Proceedings of Machine Learning Research vol 242:1–14, 2024

Robust Exploration with Adversary via Langevin Monte Carlo

Hao-Lun Hsu HAO-LUN.HSU@DUKE.EDU
Department of Computer Science
Duke University

Miroslav Pajic MIROSLAV.PAJIC@DUKE.EDU
Department of Electrical and Computer Engineering
Duke University

Abstract

In the realm of Deep Q-Networks (DQNs), numerous exploration strategies have demonstrated efficacy within controlled environments. However, these methods encounter formidable challenges when confronted with the unpredictability of real-world scenarios marked by disturbances. The optimization of exploration efficiency under such disturbances is not fully investigated. In response to these challenges, this work introduces a versatile reinforcement learning (RL) framework that systematically addresses the intricate interplay between exploration and robustness in dynamic and unpredictable environments. In particular, we propose a robust RL methodology, framed within a two-player max-min adversarial paradigm; this formulation is cast as a Probabilistic Action Robust Markov Decision Process (MDP), grounded in a cyber-physical perspective. Our methodology capitalizes on Langevin Monte Carlo (LMC) for Q-function exploration, facilitating iterative updates that empower both the protagonist and adversary to efficaciously explore. Notably, we extend this adversarial training paradigm to encompass robustness against delayed feedback episodes. Empirical evaluation, conducted on benchmark problems such as *N-Chain* and *deep brain stimulation*, underlines the consistent superiority of our method over baseline approaches across diverse perturbation scenarios and instances of delayed feedback.

Keywords: Reinforcement Learning, Langevin Monte Carlo, Game Theory.

1. Introduction

Reinforcement Learning (RL) has shown great promise in decision-making problems across various domains, including games (Mnih et al., 2013; Silver et al., 2016; Goldwaser and Thielscher, 2020), robotics (Sorokin et al., 2022; Hsu et al., 2022; Smith et al., 2023), and healthcare (Gao et al., 2022b; Sarikhani et al., 2022; Gao et al., 2023). RL algorithms, such as DQN, have achieved success relying on exploration strategies such as ϵ-greedy (Mnih et al., 2013). However, recent works (Osband et al., 2016; Fortunato and Mohammad Gheshlaghi Azar, 2017; Ishfaq et al., 2023) have introduced more efficient exploration strategies that result in improved performance. While these methods work well under the assumption of fixed and identical reward and transition distributions according to the current state and the selected action (Lykouris et al., 2021), they may struggle in real-world scenarios with unforeseeable disturbances. Thus, it is critical to develop effective exploration methods that incorporate robustness to systematically mitigate the sensitivity of the optimal policy in perturbed environments and thereby maintaining performance.

To address the challenges posed by external disturbances, we propose an RL method with robust exploration to maintain a high reward under perturbations in the action selection. We adopt a two-player adversarial framework, treating the adversary as the second agent in a zero-sum game, enhancing the robustness of the RL agent (Gu et al., 2019; Kamalaruban et al., 2020b; Pattanaik et al., 2017; Pinto et al., 2017; Zhang et al., 2021). This approach aligns with the principles of Robust Markov Decision Processes (R-MDP) (Bagnell et al., 2001; Iyengar, 2005; Nilim and Ghaoui, 2003) and is instantiated in frameworks like Robust Adversarial Reinforcement Learning (RARL), Noisy Robust Markov Decision Processes (NR-MDP), and Probabilistic Action Robust MDP (PR-MDP) (Pinto et al., 2017; Tessler et al., 2019).

In our proposed method, both the protagonist and adversary learn their Q-functions via Langevin Monte Carlo (LMC) for exploration. The iterative updates per step allow both agents to effectively explore in interaction with each other. In contrast to existing approaches (Vinitsky et al., 2020; Dong et al., 2023) that formulate their adversarial actions as a combination with the original execution in NR-MDP (Tessler et al., 2019) or only target specific entries in the action space in RARL (Pinto et al., 2017), our model considers the problem from a cyber-physical system perspective, allowing the attacker to potentially take over the execution completely with a certain probability in PR-MDP (Tessler et al., 2019). We extend our framework to handle delayed feedback, adding flexibility for real-world scenarios (Kuang et al., 2023).

We evaluate our method on the challenging exploration problem *N-Chain* (Osband et al., 2016) as well as a practical problem focused on treatment of Parkinson's disease patients using deep brain stimulation (Schmidt et al., 2023), comparing it with various exploration strategies under adversarial learning. Our results indicate that our method consistently generates more robust policies compared to baselines across different types of perturbations and delayed feedback.

