# Revisiting Langevin Monte Carlo Applied to Deep Q-Learning: An Empirical Study of Robustness and Sensitivity

Pablo Hendriks Bardaji[1] supervised by Pascal van der Vaart[1], Neil Yorke-Smith[1]

[1]EEMCS, Delft University of Technology, The Netherlands

## Abstract

Deep Reinforcement Learning has achieved superhuman performance in many tasks, such as robotic control or autonomous driving. Algorithms in Deep Reinforcement Learning still suffer from a sample efficiency problem, where, in many cases, millions of samples are needed to achieve good performance. Recently, Bayesian uncertainty-based algorithms have gained traction. This work focuses on providing a better understanding of the behaviour of Langevin Monte Carlo algorithms for Bayesian posterior approximation applied on top of Q-learning. This research builds on top of already existing algorithms, aiming to provide a better understanding of the underlying mechanics that drive them. We provide empirical experimentation with different hyperparameters in three different environments. Our results suggest that hyperparameters that were previously thought not to have a big impact on the algorithms are crucial for deep exploration.

## Research Questions

1. How well does Adam LMCDQN generalise to simpler contextual bandit settings?
2. What is the effect of changing the number of updates per batch of data?
3. What are the effects of different hyperparameter choices for noise scale, the Adam weight in Adam LMCDQN and the learning rate?
4. What is the set of hyperparameters that generalises best across environments?

## Background

**(Deep) Q-Learning**: Q-learning is a model-free, value-based, off-policy reinforcement learning algorithm. In Q-learning, an agent learns the optimal policy by interacting with an environment. The Deep Q Network (DQN) algorithm first combined this approach with neural networks to approximate the Q-function [1].

An essential part of DQN, and Reinforcement Learning in general [2], is the exploration-exploitation dilemma; when to take the best action vs when to explore apparent sub-optimal actions with the hope of finding better rewards in the future.

**Posterior Sampling in Reinforcement Learning:** A line of research proposes using the posterior of the Q-function to update parameters. In each episode, parameters are sampled from the posterior, used for the rest of the episode, and acted optimally according to the policy specified by them.

Inferring the posterior is generally not feasible; to enjoy the different benefits that the posterior offers, algorithms need to rely on approximations of it. We focus on the approximation of this posterior via Langevin Monte Carlo.

## Langevin Monte Carlo

Langevin Monte Carlo is a Markov Chain Monte Carlo method, these methods are used to sample from distributions where direct sampling is difficult. The update rule is given by: $X_{t+1} = X_t - \eta_t \nabla L(X_t) + \sqrt{2\eta_t}\xi_t$.

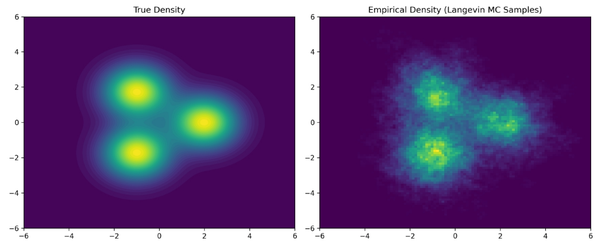


**Figure 1.** Samples from a Gaussian mixture using LMC

In our research, we focus on the Langevin Monte Carlo Least Squares Value Iteration (**LMC-LSVI**) algorithm [3], and the Adam Langevin Monte Carlo Deep Q Network (**Adam LMCDQN**) algorithm [3], a variant of LMC-LSVI based on Adam SGLD [4].

## Results

**Bandit Settings:** The Adam LMCDQN algorithm exhibits good adaptation to Bandit Settings while LMC-LSVI does not perform that well.

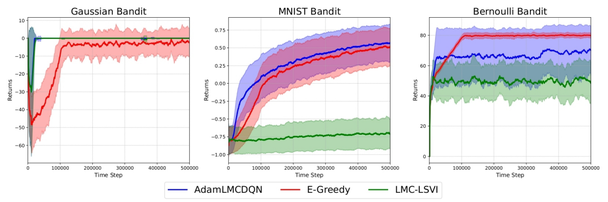


**Figure 2.** Bandit Returns for LMC-LSVI, Adam LMCDQN and $\varepsilon$-greedy.

**Number of Updates:** The number of updates per batch of data (J) does not improve performance in the Adam LMCDQN algorithm, while for LMC-LSVI, we find inconclusive results about whether it improves performance.

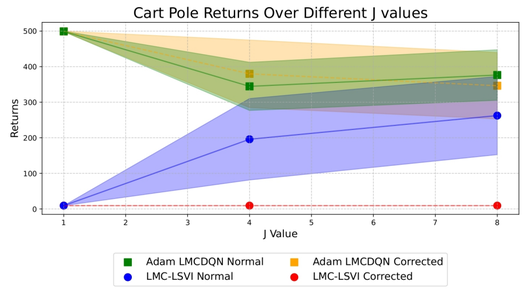


**Figure 3.** Cart Pole Returns for different values of J for the Adam LMCDQN and the LMC-LSVI algorithms.

**Noise scale, Adam weight and Learning rate:** Our results suggest that the inverse temperature (which controls the noise scale) is a crucial hyperparameter for achieving deep exploration. We also observe that the Adam weight ($a$) is a very important parameter in different kinds of environments. The learning rate seems to have some effect in LMC-LSVI, while Adam LMCDQN shows robustness to different values.

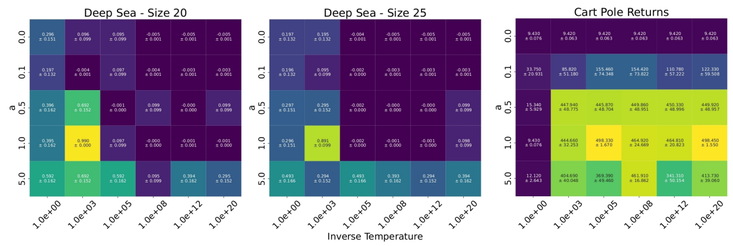


**Figure 4.** Returns in the Deep Sea environment of sizes 20 and 25 (left and middle), and in the Cart Pole environment (right) for different values of a and inverse temperature.

**Most Robust Set of Hyperparameters:** Based on the results of the previous questions, we propose a set of hyperparameters which we hypothesise will generalise well to a lot of environments. We highlight that this set of hyperparameters is proposed to be a starting point rather than a silver bullet that will work on all environments.

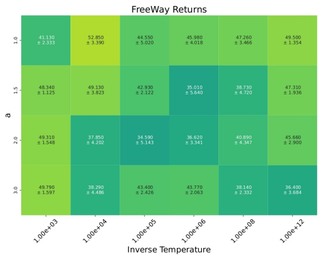


**Figure 5.** Results of the proposed set of hyperparameters in the Freeway MinAtar environment with the Adam LMCDQN algorithm.

## GitHub

## References

[1] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *ArXiv*, vol. abs/1312.5602, 2013.

[2] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction* (Adaptive computation and machine learning), en, Second edition. Cambridge, Massachusetts London, England: The MIT Press, 2020, p. 1, ISBN: 978-0-262-03924-6.

[3] H. Ishfaq *et al.*, "Provable and practical: Efficient exploration in reinforcement learning via langevin monte carlo," in *International Conference on Learning Representations*, 2024.

[4] S. Kim, Q. Song, and F. Liang, *Stochastic gradient langevin dynamics algorithms with adaptive drifts*, 2020. arXiv: 2009.09535 [stat.ML].