More Efficient Randomized Exploration for Reinforcement Learning via Approximate Sampling

Haque Ishfaq[*]
haque.ishfaq@mail.mcgill.ca
Mila, McGill University

Yixin Tan[*]
yixin.tan@duke.edu
Duke University

Yu Yang
Duke University

Qingfeng Lan
University of Alberta

Jianfeng Lu
Duke University

A. Rupam Mahmood
Amii, University of Alberta

Doina Precup
Mila, McGill University

Pan Xu
Duke University

Abstract

Thompson sampling (TS) is one of the most popular exploration techniques in reinforcement learning (RL). However, most TS algorithms with theoretical guarantees are difficult to implement and not generalizable to Deep RL. While the emerging approximate sampling-based exploration schemes are promising, most existing algorithms are specific to linear Markov Decision Processes (MDP) with suboptimal regret bounds, or only use the most basic samplers such as Langevin Monte Carlo. In this work, we propose an algorithmic framework that incorporates different approximate sampling methods with the recently proposed Feel-Good Thompson Sampling (FGTS) approach (Zhang, 2022; Dann et al., 2021), which was previously known to be computationally intractable in general. When applied to linear MDPs, our regret analysis yields the best known dependency of regret on dimensionality, surpassing existing randomized algorithms. Additionally, we provide explicit sampling complexity for each employed sampler. Empirically, we show that in tasks where deep exploration is necessary, our proposed algorithms that combine FGTS and approximate sampling perform significantly better compared to other strong baselines. On several challenging games from the Atari 57 suite, our algorithms achieve performance that is either better than or on par with other strong baselines from the deep RL literature.

1 Introduction

A fundamental problem in reinforcement learning (RL) is balancing exploration-exploitation tradeoff. One effective mechanism for addressing this challenge is Thompson Sampling (TS) (Thompson, 1933; Strens, 2000; Osband et al., 2016b), which gained popularity due to its simplicity and strong empirical performance (Osband et al., 2016a; 2018; Ishfaq et al., 2024). While there have been numerous works on TS in both RL theory (Osband et al., 2013; Russo, 2019; Zanette et al., 2020a; Ishfaq et al., 2021; Xiong et al., 2022) and deep RL literature (Osband et al., 2016a; 2018; Fortunato et al., 2018; Plappert et al., 2018; Ishfaq et al., 2024; Li et al., 2024), there remains a substantial gap between algorithms that excel in theoretical properties and those that demonstrate strong empirical performance. This disparity highlights the need for a unified framework that provides the ultimate unification of theory and practice for Thompson sampling.

In particular, heuristic RL algorithms that are motivated by TS (Osband et al., 2016a; 2018; Fortunato et al., 2018) have shown great empirical potential while often lacking any theoretical guarantee. Existing RL theory works on TS suffer from sub-optimal dimension dependency compared to its upper confidence bound (UCB) counterparts (Jin et al., 2020; Zanette et al., 2020a; Ishfaq et al., 2021;

[*]Equal contribution

2024). Recently proposed Feel-Good Thompson sampling (FGTS) (Zhang, 2022; Dann et al., 2021) bypasses this issue by incorporating an optimistic prior term in the posterior distribution of Q function. However, these works fail to provide any computationally tractable sampling procedure from this posterior distribution.

Recently there has been some works that use Langevin Monte Carlo (LMC) (Dwaracherla & Van Roy, 2020; Xu et al., 2022; Ishfaq et al., 2024; Hsu et al., 2024) to implement TS which are both provably efficient and practical. However, these works lack generality by confining only to LMC and it remains unclear whether many other advanced approximate sampling methods are compatible with this algorithmic scheme for implementing TS. Moreover, the theoretical analyses of these works are limited to linear MDPs (Jin et al., 2020) while having sub-optimal regret bound. Thus, it is unclear how the sampling error of LMC would affect the theoretical regret guarantee under more general structural assumptions. This shows a clear divergence between the RL theory and the deep RL literature when it comes to TS. To this end, we aim to design an approximate TS based RL algorithm that is generalizable and flexible enough to use different approximate samplers while achieving optimal dependency in the regret bound.

With this aim, we propose several FGTS class of algorithms that incorporates different approximate samplers from the Markov Chain Monte Carlo (MCMC) literature. Unlike previous works that assume exact posterior sampling (e.g., Zhang (2022); Dann et al. (2021); Agarwal & Zhang (2022a;b)) by assuming access to unrealistic sampling oracle, we propose practically implementable approximate posterior sampling scheme under FGTS framework using different approximate samplers.

1.1 Key Contributions

We highlight the main contributions of the paper below:

- We present a class of practical and efficient TS based online RL algorithms that prioritizes easy implementation and computational scalability. Concretely, we present practically implementable FGTS style algorithms that are based on approximate samplers from the MCMC literature. Our proposed algorithm allows flexible usage of different approximate samplers such as Langevin Monte Carlo (LMC) (Durmus et al., 2019) or Underdamped Langevin Monte Carlo (ULMC) (Chen et al., 2014; Cheng et al., 2018) and is easy to implement compared to other state-of-the-art exploration focused deep RL algorithms.
- Our main theoretical result provides regret bound under general Markov decision processes and value function classes. Our general analytical framework decomposes the regret bound into two components: 1) the idealistic regret bound assuming exact TS (as addressed in many previous works such as Agarwal & Zhang (2022b;a)), and 2) the additional term introduced by using approximate samplers. This generalizable and fine-grained analysis allows us to analyze the impact of sampling error for any RL setting with an existing exact-sampling regret bound and known convergence rates for the approximate samplers.
- When applied to linear MDPs (Jin et al., 2020), our proposed algorithm achieves a regret bound of $\tilde{O}(dH^{\frac{3}{2}}\sqrt{T})$, where d is the dimension of the feature mapping, H is the planning horizon, and T is the total number of steps. This regret bound has the best known dimension dependency for any randomized and UCB based algorithms.
- We provide extensive experiments on both N-chain environments (Osband et al., 2016a) and challenging Atari games (Bellemare et al., 2013) that require deep exploration. Our experiments indicate our proposed algorithms perform similarly or better than state-of-the-art exploration algorithms from the deep RL literature.

1.2 Additional Related Work

Randomized least-squares value iteration (RLSVI) based algorithms induce deep exploration by injecting judiciously tuned random noise into the value function (Russo, 2019; Zanette et al., 2020a; Ishfaq et al., 2021; Xiong et al., 2022). Osband et al. (2016a; 2018) propose deep RL variant of RLSVI wherein they train an ensemble of randomly initialized neural networks and view them as approximate posterior samples of Q functions. Another deep RL variant of RLSVI is Noisy-Net (Fortunato et al., 2018) that directly injects noise to neural network parameters during the training phase. More recently, Dwaracherla & Van Roy (2020); Ishfaq et al. (2024) propose using LMC to perform approximate TS. While Ishfaq et al. (2024) provides regret bound for their proposed LMC-LSVI algorithm under linear MDP, their regret bound, like other existing randomized exploration algorithms, has sub-optimal dependency on the dimension of the linear MDP. Hsu et al. (2024) further extends LMC-LSVI to the cooperative multi-agent reinforcement learning setting. Dann et al. (2021) proposes conditional posterior sampling based on FGTS (Zhang, 2022) with a regret bound that has optimal dependency on the dimension in linear MDPs, but their algorithm is intractable due to the need to access to some unknown sampler oracle.

2 Preliminary

In this paper, we consider an episodic discrete-time Markov decision process (MDP), denoted by $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$, with $\mathcal{S}$ being the state space, and $\mathcal{A}$ the action space. The MDP is non-stationary across H different stages, which form an episode. H is also often referred to as the episode length. $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ is the collection of the state transition probability distributions, $\mathbb{P}_h(\cdot \mid x, a)$ denotes the transition kernel at stage $h \in [H]$. Let $r = \{r_h\}_{h=1}^H$ be the collection of reward functions, which we assume to be deterministic and bounded in $[0, 1]$ for the simplicity of the presentation.

We define a policy π in this MDP as a collection of H functions $\{\pi_h : \mathcal{S} \to \mathcal{A}\}_{h\in[H]}$, where $\pi_h(x)$ means the action at state x given by policy π at stage h of the episode.

At stage $h \in [H]$, we define the value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ as the total expected rewards collected by the agent if it starts at state $x_h = x$ and follows policy π onwards, i.e., $V_h^\pi(x) = \mathbb{E}_{\pi,\mathbb{P}}\big[\sum_{h'=h}^H r_{h'}(x_{h'}, a_{h'})\big|x_h = x\big]$.

We also define the action-value function (or Q function) $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as the total expected rewards by the agent if it starts at state $x_h = x$ and action $a_h = a$ and follows policy π onwards, i.e., $Q_h^\pi(x, a) = \mathbb{E}_{\pi,\mathbb{P}}\big[\sum_{h'=h}^H r_{h'}(x_{h'}, a_{h'})\big|x_h = x, a_h = a\big]$.

To simplify the notation, we denote operator $[\mathbb{P}_h V_{h+1}^\pi](x, a) = \mathbb{E}_{x'\sim\mathbb{P}_h(\cdot|x,a)} V_{h+1}^\pi(x')$. Thus, we write the Bellman equation associated with a policy π as

$$Q_h^\pi(x, a) = (r_h + \mathbb{P}_h V_{h+1}^\pi)(x, a), \qquad V_h^\pi(x) = Q_h^\pi(x, \pi_h(x)), \qquad V_{H+1}^\pi(x) = 0. \tag{2.1}$$

We denote the optimal policy as $\pi^* = \{\pi_h^*\}_{h=1}^H$, which is the collection of the optimal policies at each stage h. We further denote $V_h^*(x) = V_h^{\pi^*}(x)$ and $Q_h^*(x, a) = Q_h^{\pi^*}(x, a)$. It can be shown that π^* is a deterministic policy and it satisfies $Q_h^{\pi^*}(s, a) = \max_\pi Q^\pi(s, a)$ and $V_h^{\pi^*}(s) = \max_\pi V_h^\pi(s)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ (Bertsekas, 2019).

We denote the Bellman optimality operator by $\mathcal{T}_h^*$ that maps any function Q over $\mathcal{S} \times \mathcal{A}$ to

$$[\mathcal{T}_h^* Q](x, a) = r_h(x, a) + \mathbb{E}_{x'\sim\mathbb{P}_h(\cdot|x,a)}\big[\max_{a'\in\mathcal{A}} Q_{h+1}(x', a')\big]. \tag{2.2}$$

Note that the optimal Q function satisfies $\mathcal{T}_h^* Q_{h+1}^* = Q_h^*$, for all $h \in [H]$.