1.1. Posterior Sampling in Reinforcement Learning

In value-based RL, for efficient exploration, posterior sampling introduces randomness into the value function via Gaussian noise (Strens, 2020). Randomized least-squares value iteration (RLSVI) with frequentist regret analysis was proposed for tabular MDPs (Russo, 2019; Xiong et al., 2022). RLSVI was enhanced with the reward perturbation and greedy execution on estimated state-action values for simplicity and computational ease (Ishfaq et al., 2021). However, Gaussian distribution in RLSVI may not always be a proper approximation of the true posterior (Ishfaq et al., 2023) and the good features are not always easily known (Li et al., 2021). Addressing these challenges, Adam LMCDQN (Ishfaq et al., 2023) introduced a gradient-based approximate sampling scheme through Langevin dynamics for posterior sampling in deep RL. Langevin dynamics for posterior sampling were also explored in the context of delayed feedback (Kuang et al., 2023), offline settings (Ishfaq et al., 2023) and multi-agent systems (Hsu et al., 2024b).

1.2. Robust Reinforcement Learning

Existing literature mainly considers the robust control problems from a control theory perspective (Zhou et al., 1996; Doyle et al., 2013). However, our focus narrows down to the domain of robust RL, particularly as it pertains to robust MDPs initially explored in the context of predefined uncertainty sets for environmental transitions (Bagnell et al., 2001; Iyengar, 2005; Nilim and Ghaoui, 2003). The prevailing approach to learning robust policies involves interpreting environmental changes as adversarial perturbations. This conceptualization naturally formulates a max-min problem, encompassing two agents: an agent tasked with achieving the original objectives (protagonist) and an agent responsible for generating disruptions (adversary). Noteworthy instances within this research paradigm include Robust Adversarial Reinforcement Learning (RARL) (Pinto et al.,

2017) and Noisy Robust Markov Decision Process (NR-MDP) (Tessler et al., 2019), which differ in their modeling of the adversary. Research within these frameworks has demonstrated that learning with a population of adversaries can notably enhance robustness for continuous control (Vinitsky et al., 2020; Dong et al., 2023; Hsu et al., 2024a). On the other hand, MixedNE-LD (Kamalaruban et al., 2020a) introduced a sampling perspective via Langevin dynamics in order to facilitate robustness learning.

1.3. Comparison to MixedNE-LD

While sharing the main idea with Stochastic Gradient Langevin Dynamics (SGLD) approach (Welling and Teh, 2011), MixedNE-LD introduces a variant of DDPG ((Lillicrap et al., 2019)), focusing on problems with a continuous action space. This adaptation involves two actor networks for protagonist and adversary policies, utilizing Langevin dynamics, while the critic is trained to estimate the Q-function of the joint policy. It is important to note that, in contrast, when addressing problems with discrete action spaces in our work, Langevin dynamics is directly applied to estimate the Q-function.

From a robust control framework perspective, our approach in the work formulates the problem as learning on a PR-MDP, focusing on uncertainties/disturbances in cyber-physical system framed as adversarial inputs. In contrast, MixedNE-LD adopts the NR-MDP framework, making a strong assumption that the overall effect of disturbances can be captured as a linear combination of the protagonist and adversary actions. Additionally, beyond adversarial learning in the action space, our algorithm extends to be robust against delayed feedback, and empirical results support the effectiveness of our method.

2. Robust Exploration with Adversary via LMC (REAL)

2.1. Problem Formulation for Adversarial Learning

We formulate our problem as learning on a Markov Decision Process (MDP), which is defined as a 6-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}^p, \mathcal{A}^a, \mathcal{P}, r, \gamma)$. Here, $\mathcal{S}$ denotes a finite state space, and $\mathcal{A}^p$ and $\mathcal{A}^a$ represent the sets of discrete actions that the agent (protagonist) and adversary can take, respectively. The transition function $\mathcal{P}$ models the transition to the next state based on the current state and the actions of both the protagonist and the adversary. The reward function r quantifies the reward for the protagonist, accounting for the additional impact of the adversary's action. In this zero-sum game framework, the reward function for the adversary is set to $-r$. The discounting factor, $\gamma \in [0, 1)$, is introduced to shape the temporal influence of future rewards.

