m = 250$ observations to align with the main experimental setup. Further details are provided in Table 4.

The results, shown in Figure 4a, indicate that overall runtime decreases as the number of policy updates $N_{\text{pu}}$ is reduced. However, the batch update scheduling strategy plays a crucial role in determining the efficiency of policy learning. For instance, although $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=10}$ has a larger $N_{\text{pu}}$ than $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=6}$ and might be expected to achieve higher average rewards, it frequently failed to meet the success criteria after exhausting the batch update scheduler in several experiments. We attribute this phenomenon to the importance of ensuring that high-quality samples dominate the dataset in the later stages of policy updates. If initial low-quality samples remain prevalent, the agent may struggle to fully leverage the high-quality samples for effective learning. Conversely, an overly aggressive approach with very few policy updates, as in $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=4}$, leads to difficulties in processing the large influx of new trajectories in later stages of policy updates, resulting in unstable final performance.

Table 4: Ablation Study: Batch Update Scheduling Strategy $N_{\text{pu}}$

| Model | Strategy | $N_{\text{pu}}$ | $N_{\text{inc}}$ |
|---|---|---|---|
| $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=4}$ | Tripling | 4 | $[2, 6, 18, 54]$ |
| $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=6}$ | Doubling | 6 | $[2, 4, 8, 16, 18, 32]$ |
| $\text{PURE}_{\text{ENODE}}^{N_{\text{pu}}=10}$ | Equaling | 10 | $[4, \ldots, 4]$ with length 10 |

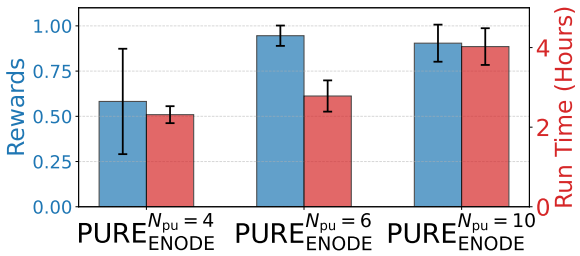
#### G.4.2 Varying Number of Measurements $m$

Similar to the ablation study in fine-tuning diffusion models (Section 6.1), we analyze the impact of the number of measurements ($m$) on effective policy learning. In the Acrobot environment, the total number of measurements for one policy update is given by $m = N_d \times \frac{t_s}{\Delta t} = 5 \times \frac{5}{0.1} = 250$. In addition to $m = 250$, we evaluate alternative settings with $m = 125, 500$, and $1000$ for trajectories included in $\mathcal{D}$. Details of the modified parameters for each setting are provided in Table 5.

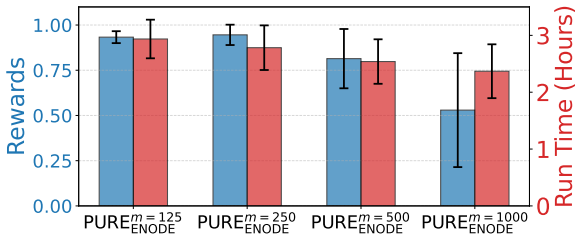
The results, shown in Figure 4b, indicate that increasing $m$ can slightly reduce total training time by decreasing the number of required trajectories. However, unlike in diffusion model experiments, where data trajectory generation is the primary computational bottleneck, the bottleneck in continuous-time control experiments lies in the policy iteration process. As a result, the reduction in training time is relatively small compared to our SEIKO experiments. Moreover, excessively large values of $m$ can negatively impact model performance, yielding low final rewards when $m = 500$ and $1000$. This highlights the necessity of selecting an optimal $m$ to balance solving continuous-time control problems effectively while maintaining training efficiency. These findings align with our claim in Theorem 5.6, reinforcing the importance of appropriately choosing $m$ to achieve the best trade-off.

Table 5: Ablation Study: Number of Measurements $m$

| Model | $m$ | $N_{\text{inc}}$ |
|---|---|---|
| $\text{PURE}_{\text{ENODE}}^{m=125}$ | 125 | $[4, 8, 16, 32, 36, 64]$ |
| $\text{PURE}_{\text{ENODE}}^{m=250}$ | 250 | $[2, 4, 8, 16, 18, 32]$ |
| $\text{PURE}_{\text{ENODE}}^{m=500}$ | 500 | $[1, 2, 4, 8, 9, 16]$ |
| $\text{PURE}_{\text{ENODE}}^{m=1000}$ | 1000 | $[1, 1, 2, 4, 4, 8]$ |




(a) PURE$_{\text{ENODE}}$ with varying $N_{\text{pu}}$




(b) PURE$_{\text{ENODE}}$ with varying $m$

Figure 4: Summary of the ablation studies for continuous-time control in the Acrobot environment. Figures 4a and 4b analyze the impact of the number of policy updates $N_{\text{pu}}$ and the number of observations $m$ on the final rewards, respectively, considering either exhausting the scheduler or achieving success, whichever occurs first.