The agent follows the following iterative interaction protocol. At the beginning of each episode $k \in [K]$, an adversary picks an initial state x_1^k for stage 1, and the agent executes a policy π^k and updates the policy in the next stage according to the received rewards. We measure the suboptimality of the agent by the total regret defined as

$$\text{Regret}(K) = \sum_{k=1}^K \text{REG}_k := \sum_{k=1}^K \big[V_1^*(x_1^k) - V_1^{\pi^k}(x_1^k)\big].$$

2.1 Notations

We use $a = O(b)$ to indicate that $a \le Cb$ for a universal constant $C > 0$. Also, we write $a = \Theta(b)$ if there are universal constants $c' > c > 0$ such that $cb \le a \le c'b$, and the notation $\tilde{O}(\cdot)$ and $\tilde{\Theta}(\cdot)$ mean

Algorithm 1 Least-Squares Value Iteration with Approximate Sampling Exploration (LSVI-ASE)

```
1: Input: feel-good prior weight $\eta$, step sizes $\{\tau_{k,h} > 0\}_{k,h\geq 1}$, temperature $\beta$, friction coefficient $\gamma$,
   loss function $L_k(w)$.
2: Initialize $w_h^{1,0} = \mathbf{0}$ for $h \in [H]$, $J_0 = 0$.
3: for episode $k = 1, 2, \ldots, K$ do
4:     Receive the initial state $s_1^k$.
5:     for step $h = H, H-1, \ldots, 1$ do
6:         $w_h^{k,0} = w_h^{k-1,J_{k-1}}$
7:         for $j = 1, \ldots, J_k$ do
8:             Generate $w_h^{k,j}$ via a sampler in Section 3.2
9:         end for
10:        $Q_h^k(\cdot,\cdot) \leftarrow \min\{Q(w_h^{k,J_k}; \phi(\cdot,\cdot)), H-h+1\}^+$
11:        $V_h^k(\cdot) \leftarrow \max_{a\in\mathcal{A}} Q_h^k(\cdot, a)$.
12:    end for
13:    for step $h = 1, 2, \ldots, H$ do
14:        Take action $a_h^k \leftarrow \mathrm{argmax}_{a\in\mathcal{A}} Q_h^k(s_h^k, a)$.
15:        Observe reward $r_h^k(s_h^k, a_h^k)$, get next state $s_{h+1}^k$.
16:    end for
17: end for
```

they hide polylog factors in the parameters. For two probability distributions p and q on the same probability space, we denote their total variation (TV) distance by $TV(p, q)$. For $T : \mathbb{R}^d \to \mathbb{R}^d$, the pushforward of a distribution p is denoted as $T_\# p$, such that $T_\# p(A) = p(T^{-1}(A))$ for any measurable set A.

3 Algorithm Design

In this section, we present our core algorithm, a general framework that leverages Feel-Good Thompson Sampling (FGTS) (Zhang, 2022; Dann et al., 2021) alongside various approximate sampling techniques such as Langevin Monte Carlo (LMC) (Durmus et al., 2019) and Underdamped Langevin Monte Carlo (ULMC) (Chen et al., 2014; Cheng et al., 2018). The proposed general algorithm is displayed in Algorithm 1.

Our algorithm design resembles that of Ishfaq et al. (2024) in the sense that, unlike other approximate TS algorithms (Russo, 2019; Zanette et al., 2020a; Ishfaq et al., 2021), it performs exploration by coupling approximate sampling into value iteration step. However, our design choice offers significant flexibility of the algorithm: it allows us to employ a wide range of prior distributions and integrate different samplers, enabling us to tailor the exploration process to specific problem characteristics. The generality of our framework allows it to address suboptimality observed in existing exploration approaches (Ishfaq et al., 2024). By incorporating a flexible prior selection mechanism, we can overcome limitations inherent in specific prior choices employed by other methods. This flexibility enables us to potentially achieve better performance across diverse exploration problems.

3.1 Feel-Good Thompson Sampling

Assume we have collected data trajectories in the first $k-1$ episodes as $D_{k-1} = \{(x_1^\tau, a_1^\tau, r(x_1^\tau, a_1^\tau)), \ldots, (x_H^\tau, a_H^\tau, r(x_H^\tau, a_H^\tau))\}_{\tau=1}^{k-1}$. To estimate the Q-function for stage h at the k-th episode of the learning process, we define the following loss function for $h \in [H]$:

$$L_h^k(w_h) = \eta \sum_{\tau=1}^{k-1} \left[r_h(x_h^\tau, a_h^\tau) + \max_{a\in\mathcal{A}} Q_{h+1}^k(x_{h+1}^\tau, a) - Q(w_h; \phi(x_h^\tau, a_h^\tau))\right]^2, \tag{3.1}$$

where $\phi(\cdot,\cdot)$ is a feature vector of the corresponding state-action pair and $Q(w_h; \phi(x_h^\tau, a_h^\tau))$ denotes any possible approximation of the Q function that is parameterized by w_h and takes $\phi(x_h^\tau, a_h^\tau)$ as input. Q_h^k is defined in Line 10 of Algorithm 1 and is the truncated estimated Q function. Moreover, we let $L_0^k(w_1) = -\lambda \max_{a\in\mathcal{A}} Q(w_1; x_1^k, a)$, where L_0^k is the Feel-Good exploration prior term (Zhang, 2022). The posterior distribution at episode k and stage $h > 1$ is then given by

$$q_k^h(w_h) \propto p_0^h(w_h) \exp(-L_h^k(w_h)),$$

where p_0^h is the prior distribution of w_h. And at stage $h = 1$, we have $q_k^1(w_1) \propto p_0^1(w_1) \exp(-L_1^k(w_1) - L_0^k(w_1))$. Then at episode k, the exact target (joint) posterior of Q, which is denoted by q_k, is given by

$$q_k(w) \propto p_0(w) \exp(-\textstyle\sum_{h=0}^H L_h^k(w_h)), \tag{3.2}$$

where $p_0(w) = \prod_{h=1}^H p_0^h(w_h)$. Compared to standard TS (Thompson, 1933; Osband et al., 2013), FGTS incorporates an additional exploration term $\exp(-L_0^k(w_1))$ in the likelihood function. This term encourages the selection of value functions at the first time step that yield large values for the initial state. This bias is particularly beneficial during early learning stages when wider exploration is crucial.

Challenges of FGTS in practice: Despite the regret of FGTS has been proven to be achieving the optimal dependency on the dimension in bandits (Zhang, 2022) and reinforcement learning (Dann et al., 2021), existing FGTS algorithms are often computationally intractable as they assume access to sampling oracles for sampling from a high-dimensional distribution at each iteration. Specifically, previous FGTS based algorithms proposed by Zhang (2022); Dann et al. (2021) simply sample Q function from the posterior distribution defined in (3.2) in the beginning of each episode and then follow a greedy policy with respect to the sampled Q function. However, this assumes access to a sampling oracle which allows one to sample from (3.2) that is not generally available in practice. Since we cannot directly sample from the true posterior distribution q_k, we propose using approximating samplers to generate posterior estimates which we describe next.

3.2 Approximate Samplers

In this subsection, we present different approximate sampling methods which we use to approximately sample from the posterior distribution defined in (3.2). Let $p \propto e^{-L}$ be a probability density on $\mathbb{R}^d$ such that L is continuously differential. The goal is to generate samples from p.

Langevin Monte Carlo. LMC leverages the Euler discretization method to approximate the continuous-time Langevin diffusion process, making it a popular sampling algorithm in machine learning (Welling & Teh, 2011). Langevin diffusion with stationary distribution p is the stochastic process defined by the stochastic differential equation (SDE) $dw_t = -\nabla L(w_t)dt + \sqrt{2}dB_t$, where B_t is a standard Brownian motion in $\mathbb{R}^d$. To obtain the LMC algorithm, we take the Euler-Murayama discretization of the SDE. For a fixed step size $\tau > 0$, temperature β and $w_0 \in \mathbb{R}^d$, LMC is defined by the iteration

$$w_{k+1} = w_k - \tau \nabla L(w_k) + \sqrt{2\beta^{-1}\tau}\xi_k,$$

where $\xi_k \sim \mathcal{N}(0, I_d)$. Previous works have thoroughly established strong theoretical guarantees for the convergence of LMC (Dalalyan, 2017; Xu et al., 2018; Zou et al., 2021).

In Line 8 of Algorithm 1, we can use LMC to approximately sample w_h^{k,J_k} from the posterior defined in (3.2). In our deep RL experiment in Section 5, we also incorporate adaptive bias term for the gradient of loss function as introduced in Ishfaq et al. (2024).

Underdamped Langevin Monte Carlo. While LMC offers an elegant approach, its scalability suffers as the problem dimension, error tolerance, or condition number increases (Zheng et al., 2024; Zhang et al., 2023). To mitigate these limitations, we exploit Underdamped Langevin Monte Carlo (ULMC), which exhibits enhanced scalability in such high-dimensional or poorly conditioned settings. The appeal of ULMC lies in its connection to Hamiltonian Monte Carlo (HMC) (Brooks et al., 2011). Since underdamped Langevin diffusion (ULD) incorporates a Hamiltonian component, its discretization can be viewed as a form of HMC. Notably, HMC has been empirically observed to converge faster to the stationary distribution compared to LMC (Cheng et al., 2018). Introducing a balance of exploration and exploitation through momentum, the ULD is given by the SDE

$$\begin{aligned} dw_t &= P_t dt, \\ dP_t &= -\nabla L(w_t)dt + \gamma P_t dt + \sqrt{2\beta^{-1}\gamma} dB_t, \end{aligned}$$

where $\gamma, \beta > 0$ are friction coefficient and temperature respectively. We note that instead of using Euler-Maruyama as for LMC, the ULMC can be implemented in the following way:

$$\begin{aligned} dw_t &= P_t dt, \\ dP_t &= -\nabla L(w_{k\tau})dt + \gamma P_t dt + \sqrt{2\beta^{-1}\gamma} dB_t, \end{aligned} \tag{3.3}$$

for $t \in [k\tau, (k+1)\tau]$, where $\tau > 0$ is the step-size. This formulation of ULMC can be integrated in a closed form (Cheng et al., 2018; Zhang et al., 2023), and hence our theoretical analysis is based on this scheme. However, obtaining the closed-form solution is computationally expensive in our setting due to the cubic cost $O(d^3)$ of Cholesky decomposition. So, we employ an adapted Euler-Maruyama method in our experiments for efficient numerical integration. Applying Euler-Maruyama with step size $\tau > 0$, we obtain the following iteration scheme of w_k and P_k:

$$\begin{aligned} w_{k+1} &= w_k + \tau P_k, \\ P_{k+1} &= P_k - \tau \nabla L(w_k) - \gamma\tau P_k + \sqrt{2\beta^{-1}\gamma\tau}\xi_k, \end{aligned} \tag{3.4}$$

where $\xi_k \sim \mathcal{N}(0, I_d)$. In practice, to improve the performance, we follow Ishfaq et al. (2024) to incorporate adaptive bias term to the gradient, which leads to the following update:

$$\begin{aligned} m_k &= \alpha_1 m_{k-1} + (1-\alpha_1)\nabla L(w_k) \\ v_k &= \alpha_2 v_{k-1} + (1-\alpha_2)\nabla L(w_k) \odot \nabla L(w_k) \\ P_k &= (1-\gamma\tau)P_{k-1} + \tau\left(\nabla L(w_k) + a m_k \oslash \sqrt{v_k + \lambda\mathbf{1}}\right) + \sqrt{2\beta^{-1}\gamma\tau}\xi_k \\ w_{k+1} &= w_k - \tau P_k, \end{aligned} \tag{3.5}$$

where the hyperparameters $\alpha_1, \alpha_2 \in [0, 1)$ control the exponential decay rates of the moving averages (Kingma & Ba, 2014).

4 Theoretical Analysis

This section presents the theoretical analysis of our proposed algorithm. We begin by establishing a regret bound for general function classes, shedding light on the impact of sampling error on regret. Subsequently, we focus on linear MDPs (Jin et al., 2020), providing a detailed analysis of both the regret bound and the corresponding sampling complexity.

4.1 Regret Bound for General Function Classes

Assume that the agent is given a Q-function class $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2 \times \ldots \times \mathcal{Q}_H$ of functions $Q = \{Q_h\}_{h\in[H]}$ where $Q_h : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. For any $Q \in \mathcal{Q}, h \in [H]$ and state-action pair x, a, we define the Bellman residual as

$$\mathcal{E}_h(Q; x, a) = \mathcal{E}(Q_h, Q_{h+1}; x, a) = Q_h(x, a) - \mathcal{T}_h^* Q_{h+1}(x, a). \tag{4.1}$$

We have the following assumptions on the value-function class:

Assumption 4.1. [Realizability]. Assume that $Q^* \in \mathcal{Q}$.