For any set $\mathcal{K}$, we use $\Delta(\mathcal{K})$ to denote the set of all possible probability distributions on $\mathcal{K}$. The protagonist's and adversary's policies are represented by $\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A}^p)$ and $\pi_\phi : \mathcal{S} \to \Delta(\mathcal{A}^a)$, respectively, with θ and ϕ denoting their respective parameters. At each time step, t, s_t captures the state of the environment, while $a_t^p \in \mathcal{A}^p$ (and $a_t^a \in \mathcal{A}^a$) denotes the action taken by the protagonist (adversary, respectively). Finally, we use

$$R(\theta, \phi) \doteq \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t^p, a_t^a) \;\mid\; a_t^p \sim \pi_\theta(s_t), a_t^a \sim \pi_\phi(s_t)\big], \tag{1}$$

to represent the cumulative discounted reward that the agent π_θ can receive under the disturbance of the adversary π_ϕ.

The objective of adversarial training (two-player max-min game) for robustness (Pinto et al., 2017; Vinitsky et al., 2020) can be defined as

$$\max_{\theta \in \Theta} \min_{\phi \in \Phi} R(\theta, \phi), \tag{2}$$

where Θ and Φ are pre-defined parameter spaces for the agent and the adversaries. In this approach, the RL agent maximizes the worst-case performance under disturbance. In this work, we follow the Probabilistic Action Robust MDP (PR-MDP) framework (Tessler et al., 2019), which can be viewed as a zero-sum game between protagonist and adversary.

Definition 1 (PR-MDP (Tessler et al., 2019)) *Consider an MDP $\mathcal{M}$, and let π_θ and π_ϕ be policies of a protagonist and an adversary. The probabilistic joint policy $\pi_p^{mix}(\pi_\theta, \pi_\phi)$ of the corresponding PR-MDP is defined as $\pi_p^{mix}(a|s) \doteq (1-p) \cdot \pi_\theta(a|s) + p \cdot \pi_\phi(a|s)$.*

To obtain the optimal probabilistic robust policy, the solution involves solving the zero-sum game described by (2). The alternating update of ϕ and θ occurs in each module, with the adversary updated in lines 4 to 15 and the protagonist updated in lines 16 to 27 within the main algorithm outlined in Algorithm 1. In each iteration, episodes are executed to estimate the Q-functions for both the protagonist and adversary using exploration, as detailed in Algorithm 2, which will be discussed in more detail in the following subsection.

The collected data trajectories in the k^{th} episode, denoted as $\{s_h^k, a_h^k, r_h^k\}_{h \in [H]}$, are collected in both lines 13 and 25 of Algorithm 1. The actions in these trajectories are defined as

$$a_h^k = \begin{cases} \arg\max_{a \in A^a} Q_{h,a}^k(s_h^k, a) & \text{w. p. } \ p, \\ \arg\max_{a \in A^p} Q_{h,p}^k(s_h^k, a) & \text{w. p. } \ 1-p. \end{cases} \tag{3}$$

by considering $p \in [0, 1]$ as the probability of encountering adversarial activity in the PR-MDP.

2.2. Deep Q-Network with Robust Efficient Exploration

We now introduce our algorithm, Robust Exploration with Adversary via LMC (REAL). To effectively estimate the Q-function, we employ a variant of deep Q-networks (DQNs) (Mnih et al., 2013) known as Adam LMCDQN. This serves as the core RL algorithm for both our the protagonist and adversary. Adam LMCDQN demonstrates theoretical guarantees in linear settings and exhibits promising empirical results in single-agent learning within the deep RL domain (Ishfaq et al., 2023).

In particular, when the Q-function's function approximation is linear, the model approximation at timestep $h \in [H]$ and episode $k \in [K]$ is denoted by $Q_h^k(\cdot, \cdot) = \min\{\mu(\cdot, \cdot)^\top \omega_h^{k,J_k}, H-h-1\}$, where $\mu(\cdot, \cdot)$ represents a feature vector of the corresponding state-action pair. The Q-function is parameterized with ω_h^{k,J_k} at timestep h and episode k, incorporating the noise gradient descent on the loss function $L_h^k(\omega_h)$ for J_k updates, where $L_h^k(\omega_h)$ is defined as the difference between the target Q value and the current Q value over the whole $k-1$ episodes as follows:

$$L_h^k(w_h) \doteq \sum_{\tau=1}^{k-1} (\tilde{y}_h^\tau - Q(\omega_h; \mu(s_h^\tau, a_h^\tau))^2 + \lambda \|\omega_h\|_2^2; \tag{4}$$

here, $\tilde{y}_h^\tau \doteq r_h^\tau + max_{a \in A} Q_{h+1}^k(s_{h+1}^\tau, a)$, ω_h is the parameter of the Q function, depending on the protagonist or adversary, and $\|\omega_h\|_2^2$ with $\lambda > 0$ is the regularization term. Specifically, the gradient

Algorithm 1 Robust Exploration with Adversary via LMC (REAL)

Input: $\eta_{k,p}$: step size for updating the agent policy, $\eta_{k,a}$: step size for updating the adversary, inverse temperature β_k, smoothing factors α_1 and α_2, bias factor a, update number J_k

Output: $\widehat{\theta}$: parameter for the agent policy.

