More Efficient Randomized Exploration for Reinforcement Learning via Approximate Sampling

Haque Ishfaq *[1,2], Yixin Tan * [3], Yu Yang [3], Qingfeng Lan [4], Jianfeng Lu [3], Rupam Mahmood [4], Doina Precup [1,2], Pan Xu [3]

[1]Mila – Quebec AI Institute [2]McGill University [3]Duke University [4]University of Alberta

Motivation

- Explorations techniques are crucial for an agent to be able to solve novel complex problems.
- Thompson sampling based on Laplace approximation is not a good estimation for the posterior distribution when the value function has more general forms than linearity.
- Sampling from a Gaussian distribution with general covariance matrix in high dimensional problem is computationally inefficient.

Highlights

- We propose a class of practical and efficient online RL algorithm Least-Squares Value Iteration with Approximate Sampling Exploration (LSVI-ASE) based on Feel-Good Thompson Sampling and various approximate sampling methods
- On high-level, LSVI-ASE only needs to perform noisy gradient descent updates for exploration.
- We theoretically prove that **LSVI-ASE** achieves a $\widetilde{\mathcal{O}}(dH^{3/2}\sqrt{T})$ regret under linear MDP settings, where d is the dimension of the feature mapping, H is the planning horizon, and T is the total number of steps.
- We provide extensive experiments on both N-chain environments and challenging Atari games that require deep exploration.

Algorithm

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

Input: feel-good prior weight η, step sizes $\{\eta_k > 0\}_k$, temperature β, friction coefficient γ.
Initialize $w^{1,0}$.
for episode $k = 1, 2, \ldots, K$ **do**
  Receive the initial state s_1^k.
  $w^{k,0} = w^{k-1,J_{k-1}}$
  for $j = 1, \ldots, J_k$ **do**
    Generate $w^{k,j}$ via an approximate sampling method
  end for
  $Q^k(\cdot,\cdot) \leftarrow Q(w^{k,J_k}; \phi(\cdot,\cdot))$
  for step $t = 1, 2, \ldots$ until end of episode **do**
    Take action $a_t^k \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} Q^k(s_t^k, a)$. Observe reward $r^k(s_t^k, a_t^k)$, get next state s_{t+1}^k.
  end for
end for

Feel-Good Thompson Sampling

- Define a general TD loss function

$$L_{\text{TD}}^k(w) = \sum_{\tau=1}^{k-1}\sum_{t=1}^{T}\Big[r(x_t^\tau, a_t^\tau) + \max_{a\in\mathcal{A}} Q^k(x_{t+1}^\tau, a) - Q(w; \phi(x_t^\tau, a_t^\tau))\Big]^2 + \lambda\|w\|^2$$

- We let $L_{\text{prior}}^k(w) = -\eta\sum_{\tau=1}^{k-1}\max_{a\in\mathcal{A}} Q(w; x_1^\tau, a)$, where L_{prior}^k is the Feel-Good exploration prior term.
- We use the overall loss function $L^k(w) = L_{\text{TD}}^k(w) + L_{\text{prior}}^k(w)$.

Langevin Monte Carlo for Reinforcement Learning

- Langevin Monte Carlo update:

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

  - It approximately samples from $\pi_k \propto \exp(-\beta L(w))$.
- It is computationally efficient due to
  - it only needs to sample ϵ_k from isotropic Gaussian $\mathcal{N}(0, I)$.
  - it only needs to perform noisy gradient descent updates.

Deep Q-Network with LMC Exploration

Algorithm 2 (Feel-Good) LMCDQN Update

$w^{k,0} = w^{k-1,J_{k-1}}, m^{k,0} = m^{k-1,J_{k-1}}, v^{k,0} = v^{k-1,J_{k-1}}$
for $j = 1, \ldots, J_k$ **do**
  $\epsilon^{k,j} \sim \mathcal{N}(0, I)$
  $w^{k,j} = w^{k,j-1} - \eta_k\Big(\nabla\widetilde{L}^k(w^{k,j-1}) + am^{k,j-1} \oslash \sqrt{v^{k,j-1} + \lambda_1\mathbf{1}}\Big) + \sqrt{2\eta_k\beta^{-1}}\epsilon^{k,j}$
  $m^{k,j} = \alpha_1 m^{k,j-1} + (1-\alpha_1)\nabla\widetilde{L}^k(w^{k,j-1})$
  $v^{k,j} = \alpha_2 v^{k,j-1} + (1-\alpha_2)\nabla\widetilde{L}^k(w^{k,j-1}) \odot \nabla\widetilde{L}_h^k(w_h^{k,j-1})$
end for