Assumption 4.2. [Boundedness] Assume that $\exists b \geq 1$ such that for all $Q \in \mathcal{Q}$ and $h \in [H]$, $Q_h(x, a) \in [0, b-1]$, for all $(x, a) \in \mathcal{S} \times \mathcal{A}$.

Assumption 4.3. [Completeness] For all $h \in [H]$ and $Q_{h+1} \in \mathcal{Q}_{h+1}$, there is a $Q_h \in \mathcal{Q}_h$ such that $Q_h = \mathcal{T}_h^* Q_{h+1}$.

It's important to note that these assumptions are only necessary for general function classes. One can verify that these assumptions are satisfied in some specific settings, such as linear MDPs (Jin et al., 2020) defined in Section 4.3.

We first introduce two metrics following Dann et al. (2021) that characterizes the structural complexity of the MDP and the effective size of the value-function class $\mathcal{Q}$ respectively.

Definition 4.4. [Decoupling Coefficient] Let $\mathcal{K}_{\mathrm{DC}}$ be the smallest quantity so that for any sequence of functions $\{Q^k\}_{k\in\mathbb{N}} \subset \mathcal{Q}$ and $h \geq 0$, it holds that,

$$\sum_{h=1}^{H}\sum_{k=1}^{K}\mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^k,x_1)}\left[\mathcal{E}_h(Q^k;x_h,a_h)\right] \leq \inf_{\mu\in(0,1]}\left[\mu\sum_{h=1}^{H}\sum_{k=1}^{K}\sum_{s=1}^{k-1}\mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^s,x_1)}[\mathcal{E}_h(Q^k;x_h,a_h)]^2 + \frac{\mathcal{K}_{\mathrm{DC}}}{4\mu}\right].$$

The decoupling coefficient, $\mathcal{K}_{\mathrm{DC}}$, measures the growth rate of average Bellman residuals compared to cumulative squared Bellman residuals. We refer the readers to Dann et al. (2021) for further details on relationship between decoupling coefficient and other complexity measures typically used in RL such as Bellman-Eluder dimension (Jin et al., 2021).

Definition 4.5. For any function $Q' \in \mathcal{Q}_{h+1}$, we define the set $\mathcal{Q}_h(\epsilon, Q') = \{Q \in \mathcal{Q}_h : \sup_{x,a}|Q(x,a) - \mathcal{T}_h^* Q'(x,a)| \leq \epsilon\}$ of functions that have small Bellman error with Q' for all state-action pairs. Using this set, we define $\kappa(\epsilon) = \sup_{Q\in\mathcal{Q}}\sum_{h=1}^{H} - \ln p_0^h(\mathcal{Q}_h(\epsilon, Q_{h+1}))$.

The set $\mathcal{Q}_h(\epsilon, Q_{h+1})$ includes the functions that approximately satisfy the Bellman equation and $p_0^h(\mathcal{Q}_h(\epsilon, Q_{h+1}))$ denotes the probability that is assigned on this set by the prior. From the definition, it is clear that the complexity $\kappa(\epsilon)$ takes a small value if the prior is high for any $Q \in \mathcal{Q}$ and in that case, it is equivalent to an approximate completeness assumption. Please refer to Dann et al. (2021) for further details on this metric.

We denote the sampled posterior by Algorithm 1 at episode k by q_k', which generally deviates from the true posterior q_k defined in (3.2) due to the inherent limitations of approximate samplers discussed in Section 3.2. At each episode k, we define the sampling error $\delta_k = TV(q_k, q_k')$ as the TV distance between the true posterior and the approximate posterior generated by our sampler.

Using the quantities defined above, we present our first theorem: a frequentist (worst-case) expected regret bound for Algorithm 1:

Theorem 4.6. Under Assumption 4.1, 4.2 and 4.3, if $\eta \leq 2/5b^2$, then

$$\mathbb{E}[\mathrm{Regret}(K)] \leq \frac{\lambda}{\eta}\mathcal{K}_{\mathrm{DC}} + \frac{2K}{\lambda}\kappa(b/K^2) + \frac{6H}{\lambda} + \frac{b}{K} + \sum_{k=1}^{K}\left[\left(\frac{\eta}{4\lambda}b^2H(k-1) + b\right)\cdot\delta_k\right],$$

where the expectation incorporates the inherent randomness of the MDP through samples drawn from it and the algorithm's own stochastic elements. If we further set $\eta = 1/4b^2$, $\lambda = \sqrt{K\kappa(b/K^2)/b^2\mathcal{K}_{\mathrm{DC}}}$ and assume $\lambda b^2 \geq 1$ and without loss of generality that $b \geq 16$, then the bound becomes

$$\mathbb{E}[\mathrm{Regret}(K)] = O\left(b\sqrt{\mathcal{K}_{\mathrm{DC}}\kappa(b^2/K)K} + b^2H + \frac{b}{K}\right) + \frac{1}{16}b^2\sum_{k=1}^{K}k\delta_k. \tag{4.2}$$

Remark 4.7. It is important to emphasize that the theorem establishes the relationship between the regret and the sampling error, without necessarily asserting that the sampling error δ_k is small for general function classes. In Section 4.3, we delve deeper into controlling the sampling error with respect to the sampling complexity for linear MDPs.

Remark 4.8. The final term in (4.2) highlights that during initial episodes (small k), our approximate samplers can have relaxed accuracy requirements. This aligns with the algorithm's exploratory phase, where precise posterior estimates are less crucial compared to later exploitation stages when accurate value estimation becomes critical.

Remark 4.9. The derived regret bound from Theorem 4.6, can be decomposed into two parts:

$$R_{origin} = \frac{\lambda}{\eta}\mathcal{K}_{\mathrm{DC}} + \frac{2K}{\lambda}\kappa(b/K^2) + \frac{6H}{\lambda} + \frac{b}{K}, \quad \text{and} \quad R_{sample} = \sum_{k=1}^{K}\left[\left(\frac{\eta}{4\lambda}b^2H(k-1)+b\right)\cdot\delta_k\right]. \tag{4.3}$$

Here R_{sample} accounts for the sampling error. It is noteworthy that R_{origin} mirrors Theorem 1 from Dann et al. (2021), and consequently, we adhere to their analytical framework for deriving this part. For R_{sample}, we separately examine different samplers for their respective sampling complexities.

Remark 4.10. Note that if we can do TS exactly at each step, i.e. $\delta_k = 0$ for all k, then Theorem 4.6 reduces to Theorem 1 in Dann et al. (2021). Also as discussed in their work, the decoupling coefficient $\mathcal{K}_{\mathrm{DC}}$ can vary in different settings.

4.2 Analysis of Errors Induced by Approximating Samplers

It is important to note that δ_k within Theorem 4.6 cannot be directly controlled by the chosen approximating samplers employed in Algorithm 1. Therefore, a further decomposition of this term is necessary (see details in Appendix A.3):

Proposition 4.11. Let δ_k^h be the sampling error (in the total variation sense) induced by our sampler at step $h \in [H]$ and episode $k \in [K]$ and let δ_k be as defined in Section 4.1. Then $\delta_k \leq \sum_{h=1}^{H}\delta_k^h$.

Remark 4.12. Proposition 4.11 allows us to decompose the sampling error δ_k into individual components δ_k^h, representing the total variation distance at step h within episode k. Notably, these individual components δ_k^h are directly controllable by our approximate samplers. This translates to the overall sampling error $\delta_k \leq \sum_{h=1}^{H}\delta_k^h$ highlighting the crucial role of sampler accuracy in each step in managing the cumulative error $\sum_{h=1}^{H}\delta_k^h$.

Remark 4.13. One should expect both sampling error and truncation error to contribute to the total error δ_k, however, by considering truncation as a transport map between probability distributions and assuming that the exact target distribution is invariant with respect to the truncation due to its belonging to the given function class $\mathcal{Q}$, we are able to disregard the effect of truncation error using the data-processing inequality. See Appendix A.3 for details.

The proposition implies that the regret arising from the approximate sampler defined in (4.3) is upper-bounded by $R_{sample} \leq \sum_{k=1}^{K}\left[(\eta/4\lambda)b^2H(k-1)+b)\cdot\sum_{h=1}^{H}\delta_k^h\right]$.

4.3 Applications to Linear MDPs

A concrete example where we can interpret the regret bound from Theorem 4.6 is the linear MDP (Jin et al., 2020; Yang & Wang, 2020; 2019) setting.

Definition 4.14. (Linear MDP). An MDP $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ is said to be a linear MDP with a feature $\phi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist d unknown (signed) measures $\mu_h = (\mu_h^{(1)}, \ldots, \mu_h^{(d)})$ over $\mathcal{S}$ and an unknown vector $\theta_h \in \mathbb{R}^d$ such that for any $(x, a) \in \mathcal{S}\times\mathcal{A}$, we have $\mathbb{P}_h(\cdot|x,a) = \langle\phi(x,a), \mu_h(\cdot)\rangle$ and $r_h(x,a) = \langle\phi(x,a), \theta_h\rangle$.

Without loss of generality, we assume $\|\phi(x,a)\|_2 \leq 1$ for all $(x,a) \in \mathcal{S}\times\mathcal{A}$, and $\max\{\|\mu_h(\mathcal{S})\|_2, \|\theta_h\|_2\} \leq \sqrt{d}$ for all $h \in [H]$.

We first bound $\kappa(\epsilon)$ defined in Definition 4.5 for linear MDP. While previous work by Dann et al. (2021) provides bounds with a uniform prior distribution over the function class, it does not align with the way TS algorithms are implemented in practice. For this, we consider a Gaussian distribution as the prior distribution.

Algorithm	Regret	Exploration	Computational Tractability	Sampling Complexity
LSVI-UCB (Jin et al., 2020)	$\tilde{O}(d^{3/2}H^{3/2}\sqrt{T})$	UCB	Yes	NA
OPT-RLSVI (Zanette et al., 2020a)	$\tilde{O}(d^2H^2\sqrt{T})$	TS	Yes	NA
ELEANOR (Zanette et al., 2020b)	$\tilde{O}(dH^{3/2}\sqrt{T})$	Optimism	No	NA
CPS (Dann et al., 2021)	$\tilde{O}(dH^2\sqrt{T})$	FGTS	No	NA
LSVI-PHE (Ishfaq et al., 2021)	$\tilde{O}(d^{3/2}H^{3/2}\sqrt{T})$	TS	Yes	NA
LMC-LSVI (Ishfaq et al., 2024)	$\tilde{O}(d^{3/2}H^{3/2}\sqrt{T})$	LMC	Yes	$\tilde{\Theta}(\frac{\kappa^3K^3H^3}{d\ln(dT)})$
LSVI-ASE with LMC sampler	$\tilde{O}(dH^{3/2}\sqrt{T})$	FGTS & LMC	Yes	$\tilde{\Theta}(\frac{\kappa^3K^3H^3}{d\ln(dT)})$
LSVI-ASE with ULMC sampler	$\tilde{O}(dH^{3/2}\sqrt{T})$	FGTS & ULMC	Yes	$\tilde{\Theta}(\frac{\kappa^{3/2}K^2H^2}{\sqrt{d\ln(dT)}})$

Table 1: Regret upper bound for episodic, non-stationary, linear MDPs. Here, computational tractability refers to the ability of a computational problem to be solved in a reasonable amount of time using a feasible amount of computational resources.

Lemma 4.15. If the stage-wise priors p_0^h are chosen as $\mathcal{N}(0, \sqrt{d}HI_d)$, then $\kappa(\epsilon) = dHO(\ln(dH/\epsilon))$.

Remark 4.16. While Gaussian priors are commonly used (He et al., 2015; Goodfellow et al., 2016), we highlight that the prior distribution p_0^h can be any distribution in practice, as long as a suitable bound for $\kappa(\epsilon)$ exists. This flexibility allows for incorporating domain-specific knowledge into the prior.