Randomly initialize $\theta_h^{1,0}$ and $\phi_h^{1,0}$ from appropriate distribution for $h \in [H]$, $J_0 = 0$, $m_h^{1,0} = 0$ and $v_h^{1,0} = 0$ for $h \in [H]$ and $k \in [K]$.
$i \leftarrow 0$, $\theta^t \leftarrow \theta$, $\phi^t \leftarrow \phi$
for Iteration $i = 0 : I - 1$ **do**
  {Update the adversary.}
  for episode $k = 1 : K$ **do**
    Receive the initial state s_1^k
    for step $h = H, H-1, ...1$ **do**
      $\phi_h^{k,0} = \phi_h^{k-1,J_{k-1}}, m_{h,a}^{k,0} = m_{h,a}^{k-1,J_{k-1}}, v_{h,a}^{k,0} = v_{h,a}^{k-1,J_{k-1}}$
      $\phi_h^{k,J_k}, m_{h,a}^{k,J_k}, v_{h,a}^{k,J_k} = aLMC(\phi_h^{k,0}, \nabla \tilde{L}_h^k(\phi_h^{k,0}), a, m_{h,a}^{k,0}, v_{h,a}^{k,0}, \eta_{k,a}, \beta_k, \alpha_1, \alpha_2)$
      $Q_{h,a}^k(\cdot,\cdot) \leftarrow Q(\phi_h^{k,J_k}; \mu(\cdot,\cdot))$
    end for
    for step $h = 1, 2, ...H$ **do**
      Take action a_h^k, observe reward r_h^k and next state s_{h+1}^k
    end for
  end for
  {Update the protagonist.}
  for episode $k = 1 : K$ **do**
    Receive the initial state s_1^k
    for step $h = H, H-1, ...1$ **do**
      $\theta_h^{k,0} = \theta_h^{k-1,J_{k-1}}, m_{h,p}^{k,0} = m_{h,p}^{k-1,J_{k-1}}, v_{h,p}^{k,0} = v_{h,p}^{k-1,J_{k-1}}$
      $\theta_h^{k,J_k}, m_{h,p}^{k,J_k}, v_{h,p}^{k,J_k} = aLMC(\theta_h^{k,0}, \nabla \tilde{L}_h^k(\theta_h^{k,0}), a, m_{h,p}^{k,0}, v_{h,p}^{k,0}, \eta_{k,p}, \beta_k, \alpha_1, \alpha_2)$
      $Q_{h,a}^k(\cdot,\cdot) \leftarrow Q(\theta_h^{k,J_k}; \mu(\cdot,\cdot))$
    end for
    for step $h = 1, 2, ...H$ **do**
      Take action a_h^k, observe reward r_h^k and next state s_{h+1}^k
    end for
  end for
end for
$\widehat{\theta} \leftarrow \theta^T$

descent update adheres to Langevin Monte Carlo (LMC) principles, introducing isotropic Gaussian noise in each update as

$$\omega_h^{k,j} = \omega_h^{k,j-1} - \eta_k \nabla L_h^k(\omega_h^{k,j-1}) + \sqrt{2\eta_k \beta_k^{-1}} \epsilon_h^{k,j}, \tag{5}$$

where η_k represents the step-size parameter, β_k stands for the inverse temperature parameter, and $\epsilon_h^{k,j}$ denotes an isotropic Gaussian random vector in $\mathbb{R}^d$, where $j \in [J_k]$.

LMC is replaced with Adam SGLD (Kim et al., 2020) in Adam LMCDQN (Ishfaq et al., 2023) due to the prevalent pathological curvature and saddle points in most deep neural networks. Within

Algorithm 2 Adam Langevin Monte Carlo $aLMC(\omega_h^{k,0}, \nabla\tilde{L}_h^k(\omega_h^{k,0}), a, m_h^{k,0}, v_h^{k,0}, \eta_k, \beta_k, \alpha_1, \alpha_2)$

for $j = 1, ..., J_k$ **do**
  $\epsilon_h^{k,j} \sim N(0, I)$
  $\omega_h^{k,j} = \omega_h^{k,j-1} - \eta_k \nabla\tilde{L}_h^k(\omega_h^{k,j-1}) + a m_h^{k,j-1} \oslash \sqrt{v_h^{k,j-1} + C_1\mathbf{1}} + \sqrt{2\eta_k\beta_k^{-1}}\epsilon_h^{k,j}$
  $m_h^{k,j} = \alpha_1 m_h^{k,j-1} + (1-\alpha_1)\nabla\tilde{L}_h^k(\omega_h^{k,j-1})$
  $v_h^{k,j} = \alpha_2 v_h^{k,j-1} + (1-\alpha_2)\nabla\tilde{L}_h^k(\omega_h^{k,j-1}) \odot \nabla\tilde{L}_h^k(\omega_h^{k,j-1})$
end for