Underdamped Langevin Monte Carlo for Reinforcement Learning

- Underdamped Langevin Monte Carlo update:

$$w_{k+1} = w_k + \eta_k P_k,$$
$$P_{k+1} = P_k - \eta_k\nabla L(w_k) - \gamma\eta_k P_k + \sqrt{2\beta^{-1}\gamma\eta_k}\epsilon_k,$$

  where $\epsilon_k \sim \mathcal{N}(0, I)$, γ is the friction coefficient, η_k is the step size and β is the temperature.
- Underdamped LMC performs better in high-dimensional and poorly conditioned settigs.

Deep Q-Network with Underdamped LMC Exploration

Algorithm 3 (Feel-Good) Underdamped LMCDQN Update

$w^{k,0} = w^{k-1,J_{k-1}}, m^{k,0} = m^{k-1,J_{k-1}}, v^{k,0} = v^{k-1,J_{k-1}}, P^{k,0} = P^{k-1,J_{k-1}}$
for $j = 1, \ldots, J_k$ **do**
  $\epsilon^{k,j} \sim \mathcal{N}(0, I)$
  $m^{k,j} = \alpha_1 m^{k,j-1} + (1-\alpha_1)\nabla\widetilde{L}^k(w^{k,j-1})$
  $v^{k,j} = \alpha_2 v^{k,j-1} + (1-\alpha_2)\nabla\widetilde{L}^k(w^{k,j-1}) \odot \nabla\widetilde{L}^k(w^{k,j-1})$
  $P^{k,j} = (1-\gamma\eta_k)P^{k,j-1} + \eta_k\Big(\nabla\widetilde{L}^k(w_k) + am^{k,j-1} \oslash \sqrt{v_k + \lambda\mathbf{1}}\Big) + \sqrt{2\beta^{-1}\gamma\eta_k}\epsilon^{k,j}$
  $w^{k,j} = w^{k,j-1} - \eta_k P^{k,j}$
end for

Theoretical Results

Table 1. Regret upper bound for episodic, non-stationary, linear MDPs.