We can now illustrate Theorem 4.6 for linear MDP:

Corollary 4.17. If we set $\eta = 2/5H^2$ and $\lambda = \sqrt{K\kappa(H/K^2)/dH^3(1+\ln(2T))}$, then the expected regret of Algorithm 1 after K episodes in a linear MDP is bounded as

$$\mathbb{E}[\text{Regret}(K)] = O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT)) + \textstyle\sum_{k=1}^{K}\alpha_k\big(\sum_{h=1}^{H}\delta_k^h\big),$$

where $\alpha_k = O(\sqrt{\ln(dT)/K}H^2k)$ and $T = HK$ is the total number of steps.

4.4 Sampling Complexity of Different Samplers

In this section, we characterize the sampling complexity of the proposed algorithms to demonstrate that we can achieve the desired regret bound as long as the chosen sampler is executed a sufficient number of times. We begin by establishing an appropriate notion of complexity.

Definition 4.18. (Sampling Complexity) The agent has access to the gradient $\nabla_w Q(w; \phi(x, a))$ for any $w \in \mathbb{R}^d$. Then, if $\nabla_w Q$ is evaluated G_k times at any episode $k \in [K]$, then we define G_k as the sampling complexity at episode k, and $SC = \sum_{k\in[K]} G_k$ be the cumulative sampling complexity.

Remark 4.19. In Algorithm 1, G_k specifically represents the total number of iterations employed by our approximate samplers from line 5 to line 9 during episode k. It follows that within our analysis, $G_k = J_k$ and $SC = \sum_{k\in[K]} G_k = \sum_{k\in[K]} J_k$.

Theorem 4.20. Consider a linear MDP defined in Definition 4.14. Assume that there exists $\kappa > 0$ such that for any $(k, h) \in [K] \times [H]$, the loss function defined in (3.1) satisfies $M_{k,h}I \geq \nabla^2 L_h^k \geq m_{k,h}I$ and $M_{k,h}/m_{k,h} \leq \kappa$ for some $M_{k,h} \geq m_{k,h} > 0$. Then we can achieve the regret bound of $O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT))$ using our approximate samplers with the cumulative sampling complexity stated below:
(1) LMC: $SC = \tilde{\Theta}(\frac{\kappa^3K^3H^3}{d\ln(dT)})$ with step size $\tau_{k,h} = \tilde{\Theta}(\frac{d\ln(dT)}{M_{k,h}H^2k^2\kappa})$;
(2) ULMC: $SC = \tilde{\Theta}(\frac{\kappa^{3/2}K^2H^2}{\sqrt{d\ln(dT)}})$ with step size $\tau_{k,h} = \tilde{\Theta}(\frac{\sqrt{d\ln(dT)}}{M_{k,h}Hk})$.

Remark 4.21. Theorem 4.20 reveals a critical relationship between the choice of sampling method and the sampling complexity of Algorithm 1. Leveraging established results on demonstrating the faster mixing of ULMC over LMC in strongly log-concave settings (see Appendix B for details), the theorem confirms that Algorithm 1, when employing ULMC, achieves the desired accuracy with lower data requirements than its LMC-based counterpart. This aligns with the intuitive notion that ULMC's momentum-based exploration enables faster learning, thereby reducing the necessary data for effective Thompson sampling.

5 Experiments

In this section, we provide an empirical evaluation of our proposed algorithms with deep Q-networks (DQNs) (Mnih et al., 2015) in two sets of environments: (1) the N-chain environment (Osband et al., 2016a) and (2) the Atari game suite (Bellemare et al., 2013; Taiga et al., 2019). We evaluate our algorithms with different implementations. In particular, we implement the Algorithm 1 with different choices of prior terms and samplers. By choosing Feel-Good exploration prior term in (3.1) and underdamped Langevin Monte Carlo sampler with adaptive bias term in (3.5), we implement the Algorithm 1 named as Feel-Good Underdamped Langevin Monte Carlo Deep Q-Network (FG-ULMCDQN). We implement the Algorithm 1 with the Feel-Good exploration prior term and the adaptive Langevin Monte Carlo sampler introduced in Ishfaq et al. (2024), named Feel-Good Langevin Monte Carlo Deep Q-Network (FG-LMCDQN). We also provide an implementation for the Algorithm 1 without the Feel-Good exploration prior term named Underdamped Langevin Monte Carlo Deep Q-Network (ULMCDQN). Then we evaluate our implementations in the above mentioned environments. Our code is available at `https://github.com/panxulab/LSVI-ASE`.

5.1 Experiments in N-Chain

We demonstrate that our proposed algorithms can explore effectively in sparse-reward environment by conducting experiments in N-Chain environment (Osband et al., 2016a) that demands deep exploration capabilities to perform well. An N-chain environment can be constructed by a chain of $N > 3$ states denoted by $s_1, s_2, \ldots, s_N$. Each episode of interaction, which starts at state s_2, lasts for $N + 9$ steps and in each step the agent can either move to the left or right. A myopic agent would gravitate toward state s_1 which has a small reward of $r = 0.001$ whereas an efficient agent with deep exploration capabilities would try to reach state s_N which has a larger reward of $r = 1$. As each episode runs for $N + 9$ steps, the optimal return for an episode is 10. We refer the reader to Appendix C.1 for a depiction of the environment.

In our experiments, we progressively increase the difficulty level by setting N to be 25, 50, 75, and 100. For each chain length, we run each learning algorithm for 10^5 steps across 20 seeds. As baseline algorithms, we use DQN (Mnih et al., 2015), Bootstrapped DQN (Osband et al., 2016a), Noisy-Net (Fortunato et al., 2018) and Adam LMCDQN (Ishfaq et al., 2024). The performance of each algorithm in each run is measured by the mean return of the last 10 evaluation episodes. We sweep the learning rate and pick the one with the best performance for each algorithm. For our algorithms which use ULMC as a sampler, we sweep the friction coefficient γ. For FG-LMCDQN and FG-ULMCDQN, we sweep the weight for the feel-good prior term η in the loss function. Please check Appendix C.1 for further details.

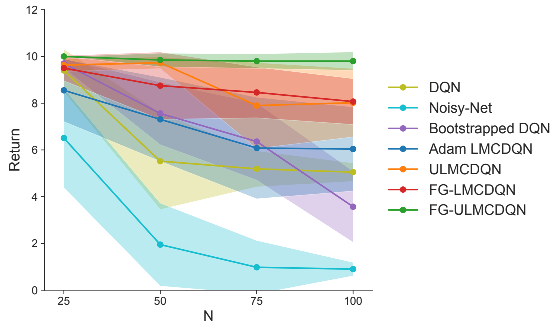


Figure 1: A comparison of different methods in N-chain with different chain lengths N. As N increases, the exploration hardness increases. All results are averaged over 20 runs and the shaded areas represent 95% confidence interval.

Figure 1 shows the performance of our proposed algorithms as well as the baseline algorithms under different chain lengths. The solid lines represent the average return over 20 random seeds and the shaded areas represent the 95% confidence interval. For all of our proposed algorithms, namely ULMCDQN, FG-ULMCDQN and FG-LMCDQN, we set $J_k = 4$ in Algorithm 1 for all chain lengths.

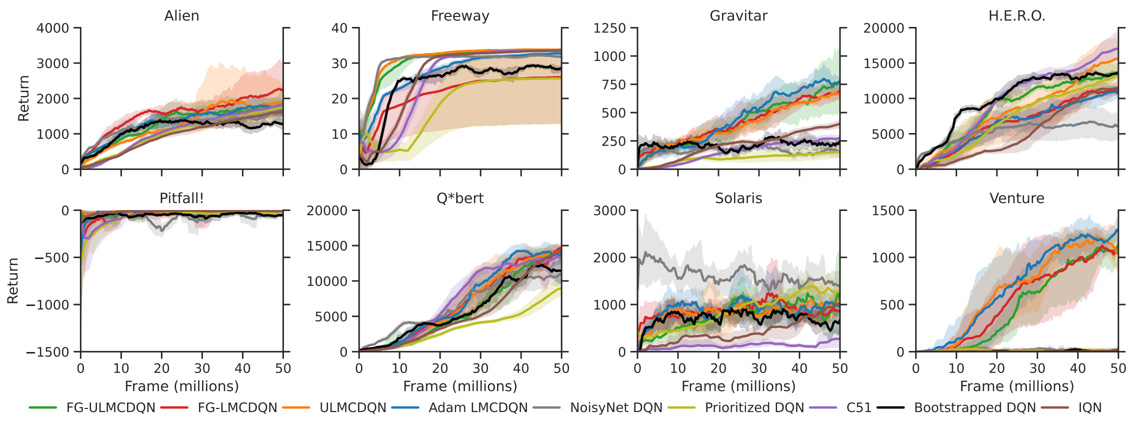


Figure 2: The return curves of various algorithms in eight hard Atari tasks over 50 million training frames. Solid lines correspond to the median performance over 5 random seeds, and the shaded areas correspond to 95% confidence interval.

From Figure 1, we see that as the chain length N increases, the performance of the baselines drops drastically. Whereas, our FGTS based algorithms FG-LMCDQN and FG-ULMCDQN are able to maintain steady performance. In particular, we would like to highlight that FG-ULMCDQN is able to get almost close to the optimal return of 10 for all values of N, showing the benefit of using Feel Good prior along with underdamped LMC together in environments where deep exploration is absolutely necessary to perform well.

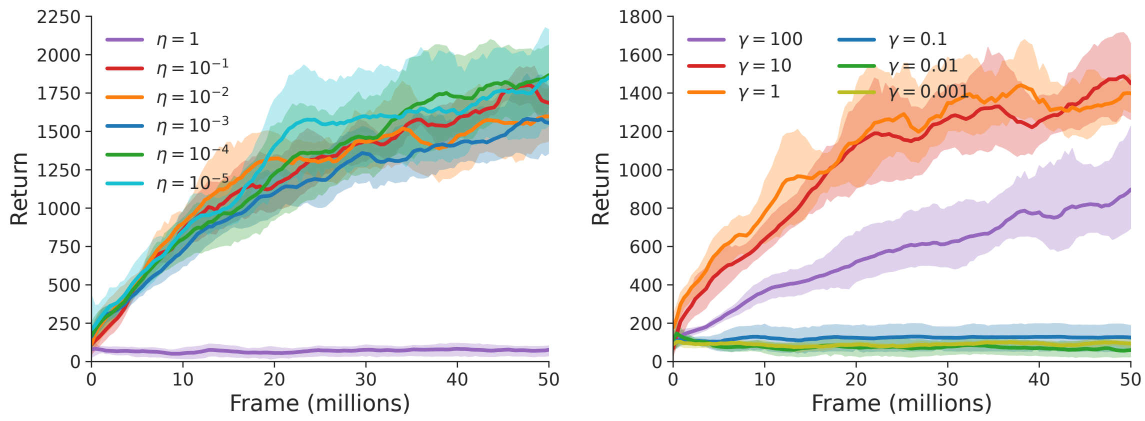
5.2 Experiments in Atari Games

We evaluate our algorithms in 8 visually complicated hard exploration games, namely Alien, Freeway, Gravitar, H.E.R.O., Pitfall, Qbert, Solaris, and Venture from the Atari game suite (Bellemare et al., 2013; Taiga et al., 2019). As classified in Taiga et al. (2019), among these games, Alien, H.E.R.O., and Qbert are dense reward environments and Freeway, Gravitar, Pitfall, Solaris, and Venture are sparse reward environments. In our experiments, we set $J_k = 1$ in Algorithm 1 to finish the training in a reasonable time. Following (Ishfaq et al., 2024), we also incorporate the double Q trick (Van Hasselt, 2010; Van Hasselt et al., 2016) in our implementation. As baselines we consider Adam LMCDQN (Ishfaq et al., 2024), Noisy-Net (Fortunato et al., 2018), Prioritized DQN (Schaul et al., 2015), C51 (Bellemare et al., 2017), Bootstrapped DQN (Osband et al., 2016a) and IQN (Dabney et al., 2018). All algorithms are trained for 50M frames (i.e., 12.5M steps) and run for 5 different random seeds. We refer the reader to Appendix C.2.1 for further details on training and hyper-parameter choices. Figure 2 depicts the learning curves of all algorithms in 8 Atari games. Compared to the baseline algorithms, our algorithms appear to be quite competitive despite being much simpler in implementation. We highlight the advantages of approximate sampling based algorithms in Gravitar and Venture.