Algorithm 2 (aLMC), $\nabla\tilde{L}_h^k(\omega_h^{k,j-1})$ represents an estimate of $\nabla L_h^k(\omega_h^{k,j-1})$ based on one mini-batch of data sampled from the replay buffer. The smoothing factors for the first and second moments of stochastic gradients are denoted by α_1 and α_2, respectively. Additionally, α serves as the bias factor, and C_1 is a small constant introduced to prevent zero-divisors. Note that in this context, $\odot$ and $\oslash$ represent the element-wise vector product and division, respectively. The term $v_h^{k,j}$ can be considered an approximator of the true second-moment matrix $\mathbb{E}(\nabla\tilde{L}_h^k(\omega_h^{k,j-1})\nabla\tilde{L}_h^k(\omega_h^{k,j-1})^\top)$, and the bias term $m_h^{k,j-1} \oslash \sqrt{v_h^{k,j-1} + C_1\mathbf{1}}$ can be interpreted as the rescaled momentum, which is isotropic near stationary points.

2.3. Deep Q-Network with Robustness to Delayed Feedback

We account for stochastic delays across episodes, where the trajectory generated in each episode is not immediately observable due to delays. The definition of episodic delayed feedback, as adopted in this work, is provided below.

Definition 2 (Episodic Delayed Feedback (Kuang et al., 2023)) *In each episode $k \in [K]$, the execution of a fixed policy π^k produces a trajectory $\{s_h^k, a_h^k, r_h^k\}_{h\in[H]}$. Such trajectory information, termed the feedback of episode k, is subject to a random delay denoted as τ_k, representing the time gap between the completion of the rollout in episode k and the time point at which its feedback becomes observable.*

The feedback $\{s_h^k, a_h^k, r_h^k\}_{h\in[H]}$ of an episode k can only be observed after the initiation of the $k + \tau_k$-th episode, indicating that the delayed version of the loss function used in Algorithm 1 effectively becomes

$$L_h^k(w_h) \doteq \sum_{\tau=1}^{k-1} \mathbb{1}_{\tau,k-1}(\tilde{y}_h^\tau - Q(\omega_h; \mu(s_h^\tau, a_h^\tau))^2 + \lambda\|\omega_h\|_2^2,$$

where $\mathbb{1}$ represents the indicator whether the previous history from episode τ to $k-1$ are observable.

3. Evaluations

In this section, we empirically evaluate the proposed method by validating the robustness of our method against existing baselines in two tasks: N-Chain and Parkinson's symptom suppression via Deep Brain Stimulation (DBS). Note that the deployed adversary model during evaluation is the same as the trained adversary model after convergence.

3.1. N-Chain

In our N-Chain experiments (Osband et al., 2016), we aim to demonstrate that Adam LMCDQN exhibits enhanced robustness under adversarial learning in comparison to existing baselines. The N-Chain environment comprises a chain of N states, with the RL agent starting from the second state and having the option to move left or right. The agent receives a small reward of $r = 0.001$ in the first state and a larger reward of $r = 1$ in the final state. The horizon length is $N + 9$, resulting in an optimal return of 10.

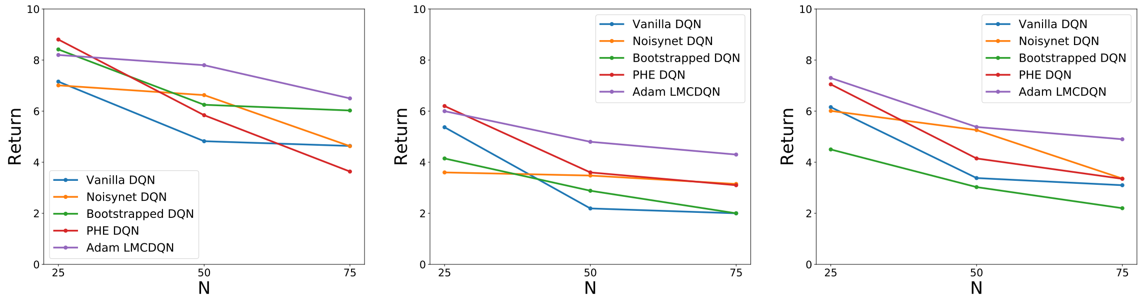
Despite the apparent simplicity of this environment, it presents a non-trivial challenge for exploration strategies. The propensity for the agent to become ensnared in the initial state, with its diminutive but immediate reward, accentuates the complexity of the task. Notably, as the chain length N increases, the exploration hardness also escalates. We compare our approach with several baselines, including vanilla DQN (Mnih et al., 2013), Bootstrapped DQN (Osband et al., 2016), Noisynet DQN (Fortunato and Mohammad Gheshlaghi Azar, 2017), and DQN with perturbed history exploration (PHE) as the exploration strategy (Ishfaq et al., 2021). We consider different numbers of states N; specifically, 25, 50, or 75.