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

N-Chain Environment

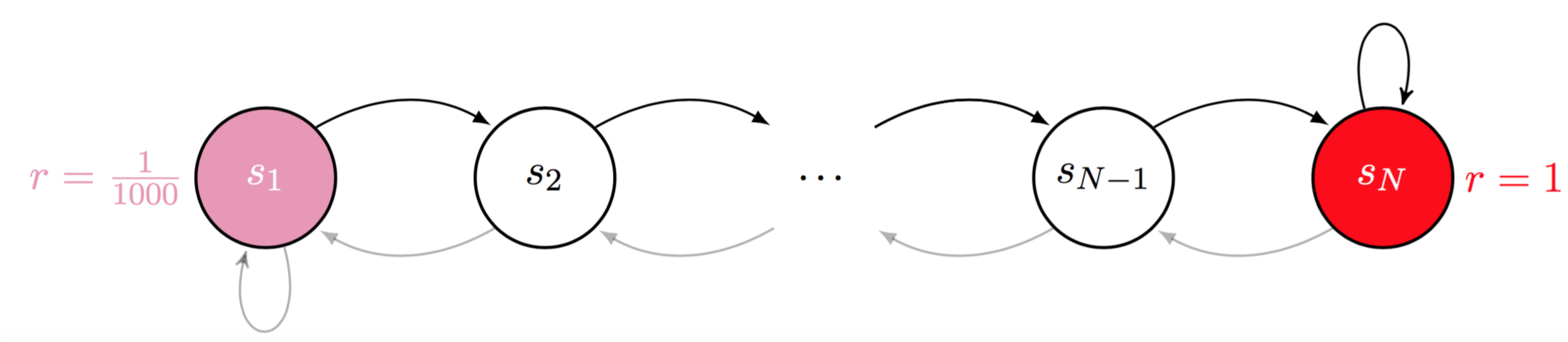

Figure 1. The N-Chain environment

N-Chain Experiments

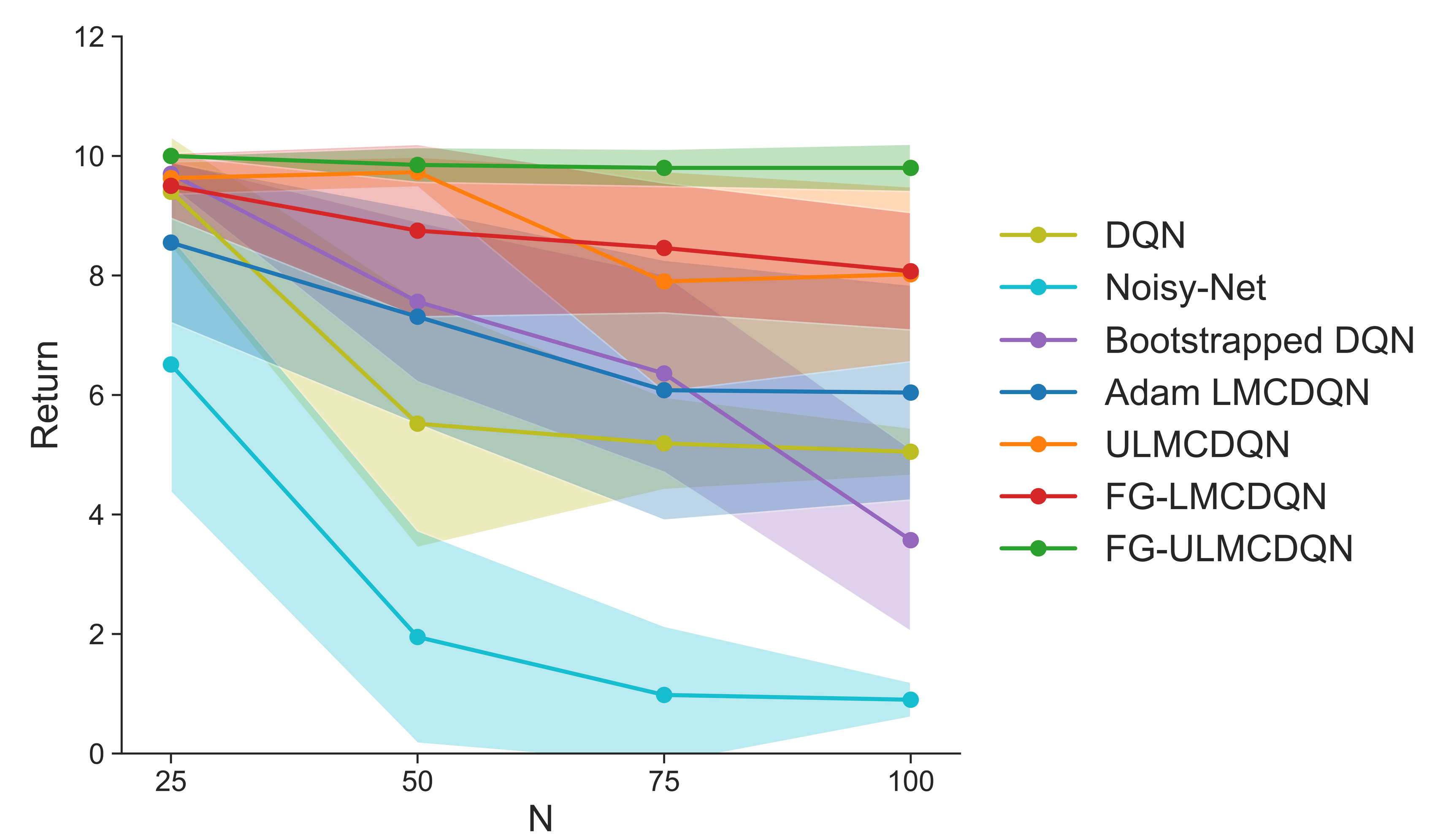

Figure 2. As N increases, the exploration hardness increases. All results are averaged over 20 runs and the shaded areas represent 95% confidence interval.