Sensitivity Analysis. In Figure 3a, we draw the learning curves of FG-ULMCDQN with different weight factor η for the FG prior term. We observe that as long the value of η is not too high, the performance of FG-ULMCDQN is less sensitive to the value of η. In Figure 3b, we observe that for very high or low value of friction coefficient γ, the performance of ULMCDQN collapses.

6 Conclusion

This work introduces a novel algorithmic framework that leverages efficient approximate samplers to make FGTS practical for real-world RL. Unlike prior approaches reliant on unrealistic sampling oracles, our framework enables computationally feasible exploration. Furthermore, our theoretical



(a) Different FG prior weight η in FG-ULMCDQN (b) Different friction coefficient γ in ULMCDQN

Figure 3: (a) A comparison of FG-ULMCDQN with different values of weight η for the feel good prior term in Alien. Solid lines correspond to the average performance over 5 random seeds, and shaded areas correspond to 95% confidence interval. The performance of FG-ULMCDQN is not very sensitive to the values of η as long it is not very large. (b) A comparison of ULMCDQN with different values of the friction coefficient γ in Alien.

analysis provides a deeper understanding of the relationship between samplers and regret in FGTS algorithms. This newfound knowledge paves the way for practical exploration strategies with strong provable guarantees. Notably, our algorithm achieves an improved regret bound in linear MDPs, and showcases consistent performance in deep exploration environments.

Future directions include exploring the integration of alternative approximate samplers within our framework. Promising candidates include Metropolis-adjusted Langevin Acceptance (MALA) (Besag et al., 1995) and various proximal sampling algorithms (Lee et al., 2021). Investigating efficient methods to incorporate these samplers into the RL setting while maintaining the framework's strengths will further enhance its applicability to diverse exploration problems.

Acknowledgments

We gratefully acknowledge funding from the Canada CIFAR AI Chairs program, the Reinforcement Learning and Artificial Intelligence (RLAI) laboratory, Mila - Quebec Artificial Intelligence Institute, the Natural Sciences and Engineering Research Council (NSERC) of Canada, the US National Science Foundation (DMS-2323112) and the Whitehead Scholars Program at the Duke University School of Medicine. Qingfeng Lan would also like to acknowledge Alberta Innovates for the support they provided him for his research.

References

Alekh Agarwal and Tong Zhang. Model-based RL with optimistic posterior sampling: Structural conditions and sample complexity. In *Advances in Neural Information Processing Systems*, 2022a. (p. 2.)

Alekh Agarwal and Tong Zhang. Non-linear reinforcement learning in large action spaces: Structural conditions and sample-efficiency of posterior sampling. In *Conference on Learning Theory*, pp. 2776–2814. PMLR, 2022b. (p. 2.)

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013. (pp. 2, 10, and 11.)

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017. (pp. 11 and 25.)

Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019. (p. 3.)

Julian Besag, Peter Green, David Higdon, and Kerrie Mengersen. Bayesian computation and stochastic systems. *Statistical science*, pp. 3–41, 1995. (p. 12.)

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, May 2011. ISBN 9780429138508. doi: 10.1201/b10905. URL `http://dx.doi.org/10.1201/b10905`. (p. 5.)

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pp. 1683–1691. PMLR, 2014. (pp. 2 and 4.)

Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Conference on Learning Theory*, pp. 300–323. PMLR, 2018. (pp. 2, 4, and 6.)

Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of Langevin Monte Carlo from Poincare to Log-Sobolev. In *Conference on Learning Theory*, pp. 1–2. PMLR, 2022. (p. 23.)

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International Conference on Machine Learning*, pp. 1096–1105. PMLR, 2018. (pp. 11 and 25.)

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):651–676, 2017. (p. 5.)

Christoph Dann, Mehryar Mohri, Tong Zhang, and Julian Zimmert. A provably efficient model-free posterior sampling method for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12040–12051, 2021. (pp. 1, 2, 3, 4, 5, 7, 8, 9, 18, 20, and 22.)

Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of Langevin Monte Carlo via convex optimization. *The Journal of Machine Learning Research*, 20(1):2666–2711, 2019. (pp. 2 and 4.)

Vikranth Dwaracherla and Benjamin Van Roy. Langevin DQN. *arXiv preprint arXiv:2002.07282*, 2020. (pp. 2 and 3.)

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *International Conference on Learning Representations*, 2018. (pp. 1, 3, 10, 11, and 25.)

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. (p. 9.)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015. (p. 9.)

Hao-Lun Hsu, Weixin Wang, Miroslav Pajic, and Pan Xu. Randomized exploration in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2404.10728*, 2024. (pp. 2 and 3.)

Haque Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin Yang. Randomized exploration in reinforcement learning with general value function approximation. In *International Conference on Machine Learning*, pp. 4607–4616. PMLR, 2021. (pp. 1, 2, 4, and 9.)

Haque Ishfaq, Qingfeng Lan, Pan Xu, A Rupam Mahmood, Doina Precup, Anima Anandkumar, and Kamyar Azizzadenesheli. Provable and practical: Efficient exploration in reinforcement learning via Langevin Monte Carlo. In *The Twelfth International Conference on Learning Representations*, 2024. (pp. 1, 2, 3, 4, 5, 6, 9, 10, 11, 24, and 25.)

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020. (pp. 1, 2, 6, 7, 8, and 9.)

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. (p. 7.)

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (p. 6.)

Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Structured logconcave sampling with a restricted Gaussian oracle. In *Conference on Learning Theory*, pp. 2993–3050. PMLR, 2021. (p. 12.)

Yingru Li, Jiawei Xu, Lei Han, and Zhi-Quan Luo. Q-star meets scalable posterior sampling: Bridging theory and practice via hyperagent. In *Forty-first International Conference on Machine Learning*, 2024. (p. 1.)

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. (pp. 10 and 24.)

Ian Osband, Daniel Russo, and Benjamin Van Roy. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013. (pp. 1 and 5.)

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *Advances in Neural Information Processing Systems*, 29, 2016a. (pp. 1, 2, 10, 11, 24, and 25.)

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pp. 2377–2386. PMLR, 2016b. (p. 1.)

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018. (pp. 1 and 2.)

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018. (p. 1.)

John Quan and Georg Ostrovski. DQN Zoo: Reference implementations of DQN-based agents, 2020. URL `http://github.com/deepmind/dqn_zoo`. (p. 25.)

Daniel Russo. Worst-case regret bounds for exploration via randomized value functions. In *Advances in Neural Information Processing Systems*, pp. 14410–14420, 2019. (pp. 1, 2, and 4.)

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015. (pp. 11 and 25.)

Malcolm Strens. A Bayesian framework for reinforcement learning. In *ICML*, volume 2000, pp. 943–950, 2000. (p. 1.)

Adrien Ali Taiga, William Fedus, Marlos C Machado, Aaron Courville, and Marc G Bellemare. On bonus based exploration methods in the Arcade learning environment. In *International Conference on Learning Representations*, 2019. (pp. 10 and 11.)

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. (pp. 1 and 5.)

Hado Van Hasselt. Double Q-learning. *Advances in Neural Information Processing Systems*, 23, 2010. (p. 11.)

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016. (pp. 11 and 25.)

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. *Advances in Neural Information Processing Systems*, 32, 2019. (p. 23.)

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688. Citeseer, 2011. (p. 5.)

Jiayi Weng, Huayu Chen, Dong Yan, Kaichao You, Alexis Duburcq, Minghao Zhang, Yi Su, Hang Su, and Jun Zhu. Tianshou: A highly modularized deep reinforcement learning library. *Journal of Machine Learning Research*, 2022. (pp. 24 and 25.)

Zhihan Xiong, Ruoqi Shen, Qiwen Cui, Maryam Fazel, and Simon Shaolei Du. Near-optimal randomized exploration for tabular Markov decision processes. In *Advances in Neural Information Processing Systems*, 2022. (pp. 1 and 2.)

Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of Langevin dynamics based algorithms for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018. (p. 5.)

Pan Xu, Hongkai Zheng, Eric V Mazumdar, Kamyar Azizzadenesheli, and Animashree Anandkumar. Langevin Monte Carlo for contextual bandits. In *International Conference on Machine Learning*, pp. 24830–24850. PMLR, 2022. (p. 2.)

Lin Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pp. 6995–7004. PMLR, 2019. (p. 8.)

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020. (p. 8.)

Andrea Zanette, David Brandfonbrener, Emma Brunskill, Matteo Pirotta, and Alessandro Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964. PMLR, 2020a. (pp. 1, 2, 4, and 9.)

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent Bellman error. In *International Conference on Machine Learning*, pp. 10978–10989. PMLR, 2020b. (p. 9.)

Shunshi Zhang, Sinho Chewi, Mufan Li, Krishna Balasubramanian, and Murat A Erdogdu. Improved discretization analysis for underdamped Langevin Monte Carlo. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 36–71. PMLR, 2023. (pp. 5, 6, and 23.)

Tong Zhang. Feel-good Thompson sampling for contextual bandits and reinforcement learning. *SIAM Journal on Mathematics of Data Science*, 4(2):834–857, 2022. (pp. 1, 2, 3, 4, and 5.)

Haoyang Zheng, Wei Deng, Christian Moya, and Guang Lin. Accelerating approximate Thompson sampling with underdamped Langevin Monte Carlo. *arXiv preprint arXiv:2401.11665*, 2024. (p. 5.)

Difan Zou, Pan Xu, and Quanquan Gu. Faster convergence of stochastic gradient Langevin dynamics for non-log-concave sampling. In *Uncertainty in Artificial Intelligence*, pp. 1152–1162. PMLR, 2021. (p. 5.)

A Regret Analysis

A.1 Proof of Main Results

In this section, we restate and provide the proof of our main result Theorem 4.6 and its corollary for the linear MDP.