Initially, we train all algorithms in the standard RL pipeline to establish the performance of Adam LMCDQN across different N (see Figure 1(a)). Bootstrapped DQN and PHE are competitive with $N = 25$, but their returns drop significantly when N increases. Given the simplicity of this environment with a discrete action space $\mathcal{A} = 2$, we set a small adversarial probability $p = 0.01$. We then evaluate the trained policies under the adversarial environment, where all methods experience a drop in return compared to the non-adversarial setting. However, Adam LMCDQN consistently outperforms other methods in general (see Figure 1(b)).

Finally, we proceed to train all methods under adversarial learning in PR-MDP with an adversarial probability $p = 0.01$, wherein the adversary tends to take over the action by moving left under the pre-defined probability. Adversarial training improves all exploration strategies in Figure 1(c) against Figure 1(b), and our proposed framework REAL based on Adam LMCDQN consistently exhibits robustness (denoted as "Adam LMCDQN" in Figure 1(c)). It is imperative to highlight that, in stark contrast, Bootstrapped DQN does not exhibit robustness to the adversarial attack, even with a chain length of $N = 25$, irrespective of whether it undergoes adversarial training or not. This observation holds for all subfigures in Figure 1. The performance of each algorithm is averaged over 10 seeds.

3.2. Deep Brain Stimulation

Deep brain stimulation (DBS) constitutes a surgical intervention aimed at alleviating motor symptoms by administering electrical pulses to the basal ganglia (BG) region of the brain (Benabid, 2003; Okun, 2012). The BG encompasses three primary sub-regions: the subthalamic nucleus (STN), globus pallidus pars externa (GPe), and globus pallidus pars interna (GPi). For a comprehensive understanding and quantification of Parkinson's disease (PD) manifestations, it becomes crucial to incorporate not only these principal sub-regions but also include the thalamic region (TH) and sensory-motor cortex (SMC) inputs within the PD-specific brain model, as illustrated in Figure 2. Assuming the presence of n neurons in each sub-region, the state emanating from the computational BG model at each time step t can be succinctly represented as a vector denoting electrical potential – i.e., $\boldsymbol{v}^q(t) = [\nu_1^q, ..., \nu_n^q]$, where $\nu_i^q(\cdot)$ signifies the i^{th} neuron in the corresponding sub-region



(*a*) Train(s), Test(s) (*b*) Train(s), Test(a) (*c*) Train(a), Test(a)

Figure 1: The comparison among all exploration strategies, including Adam LMCDQN, is conducted in N-Chain environment with varying chain lengths N. Different subfigures capture distinct training and testing conditions: (s) denotes standard setting without an adversary and (a) indicates setting under adversarial attack. Note that Adam LMCDQN in (c) with adversarial training is our proposed method (REAL). All results are averaged over 10 runs. Since the standard errors are not significantly different, they are not depicted.

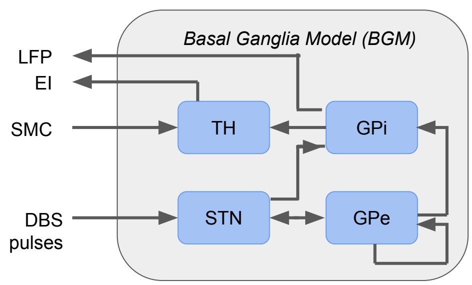


Figure 2: An illustration of the computational brain model (Jovanov et al., 2018). Deep Brain Stimulation (DBS) pulses are applied to the Subthalamic Nucleus (STN), with its effects propagating to other sub-regions. The Error Index (EI) is computed based on the activations passing from the sensorimotor cortex (SMC) to the thalamus (TH).

$q \in \{STN, GPe, GPi, TH\}$. The initial states of these neurons are treated as model parameters, stochastically determined within the experimental setup.