Atari Experiments

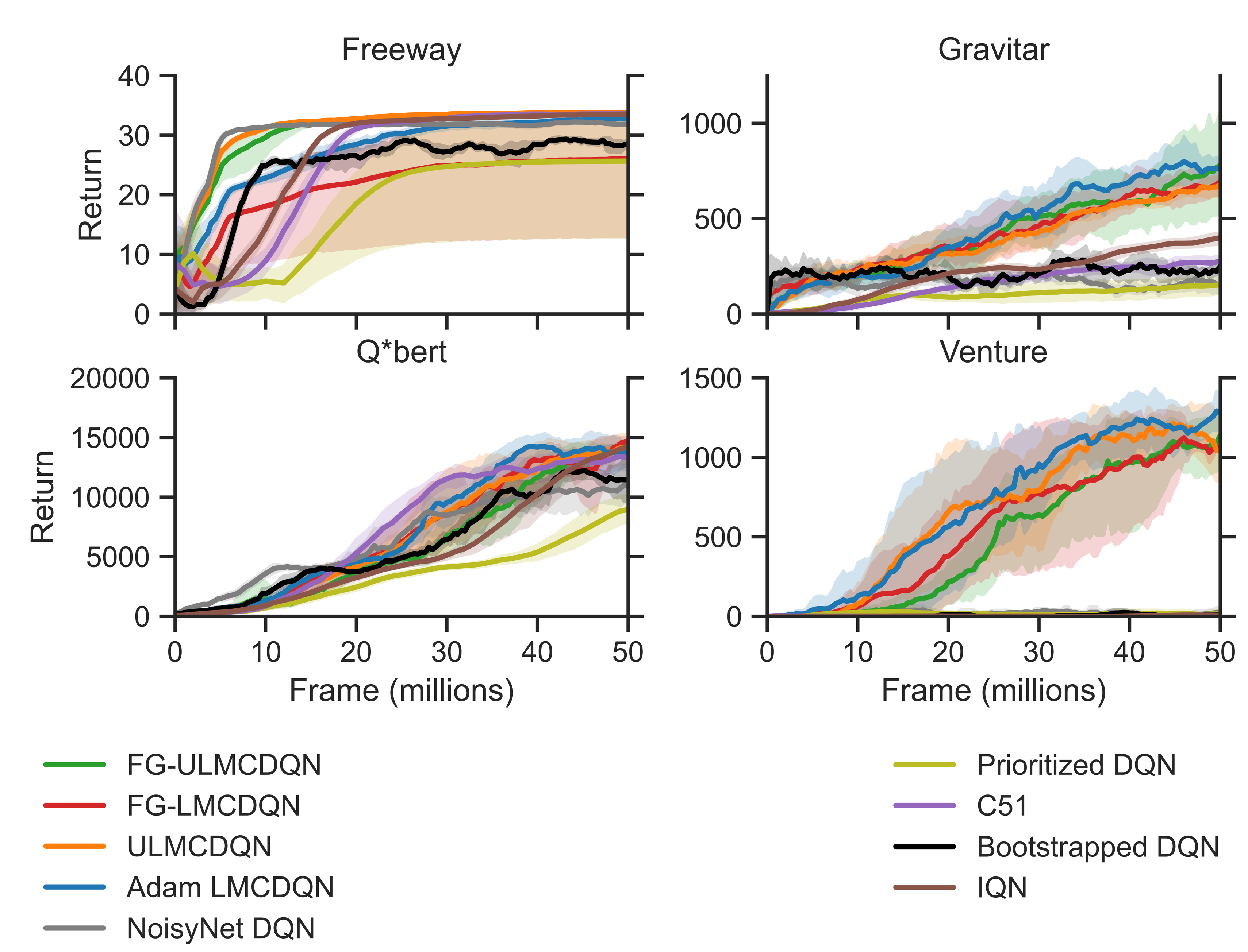

Figure 3. Return curves of various algorithms in Atari tasks over 50 million training frames. Solid lines correspond to the median performance over 5 random seeds, and the shaded areas correspond to 90% confidence interval.

haque.ishfaq@mail.mcgill.ca