Theorem A.1. [Restatement of Theorem 4.6] Under Assumption 4.1, 4.2 and 4.3, if $\eta \leq 2/5b^2$, then

$$\mathbb{E}[\text{Regret}(K)] \leq \frac{\lambda}{\eta}\mathcal{K}_{\text{DC}} + \frac{2K}{\lambda}\kappa(b/K^2) + \frac{6H}{\lambda} + \frac{b}{K} + \sum_{k=1}^{K}\left[\left(\frac{\eta}{4\lambda}b^2H(k-1)+b\right)\cdot\delta_k\right],$$

where the expectation incorporates the inherent randomness of the MDP environment through samples drawn from it and the algorithm's own stochastic elements. If we further set $\eta = 1/4b^2$ and $\lambda = \sqrt{\frac{K\kappa(b/K^2)}{b^2\mathcal{K}_{\text{DC}}}}$ and assume $\lambda b^2 \geq 1$ and without loss of generality that $b \geq 16$, then the bound becomes

$$\mathbb{E}[\text{Regret}(K)] = O\left(b\sqrt{\mathcal{K}_{\text{DC}}\kappa(b^2/K)K} + b^2H + \frac{b}{K}\right) + \frac{1}{16}b^2\sum_{k=1}^{K}k\delta_k. \tag{A.1}$$

Proof of Theorem 4.6. Given any policy π^k and initial state x_1^k, by Lemma A.4, we can decompose the regret:

$$\begin{aligned}\text{REG}_k &= \sum_{h=1}^{H}\mathbb{E}_{\pi^k}\left[Q_h^k(x_h^k,a_h^k) - r_h(x_h^k,a_h^k) - \mathbb{E}_{x_{h+1}\sim\mathbb{P}_h(\cdot|x_h^k,a_h^k)}\max_{a\in\mathcal{A}}Q_{h+1}^k(x_{h+1},a)\right]\\ &\quad - \left[V_1^k(x_1^k) - V_1^*(x_1^k)\right]\\ &=: \sum_{h=1}^{H}\text{BE}_k^h - \text{FG}_k.\end{aligned}$$

and hence we can rewrite the expected regret of episode k (scaled by λ) as

$$\begin{aligned}\lambda\mathbb{E}_{Q^k\sim q_k'}\text{REG}_k &= \lambda\mathbb{E}_{Q^k\sim q_k'}\left[\sum_{h=1}^{H}\text{BE}_k^h - \text{FG}_k\right]\\ &= \mathbb{E}_{Q^k\sim q_k'}\sum_{h=1}^{H}\left[\lambda\text{BE}_k^h - \frac{\eta}{4}\sum_{s=1}^{k-1}\mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^s,x_1)}[\mathcal{E}_h(Q^k;x_h,a_h)]^2\right]\\ &\quad + \mathbb{E}_{Q^k\sim q_k'}\left[\sum_{h=1}^{H}\frac{\eta}{4}\sum_{s=1}^{k-1}\mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^s,x_1)}[\mathcal{E}_h(Q^k;x_h,a_h)]^2 - \lambda\text{FG}_k\right]\\ &=: F_k^{dc} + F_k^{\kappa}.\end{aligned}$$

Summing over $k = 1, 2, \dots, K$, we obtain that

$$\lambda\mathbb{E}[\text{Regret}(K)] = \sum_{k=1}^{K}F_k^{dc} + \sum_{k=1}^{K}F_k^{\kappa}.$$

Using Definition 4.4, we can bound the first term by

$$\sum_{k=1}^{K}F_k^{dc} \leq \frac{\lambda^2}{\eta}\mathcal{K}_{\text{DC}} \tag{A.2}$$

for any $\frac{\eta}{4\lambda} < 1$. For the F_k^κ term, by Assumption 4.2, $\mathcal{E}_h(Q^k; x_h, a_h)^2 \le b^2$ and $|\mathrm{FG}_k| \le b$. Therefore,

$$\left| \sum_{h=1}^{H} \frac{\eta}{4} \sum_{s=1}^{k-1} \mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^s,x_1)} [\mathcal{E}_h(Q^k; x_h, a_h)]^2 - \lambda \mathrm{FG}_k \right| \le \frac{\eta}{4} b^2 H(k-1) + \lambda b.$$

Then by the property of total variation distance,

$$F_k^\kappa \le \mathbb{E}_{Q^k\sim q_k}\left[\sum_{h=1}^{H} \frac{\eta}{4} \sum_{s=1}^{k-1} \mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^s,x_1)} [\mathcal{E}_h(Q^k; x_h, a_h)]^2 - \lambda \mathrm{FG}_k \right] + \left[\frac{\eta}{4} b^2 H(k-1) + \lambda b\right] \cdot \delta_k,$$

where $\delta_k = TV(q_k, q_k')$. The first term on the right handside is exactly the analog of F_k^κ with the expectation taken over the exact target distribution of our algorithm. By Theorem A.5 (which is Theorem 1 of Dann et al. (2021)),

$$\sum_{k=1}^{K} \mathbb{E}_{Q^k\sim q_k}\left[\sum_{h=1}^{H} \frac{\eta}{4} \sum_{s=1}^{k-1} \mathbb{E}_{[x_h,a_h]\sim p(\cdot|Q^s,x_1)} [\mathcal{E}_h(Q^k; x_h, a_h)]^2 - \lambda \mathrm{FG}_k \right] \le 2K\kappa(b/K^2) + 6H + \frac{\lambda b}{K}.$$

Hence

$$\sum_{k=1}^{K} F_k^\kappa \le 2K\kappa(b/T^2) + 6H + \frac{\lambda b}{K} + \sum_{k=1}^{K} \left[\frac{\eta}{4} b^2 H(k-1) + \lambda b\right] \cdot \delta_k.$$

Combining this inequality with (A.2), we obtain that

$$\begin{aligned} \mathbb{E}[\mathrm{Regret}(K)] &\le \frac{1}{\lambda}\left[\sum_{k=1}^{K} F_k^{dc} + \sum_{k=1}^{K} F_k^\kappa \right] \\ &\le \frac{\lambda}{\eta}\mathcal{K}_{\mathrm{DC}} + \frac{2K}{\lambda}\kappa(b/K^2) + \frac{6H}{\lambda} + \frac{b}{K} + \sum_{k=1}^{K}\left[\frac{\eta}{4\lambda} b^2 H(k-1) + b\right] \cdot \delta_k. \end{aligned}$$

If we further set $\eta = 1/4b^2$ and $\lambda = \sqrt{\frac{K\kappa(b/K^2)}{b^2\mathcal{K}_{\mathrm{DC}}}}$, then a direct calculation gives us

$$\frac{\lambda}{\eta}\mathcal{K}_{\mathrm{DC}} + \frac{2K}{\lambda}\kappa(b/K^2) + \frac{6H}{\lambda} = 6b\sqrt{\mathcal{K}_{\mathrm{DC}}\kappa(b/K^2)K} + 6H\sqrt{\frac{b^2\mathcal{K}_{\mathrm{DC}}}{\kappa(b/K^2)K}}.$$

Since $\lambda b^2 \ge 1$, we have

$$\sqrt{\frac{b^2\mathcal{K}_{\mathrm{DC}}}{\kappa(b/K^2)K}} \le b^2.$$

And since $b \ge 16$, we have

$$\frac{\eta}{4\lambda} b^2 H(k-1) + b \le \frac{1}{16} b^2 (k-1) + b \le \frac{1}{16} b^2 k.$$

Combining these results, we obtain (4.2). □

Next, we restate Corollary 4.17 and provide proof for it.

Corollary A.2. Assume Algorithm 1 is run on a d-dimensional linear MDP. If we set $\eta = \frac{2}{5H^2}$ and $\lambda = \sqrt{\frac{K\kappa(H/K^2)}{dH^3(1+\ln(2T))}}$, then the expected regret after K episodes is bounded as

$$\mathbb{E}[\mathrm{Regret}(K)] \le O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT)) + \sum_{k=1}^{K} \alpha_k \Big(\sum_{h=1}^{H} \delta_k^h\Big),$$

where $\alpha_k = O\left(\sqrt{\frac{\ln(dT)}{K}} H^2 k\right)$ and $T = HK$ is the total number of steps.

Proof of Corollary 4.17. For linear MDPs, by proposition A.8, $b = H$, $\mathcal{K}_{\mathrm{DC}} \leq 2dH(1 + 2\ln(KH))$ and $\kappa(H/K^2) = dH\ln(dHK)$. Therefore R_{origin} defined in (4.3) becomes $O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT))$. Moreover, by a direct calculation,

$$\alpha_k := \frac{\eta}{4\lambda}b^2H(k-1) + b = O\left(\sqrt{\frac{\ln(dT)}{K}}H^2k\right)$$

and hence R_{sample} defined in (4.3) becomes

$$\sum_{k=1}^{K}\alpha_k\delta_k \leq \sum_{k=1}^{K}\alpha_k\big(\sum_{h=1}^{H}\delta_k^h\big),$$

where the last inequality is due to Proposition 4.11. In conclusion,

$$\mathbb{E}[\mathrm{Regret}(K)] \leq R_{origin} + R_{sample} = O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT)) + \sum_{k=1}^{K}\alpha_k\big(\sum_{h=1}^{H}\delta_k^h\big).$$

□

With this result, we are ready to prove Theorem 4.20:

Theorem A.3 (Restatement of Theorem 4.20)**.** Consider a linear MDP defined in Definition 4.14. Assume that there exists $\kappa > 0$ such that for any $(k,h) \in [K]\times[H]$, the loss function defined in (3.1) satisfies for some $M_{k,h} \geq m_{k,h} > 0$:

$$M_{k,h}I \geq \nabla^2 L_h^k \geq m_{k,h}I, \quad M_{k,h}/m_{k,h} \leq \kappa.$$

Then we can achieve the regret bound $O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT))$ using our approximate samplers with the cumulative sampling complexity stated below:
(1) LMC: $SC = \tilde{\Theta}(\frac{\kappa^3K^3H^3}{d\ln(dT)})$ with step size $\tau_{k,h} = \tilde{\Theta}(\frac{d\ln(dT)}{M_{k,h}H^2k^2\kappa})$;
(2) ULMC: $SC = \tilde{\Theta}(\frac{\kappa^{3/2}K^2H^2}{\sqrt{d\ln(dT)}})$ with step size $\tau_{k,h} = \tilde{\Theta}(\frac{\sqrt{d\ln(dT)}}{M_{k,h}Hk})$.

Proof of Theorem 4.20. We give the proof for ULMC. The proof for LMC is essentially the same using Theorem B.1. Note that for linear MDP, if we let

$$\max_{h\in[H]}\delta_k^h \leq \frac{O(Hd\ln(dT)/\sqrt{K})}{\alpha_k} = O\left(\frac{d\sqrt{\ln(dT)}}{Hk}\right) \tag{A.3}$$

at episode k, then $\alpha_k\sum_{h=1}^{H}\delta_k^h \leq O(H^2d\ln(dT)/\sqrt{K})$. This implies that for linear MDP, $\mathbb{E}[\mathrm{Regret}(K)] = O(dH^{\frac{3}{2}}\sqrt{T}\ln(dT))$. For ULMC, by Theorem B.2, the requirements in (A.3) can be achieved by setting the step size $\tau_{k,h} = \tilde{\Theta}(\frac{\sqrt{d\ln(dKH)}}{M_{k,h}Hk})$ and after $N_{k,h} = \tilde{\Theta}(\frac{\kappa^{3/2}Hk}{\sqrt{d\ln(dKH)}})$ iterations. Summing $N_{k,h}$ over all $k \in [K]$ and $h \in [H]$, we obtain that the cumulative sample complexity is $\tilde{\Theta}(\frac{\kappa^{3/2}K^2H^2}{\sqrt{d\ln(dKH)}})$. □

A.2 Useful Lemmas

In this section, we give some lemmas that are useful in the proof of our main results.