For the training and evaluation of RL methods in the context of DBS, a computational Basal Ganglia Model (BGM) (Jovanov et al., 2018) is cast as an OpenAI gym environment. Two essential metrics, namely Beta-band Power Spectral Density (P_β) and Error Index (EI), are introduced following the methodology outlined in (Gao et al., 2020). These metrics replace the direct observation of the entire electrical potential vector $\boldsymbol{v}^q(t)$. Specifically, P_β gauges the power spectral density of neuron potentials within the beta band for the GPi sub-region. Pathological oscillations of neurons in this band are indicative of Parkinson's disease. On the other hand, EI is defined as the percentage of erroneously activated neurons in the TH in response to inputs from the SMC. Note that the Error Index (EI) is constrained within the range $[0, 1]$, as it is defined as a ratio.

The objective of a DBS controller is to minimize the value of EI. While EI serves as an oracle for estimating the severity of Parkinson's disease symptoms, and the goal is to minimize its value, it is not accessible during training in practical scenarios. Consequently, unlike the reward function and states in (Gao et al., 2020, 2022a), we do not incorporate EI as a component of our reward function and states during the training phase. Instead, EI is solely considered as the final evaluation metric.

Following the formulated MDP detailed in Section 2.1, we model the dynamics of the neuron activities in the BGM. Specifically, the state $s_t \in \mathcal{S}$ is defined as the discretized sequence of past P_β signals. In essence, each state encompasses a sequence of P_β signals sampled periodically to facilitate improved training. In the computational BGM, the stimulus is executed once a pulse is triggered at that specific time point.

We define the action space for both the protagonist $\mathcal{A}^p$ and the adversary $\mathcal{A}^a$ in the MDP as a discrete action $a_t \in [1, 12]$ at time step t, representing the selected stimulus frequency. The maximum stimulus frequency is constrained to 180 Hz, and $F = 15i$ (for instance, when $i = 12$, the stimulus frequency reaches 180 Hz). The selected a_t is then mapped back to the stimulus for the BGM. To mitigate potential severe side effects arising from high-frequency stimulus (Beudel and Brown, 2016), the reward function is defined as $r(t) = -\bar{s}_{t+1} - C \cdot a_t$, where $\bar{s}_{t+1}$ denotes the average P_β over the entire sampling period. The second term of the reward function can be interpreted as a constant penalty $C \in \mathbb{R}$ on the frequency of the action a_t. Finally, it is important to note that determining a_t is influenced either by the protagonist or the adversary, depending on whether the protagonist is under attack during the time step t.

HYPERPARAMETER TUNING OF PENALTY COEFFICIENT

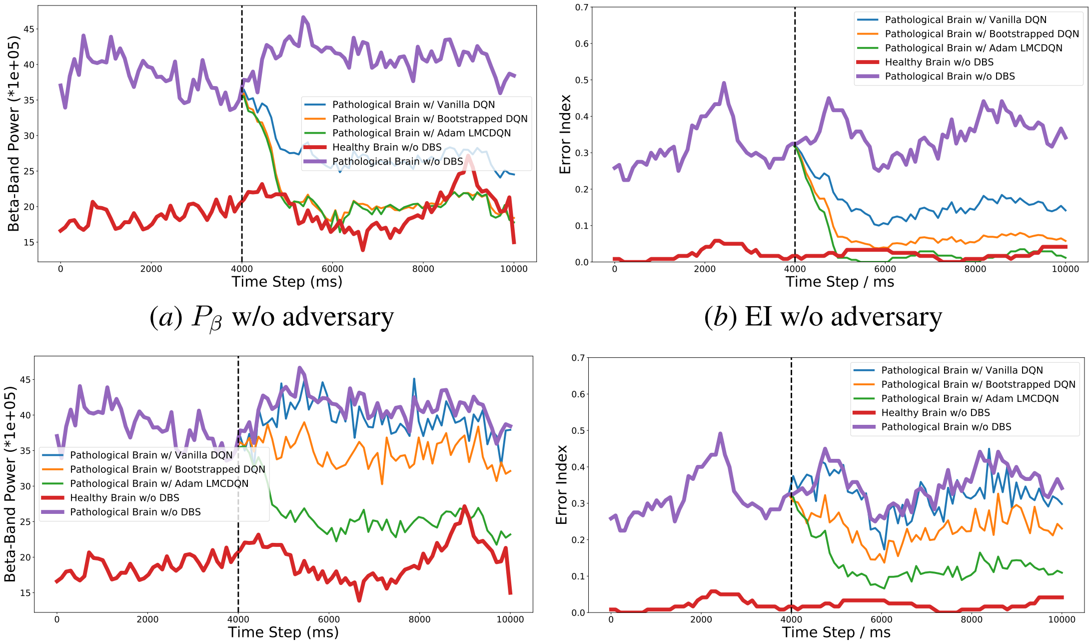
Our penalty coefficient C is subject to tuning within a specified search space. Considering that the value of the penalty coefficient C significantly impacts both the reward function and EI, our objective is to identify a suitable C that enables the learned policy to consistently maintain a low EI (below 0.1) (Gao et al., 2020) while employing a lower stimulation frequency with reduced energy consumption and side effects.