Lemma A.4 (Regret Decomposition)**.** The regret at episode k can be decomposed into two terms

$$\mathrm{REG}_k = \mathbb{E}_{\pi^k,\mathbb{P}}\left[\sum_{h=1}^{H}Q_h^k(x_h^k,a_h^k)) - [\mathcal{T}_h^*Q_{h+1}^k](x_h^k,a_h^k)\right] - [V_1^k(x_1^k) - V_1^*(x_1^k)]$$

Proof of Lemma A.4. Recall that the Bellman optimality operator $\mathcal{T}_h^*$ maps any state-action function Q_{h+1}^k to

$$[\mathcal{T}_h^* Q_{h+1}^k](x,a) = r_h(x,a) + \mathbb{E}_{x' \sim \mathbb{P}_h(x,a)}[\max_{a' \in \mathcal{A}} Q_{h+1}^k(x',a')],$$

and hence

$$r_h(x_h^k, a_h^k) = [\mathcal{T}_h^* Q_{h+1}^k](x_h^k, a_h^k) - \mathbb{E}_{x_{h+1}^k \sim \mathbb{P}_h(x_h^k, a_h^k)}[\max_{a \in \mathcal{A}} Q_{h+1}^k(x_{h+1}^k, a)].$$

By definition,

$$V_1^{\pi^k}(x_1^k) = \mathbb{E}_{\pi^k}\left[\sum_{h=1}^H r_h(x_h^k, a_h^k) \middle| x_1 = x_1^k\right].$$

And then, we have

$$\begin{aligned}
V_1^{\pi^k}(x_1^k) &= \mathbb{E}_{\pi^k}\left[\sum_{h=1}^H r_h(x_h^k, a_h^k) \middle| x_1 = x_1^k\right] \\
&= \mathbb{E}_{\pi^k}\left[\sum_{h=1}^H \left[[\mathcal{T}_h^* Q_{h+1}^k](x_h^k, a_h^k) - \mathbb{E}_{x_{h+1}^k \sim \mathbb{P}_h(x_h^k, a_h^k)}[\max_{a \in \mathcal{A}} Q_{h+1}^k(x_{h+1}^k, a)]\right] \middle| x_1 = x_1^k\right] \\
&= \mathbb{E}_{\pi^k, \mathbb{P}}\left[\sum_{h=1}^H \left[[\mathcal{T}_h^* Q_{h+1}^k](x_h^k, a_h^k) - \max_{a \in \mathcal{A}} Q_{h+1}^k(x_{h+1}^k, a)\right] \middle| x_1 = x_1^k\right] \\
&= \mathbb{E}_{\pi^k, \mathbb{P}}\left[\sum_{h=1}^H \left[[\mathcal{T}_h^* Q_{h+1}^k](x_h^k, a_h^k) - Q_{h+1}^k(x_{h+1}^k, a_{h+1}^k)\right] \middle| x_1 = x_1^k\right] \\
&= \mathbb{E}_{\pi^k, \mathbb{P}}\left[\sum_{h=1}^H \left[[\mathcal{T}_h^* Q_{h+1}^k](x_h^k, a_h^k) - Q_h^k(x_h^k, a_h^k)\right] \middle| x_1 = x_1^k\right] + \mathbb{E}_{\pi^k}\left[Q_1^k(x_1^k, a_1^k)\right] \\
&= \mathbb{E}_{\pi^k, \mathbb{P}}\left[\sum_{h=1}^H \left[[\mathcal{T}_h^* Q_{h+1}^k](x_h^k, a_h^k) - Q_h^k(x_h^k, a_h^k)\right]\right] + V_1^k(x_1^k).
\end{aligned}$$

□

We restate Theorem 1 of Dann et al. (2021) below.

Theorem A.5 (Theorem 1 of Dann et al. (2021))**.** Assume that parameter $\eta \le \frac{2}{5b^2}$ is set sufficiently small and that Assumption 4.2 holds. Then the expected regret (with Thompson sampling excuted exactly) after K episodes on any MDP M is bounded as

$$\mathbb{E}[\text{Regret}(K)] \le \frac{\lambda}{\eta}\mathcal{K}_{\text{DC}} + \frac{2K}{\lambda}\kappa(b/T^2) + \frac{6H}{\lambda} + \frac{b}{K},$$

where the expectation is over the samples drawn from the MDP and the algorithm's internal randomness.

A.3 Analysis of Sampling Error

Recalling the procedure for obtaining the Q function at episode k: we first sample w_H to obtain Q_H^k, and then for $h = H-1, \dots, 1$, we sample w_h conditional on Q_{h+1}^k to obtain Q_h^k. Let's denote the target conditional distribution of Q_h^k given Q_{h+1}^k as $q_{h,h+1}^k$. If Thompson sampling is executed precisely at each step, we should acquire the Q function $Q_h^k \sim q_k^h$ such that $q_k^h = q_{h,h+1}^k(\cdot|Q_{h+1}^k)$ for all $h \in [H]$. We denote the joint distribution of $\{q_k^h\}_{h\in[H]}$ as q_k. However, due to the high computational complexity, we can only obtain a sequence of $\{\widetilde{Q}_h^k \sim \widetilde{q}_k^h\}_{h\in[H]}$ that satisfies $\widetilde{q}_k^h = \widetilde{q}_{h,h+1}^k(\cdot|\widetilde{Q}_{h+1}^k)$, where $\widetilde{q}^k_{h,h+1}$ is a transition kernel close to $q^k_{h,h+1}$ and depends on the approximating sampler. We denote the joint distribution of $\{\widetilde{q}^h_k\}_{h\in[H]}$ as q'_k. Note that given $\widetilde{Q}^k_{h+1}$ generated by our algorithm, our goal is to sample from $q^k_{h,h+1}(\cdot|\widetilde{Q}^k_{h+1})$ while we can only obtain a sample close to that, denoted as $\widetilde{q}^h_k$. Our samplers can only control the distance (error) between $\widetilde{q}^k_{h,h+1}(\cdot|\widetilde{Q}^k_{h+1})$ and $q^k_{h,h+1}(\cdot|\widetilde{Q}^k_{h+1})$. We denote their total variation distance by δ^h_k. However, the sampling error in Theorem 4.6 is in terms of the distance between the joint distributions at episode k: $\delta_k = \mathrm{TV}(q_k, q'_k)$. Therefore, for a concrete analysis of sampling error, it is imperative to express δ_k in terms of $\{\delta^h_k\}_{h\in[H]}$. The expression relies on the following proposition:

Proposition A.6. Suppose that we have four random variables X_i and Y_i, $i = 1, 2$. Denote the conditional distribution of Y_i given $X_i = x$ by $p_i(\cdot|x)$, $i = 1, 2$. Let q_i be the joint distribution of (X_i, Y_i) and q^X_i be the distribution of X_i, $i = 1, 2$. If $\sup_x \mathrm{TV}(p_1(\cdot|x), p_2(\cdot|x)) \le \epsilon < \infty$, then

$$\mathrm{TV}(q_1, q_2) \le \mathrm{TV}(q^X_1, q^X_2) + \epsilon.$$

Proof of Proposition A.6. In this proof, we abuse notation by identifying a measure with its density for convenience. By definition of total variation distance,

$$\begin{aligned}
2\mathrm{TV}(q_1, q_2) &= \int |p_1(y|x)q^X_1(x) - p_2(y|x)q^X_2(x)|dydx \\
&\le \int |p_1(y|x)q^X_1(x) - p_2(y|x)q^X_1(x)|dydx + \int |p_2(y|x)q^X_1(x) - p_2(y|x)q^X_2(x)|dydx \\
&\le \int \left(\int |p_1(y|x) - p_2(y|x)|dy \right) q^X_1(x)dx + \int \left(\int p_2(y|x)dy \right) |q^X_1(x) - q^X_2(x)|dx \\
&= 2\int \mathrm{TV}(p_1(\cdot|x), p_2(\cdot|x))q^X_1(x)dx + \int |q^X_1(x) - q^X_2(x)|dx \\
&\le 2\epsilon + 2\mathrm{TV}(q^X_1, q^X_2),
\end{aligned}$$

where for the last equality, we use the fact that for any fixed x, $p_2(y|x)$ is a probability density and hence $\int p_2(y|x)dy = 1$. This concludes the proof. □

With this proposition in hand, we are ready to prove Proposition 4.11 which we restate here first.

Proposition A.7. Let δ^h_k be the sampling error (in the total variation sense) induced by our sampler at step h episode k and δ_k be defined in section 3.1, $h \in [H]$ and $k \in [K]$. Then $\delta_k \le \sum_{h=1}^H \delta^h_k$.

Proof of Proposition 4.11. Let $q^{h:H}_k$ be the joint distribution of $\{q^s_k\}_{h\le s\le H}$ and $\tilde{q}^{h:H}_k$ be the joint distribution of $\{\tilde{q}^s_k\}_{h\le s\le H}$. Then by Proposition A.6,

$$\delta_k = \mathrm{TV}(q_k, q'_k) \le \delta^1_k + \mathrm{TV}(q^{2:H}_k, \tilde{q}^{2:H}_k).$$

Likewise, for $h = 2, \dots, H-1$,

$$\mathrm{TV}(q^{h:H}_k, \tilde{q}^{h:H}_k) \le \delta^h_k + \mathrm{TV}(q^{h+1:H}_k, \tilde{q}^{h+1:H}_k).$$

Since $q^{H:H}_k = q^H_k$ and $\tilde{q}^{H:H}_k = \tilde{q}^H_k$, we have $\mathrm{TV}(q^{H:H}_k, \tilde{q}^{H:H}_k) = \delta^H_k$. Then we conclude the proof by combining the above inequalities. □

Now let T be the truncation map

$$T(x) := \min\{x, b\}^+$$

used in Algorithm 1. Then conditional on $\tilde{Q}^k_{h+1}$, $\tilde{q}^h_k$ is now given by

$$\widetilde{q}^h_k = T_\#[\widetilde{q}^k_{h,h+1}(\cdot|\widetilde{Q}^k_{h+1})].$$

Since we assume that $Q_h^k \in \mathcal{Q}$ for all h and k, we have that $q_k^h = T_\#[q_k^h]$. And therefore the data-processing inequality gives

$$TV(\tilde{q}_k^h, q_k^h) = TV(T_\#[\widetilde{q}^k_{h,h+1}(\cdot|\widetilde{Q}^k_{h+1})], T_\#[q_k^h]) \leq TV(\widetilde{q}^k_{h,h+1}(\cdot|\widetilde{Q}^k_{h+1}), q_k^h),$$

which is exactly the sampling error in Proposition A.7. And hence the conclusion in the proposition still holds for δ_k with the truncation error.

A.4 Proof of Linear MDPs

In this section, we prove some properties for linear MDPs.

Proposition A.8. In linear MDPs, the linear function class $\mathcal{Q}$ satisfies Assumption 4.1 and 4.2 with $b = H$. And the decoupling coefficient is bounded by

$$\mathcal{K}_{\mathrm{DC}} \leq 2dH(1 + \ln(2KH)).$$

Proof of Proposition A.8. Note Assumption 4.1 can be verified directly by the definition of linear MDP. This boundedness follows from the fact that $r_h \in [0,1]$ for all $h \in [H]$ and hence $Q_h(x,a) \leq H - h + 1 \leq H$ for any $x, a \in \mathcal{S} \times \mathcal{A}$. Since Q_h is arbitrary here, we have $b = H$. For the upper bound of the decoupling coefficient, we refer to (Dann et al., 2021, Proposition 1). □

Proposition A.9. In linear MDPs, the linear function class $\mathcal{Q}$ satisfies Assumption 4.3. And given any state $x \in \mathcal{S}$ and $h \in [H]$, we have the following representation of $Q \in \mathcal{Q}$:

$$Q_h(x,a) - [\mathcal{T}_h^* Q_{h+1}](x,a) = \langle u_h, \phi(x,a)\rangle.$$

Proof of Proposition A.9. The linearity of the action-value functions directly follows from the Bellman equation:

$$Q_h(x,a) = r_h(x,a) + (\mathbb{P}_h V_{h+1})(x,a) = \langle \phi(x,a), \theta_h\rangle + \int_{\mathcal{S}} V_{h+1}(x')\langle \phi(x,a), d\mu_h(x')\rangle.$$

And likewise

$$[\mathcal{T}_h^* Q_{h+1}](x,a) = \langle \phi(x,a), \theta_h\rangle + \int_{\mathcal{S}} \max_{a'\in\mathcal{A}} Q_{h+1}(x',a')\langle \phi(x,a), d\mu_h(x')\rangle.$$

Then the completeness follows by defining

$$Q_h(x,a) = \langle \phi(x,a), \theta_h + \int_{\mathcal{S}} \max_{a'\in\mathcal{A}} Q_{h+1}(x',a') d\mu_h(x')\rangle.$$

And therefore

$$Q_h(x,a) - [\mathcal{T}_h^* Q_{h+1}](x,a) = \langle \phi(x,a), u_h\rangle,$$

where $u_h = \int_{\mathcal{S}} (V_{h+1}(x') - \max_{a'\in\mathcal{A}} Q_{h+1}(x',a')) d\mu_h(x')$. □

Next, we restate Lemma 4.15 and provide proof for it.

Lemma A.10. If the stage-wise priors p_0^h are chosen as $\mathcal{N}(0, \sqrt{d}HI_d)$, then $\kappa(\epsilon) = dHO(\ln(dH/\epsilon))$.

Proof of Lemma 4.15. By our choice of p_0^h,

$$p_0^h(\mathcal{Q}_h(\epsilon, Q_{h+1})) = O(\epsilon^d (2\pi\sqrt{d}H)^{d/2}).$$

And hence

$$\ln\left(\frac{1}{p_0^h(\mathcal{Q}_h(\epsilon, Q_{h+1}))}\right) = O(d\ln(1/\epsilon) + d\ln(dH)) = dO(\ln(dH/\epsilon)).$$

And finally,

$$\kappa(\epsilon) = \sum_{h=1}^{H} \ln\left(\frac{1}{p_0^h(\mathcal{Q}_h(\epsilon, Q_{h+1}))}\right) = dHO(\ln(dH/\epsilon)).$$

□

B More Details on Approximate Samplers

In this section, we provide details of samplers with the target distribution $\mu \propto e^{-L}$ in $\mathbb{R}^d$, where L is twice continuously differentiable, satisfying the conditions $mI_d \preccurlyeq \nabla^2 L \preccurlyeq MI_d$ for some $M \geq m > 0$. We also define the condition number of μ by $\kappa = \frac{M}{m}$.

B.1 Langevin Monte Carlo (LMC)

The Langevin Monte Carlo (LMC) algorithm samples from a target distribution $\mu \propto \exp(-L)$, using a discretized version of the continuous-time Langevin diffusion. Given an initial distribution p_0, and a step size $\tau > 0$, LMC generates a Markov chain $\{w_n\}_{n=0}^N$, starting from $w_0 \sim p_0$ where $w_n \in \mathbb{R}^d$. At each iteration $(n+1)$, the chain updates its state, w_n, using:

$$w_{n+1} = w_n - \tau\nabla L(w_n) + \sqrt{2\tau}\xi_n,$$

where ξ_n's are samples from the d-dimensional standard Gaussian independent of w_n. N is the total number of iteration. The convergence of LMC to the target distribution is a well-known result, holding true under specific assumptions (Chewi et al., 2022). For illustration, The following theorem provides a concrete example with sufficient conditions for convergence, which establishes the theoretical foundation for our analysis.

Theorem B.1 (Vempala & Wibisono (2019))**.** Denote the distribution of w_n by p_n. For any $\epsilon \in [0, \kappa\sqrt{d}]$, if we take $\tau = O(\frac{\epsilon^2}{\kappa M d})$, then we obtain the guarantee $TV(p_N, \mu) \leq \epsilon$ after

$$N = \tilde{\Theta}\left(\frac{\kappa^2 d}{\epsilon^2}\right) \quad \text{iterations.}$$

B.2 Underdamped LMC

The underdamped Langevin dynamics (ULD) for $(w_t, P_t) \in \mathbb{R}^{2d}$ is driven by the SDE

$$\begin{aligned} dw_t &= P_t dt, \\ dP_t &= -\nabla L(w_t)dt + \gamma P_t dt + \sqrt{2\beta^{-1}\gamma} dB_t, \end{aligned} \tag{B.1}$$

which can be viewed as a second-order Langevin dynamics. We can use different discretization schemes to obtain discrete-time algorithms. For ULMC using the scheme (3.3), we have the following convergence result:

Theorem B.2 (Zhang et al. (2023))**.** For ULMC, assume that our target distribution satisfies that $\mathbb{E}_\mu[\|\cdot\|] = m_1 < +\infty$, $\nabla L(0) = 0$ (without loss of generality) and $L(0) - \min L = \tilde{O}(d)$. Then if we set $\tau = \tilde{\Theta}(\frac{\epsilon m_1^{1/2}}{M d^{1/2}})$ and $\gamma = \Theta(\sqrt{M})$ with a warm start, the law of the N-th iterate of ULMC p_N satisfies

$$TV(p_N, \mu) \leq \epsilon \quad \text{after} \quad N = \tilde{\Theta}\left(\frac{\kappa^{3/2} d^{1/2}}{\epsilon}\right) \quad \text{iterations.}$$

C Experiment Details

In this section, we provide further details on our experiment and implementation.

C.1 N-Chain

We use $\phi_{\text{therm}}(s_t) = (\mathbf{1}\{x \le s_t\})$ in $\{0,1\}^N$ as input feature following Osband et al. (2016a). We further follow the same protocol as in Ishfaq et al. (2024) for our experiments. For all the baseline algorithms, as well as our proposed algorithms FG-LMCDQN, FG-ULMCDQN and ULMCDQN, we parameterize the Q function using a multi-layer perceptron (MLP). We use $[32, 32]$ sized hidden layers in the MLP and *ReLU* as the activation functions. All algorithms are trained for 10^5 steps where the experience replay buffer size is 10^4. The performance of each algorithm is measured by the mean return of the last 10 test episodes. We use mini-batch size of 32 and set discount factor as $\gamma_{\text{discount}} = 0.99$. The target network is updated every 100 steps.

For our proposed algorithms, we do a grid search for the hyper-parameters: learning rate τ, bias factor a in the optimizers, the temperature β, the friction coefficient γ and the Feel-Good prior weight η. We list the detailed values of all swept hyper-parameters in Table 2. Following Ishfaq et al. (2024), for the adaptive bias term, we set $\alpha_1 = 0.9$, $\alpha_2 = 0.99$, and $\lambda = 10^{-8}$ in (3.5).

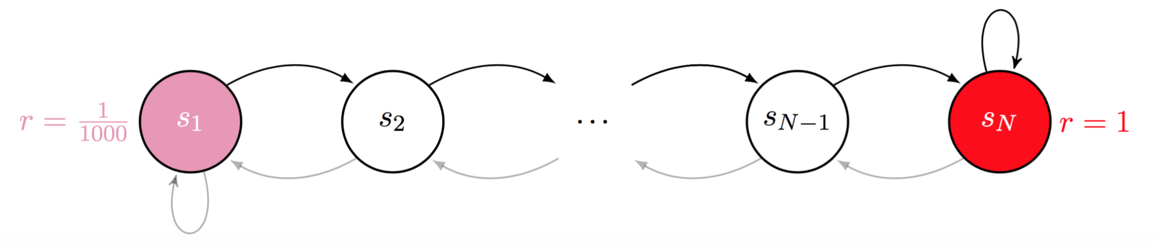


Figure 4: N-Chain environment (Osband et al., 2016a).

Hyper-parameter	Values
learning rate τ	$\{0.01, 0.001\}$
bias factor a	$\{1.0, 0.1, 0.01\}$
temperature β	$\{10^{12}, 10^{10}, 10^{8}\}$
update number J_k	$\{4\}$
friction coefficient γ	$\{1, 0.1, 0.01\}$
feel-good prior weight η	$\{1, 0.1, 0.01\}$

Table 2: The swept hyper-parameter in N-Chain experiments.

C.2 Atari

C.2.1 Experiment Setup

For our Atari experiments, our training and evaluation protocol follows that of Mnih et al. (2015); Ishfaq et al. (2024). We implement FG-LMCDQN, FG-ULMCDQN and ULMCDQN using the Tianshou framework (Weng et al., 2022).

To maintain a reasonable training time, for all our algorithms FG-LMCDQN, FG-ULMCDQN and ULMCDQN, we set $J_k = 1$. Similar to the N-chain experiments, following Ishfaq et al. (2024), for the adaptive bias term, we set $\alpha_1 = 0.9$, $\alpha_2 = 0.99$, and $\lambda = 10^{-8}$ in (3.5). We list the detailed values of all swept hyper-parameters in Table 3.

Hyper-parameter	Values
learning rate τ	$\{0.01,\ 0.001,\ 0.0001\}$
bias factor a	$\{0.1, 0.01\}$
temperature β	$\{10^{16}, 10^{14}, 10^{12}\}$
update number J_k	$\{1\}$
friction coefficient γ	$\{10,\ 1\}$
feel-good prior weight η	$\{1,\ 10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}\}$

Table 3: The swept hyper-parameter in Atari games.

C.2.2 Raw Scores in Atari

In Table 4, we directly compare the performance of our proposed algorithms FG-LMCDQN, FG-ULMCDQN and ULMCDQN with other baselines by presenting the maximal score obtained by each algorithm in each of the 8 Atari games. The results are averaged over 5 random seeds.

Methods\Games	Alien	Freeway	Gravitar	H.E.R.O	Pitfall!	Q*bert	Solaris	Venture
Human	7128	30	3351	30826	6464	13455	12327	1188
Random	228	0	173	1027	-229	164	1263	0
IQN	1691 ± 155	34 ± 0	413 ± 31	11229 ± 1098	-5 ± 4	14324 ± 643	1082 ± 258	3 ± 3
BootstrappedDQN	1067 ± 133	29 ± 1	312 ± 50	13538 ± 696	-53 ± 19	11786 ± 1474	923 ± 383	12 ± 12
C51	1878 ± 1878	34 ± 0	254 ± 52	**17500** ± **2170**	-19 ± 16	13394 ± 1500	303 ± 191	31 ± 31
PrioritizedDQN	1799 ± 202	23 ± 10	149 ± 37	14462 ± 1249	-46 ± 16	9464 ± 880	**1377** ± **423**	23 ± 23
NoisyNetDQN	1545 ± 217	32 ± 0	124 ± 47	6003 ± 1903	-31 ± 21	11046 ± 1594	1373 ± 361	24 ± 21
AdamLMCDQN	1772 ± 188	33 ± 0	799 ± 34	11366 ± 348	-6 ± 4	14628 ± 498	938 ± 189	**1326** ± **92**
ULMCDQN	1999 ± 687	34 ± 0	697 ± 64	15201 ± 3141	-3 ± 2	**14704** ± **794**	1195 ± 390	1132 ± 195
FG-LMCDQN	**2380** ± **645**	23 ± 10	744 ± 73	11576 ± 202	**0** ± **0**	14674 ± 436	735 ± 132	1069 ± 60
FG-ULMCDQN	2030 ± 446	**34** ± **0**	**844** ± **312**	14044 ± 1220	-9 ± 9	14385 ± 579	1278 ± 851	1198 ± 198

Table 4: Experiments results on 8 Atari Games. Table 4 presents the scores in each environment with 50M frames. For each environment, the algorithms perform 5 runs with random seeds. Then we average the scores for each game over 5 runs as the final result with a 95% confidence interval. We consider 6 baselines: IQN (Dabney et al., 2018), Bootstrapped DQN (Osband et al., 2016a), C51 (Bellemare et al., 2017), Prioritized DQN (Schaul et al., 2015), NoisyNet DQN (Fortunato et al., 2018) and AdamLMCDQN (Ishfaq et al., 2024). The scores for IQN, C51 and Prioritized DQN are taken from DQN Zoo (Quan & Ostrovski, 2020). The scores for Bootstrapped DQN, NoisyNet DQN and AdamLMCDQN are taken from `https://github.com/hmishfaq/LMC-LSVI` which is the official github repository of Ishfaq et al. (2024). We implemented our algorithms ULMCDQN, FG-LMCDQN and FG-UMLCDQN with Tianshou framework (Weng et al., 2022) in which we also use the double Q method (Van Hasselt et al., 2016). Our algorithms have demonstrated advantages in Alien, Freeway, Gravitor and Pitfall compared with baselines.