Inherent in this optimization is a trade-off between task performance and safety considerations. A higher stimulation frequency may be more effective in suppressing Parkinson's disease (PD) symptoms, while a larger C in the reward function discourages the policy from selecting a higher stimulation frequency to mitigate potential side effects. The primary objective is to choose the lowest average stimulation frequency while prioritizing effective task performance.

We evaluate three exploration strategies: vanilla DQN, Bootstrapped DQN (previously successful in N-Chain), and Adam LMCDQN. PHE and Noisynet DQN are omitted from the comparison due to scalability limitations (Ishfaq et al., 2023) and lower competitiveness in the N-Chain environment, respectively. To ensure a fair comparison, we tune the constant C within the range of $[0.09, 0.17]$ for all algorithms to achieve lower EI values.

PERFORMANCE OF THE PROPOSED METHOD – REAL

We initially train three exploration strategies without adversarial learning and evaluate them in the same environment. The results, along with those for the untreated PD brain and the healthy brain, are presented based on P_β and EI in Figure 3(*a*) and Figure 3(*b*). The entire evaluation period is demarcated by a dashed line, signifying the activation of all DBS controllers to produce their respective outputs after 4000 time steps. Consequently, excluding the healthy brain, all other controllers commence with the same oscillation characterized by higher P_β and EI.



(*a*) P_β w/o adversary (*b*) EI w/o adversary

(*c*) P_β w/ adversary (*d*) EI w/ adversary

Figure 3: P_β and EI over time in model PD brains without and with various types of stimulation, as well as in healthy brain. **First row:** training and testing **without** adversary. **Second row:** training and testing **with** adversary.

Adam LMCDQN demonstrates a superior trade-off between exploration and exploitation, resulting in lower P_β and EI values in the same environment. Subsequently, we conduct additional training for all exploration strategies under PR-MDP with $p = 0.1$ in Figure 3(*c*) and Figure 3(*d*). Notably, the learned adversary for each method represents its worst adversary, as we further learn the adversary π_ϕ after the convergence of the protagonist π_θ. Despite the increase in P_β and EI values for all variants of DQNs, Our REAL method, based on Adam LMCDQN (depicted in green) consistently maintains an EI value around 0.1, showcasing its efficacy as a DBS treatment.

Finally, an evaluation of the successfully trained Adam LMCDQN in an environment with episode delay following a Poisson distribution (Kuang et al., 2023) indicates that episode delay, viewed as a form of disturbance, could be effectively handled through the construction of varying episode delays during training, as outlined in Algorithm 1.

4. Conclusion

In this study, we have addressed the challenge of efficient exploration in the presence of unforeseeable adversaries or perturbations, specifically focusing on Deep Q-Networks (DQN) with discrete action space. We have assumed that the adversaries would follow PR-MDP formulation within a two-player zero-sum game framework. Both the protagonist and adversary use noisy gradient descent updates to approximate samples from the posterior distribution of the data, promoting exploration. Further, we have extended our adversarial learning framework to accommodate episodic delayed feedback, enhancing adaptability to more challenging scenarios. Finally, we have presented empirical results on an exploration problem, N-Chain, and a real-world application involving DBS.

Acknowledgments

This work has been sponsored in part by the ONR under agreement N00014-23-1-2206, AFOSR under the award number FA9550-19-1-0169, by the NIH UH3 NS103468 award, and by the NSF under the CNS-1652544 award as well as the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, Grant CNS-2112562.

References

J Andrew Bagnell, Andrew Y Ng, and Jeff G Schneider. Solving uncertain markov decision processes. *Citeseer*, 2001.

Alim Louis Benabid. Deep brain stimulation for parkinson's disease. In *Current opinion in neurobiology*, pages 696–706, 2003.

Martijn Beudel and Peter Brown. Adaptive deep brain stimulation in parkinson's disease. In *Parkinsonism & related disorders*, pages 123–126, 2016.

Juncheng Dong, Hao-Lun Hsu, Qitong Gao, Vahid Tarokh, and Miroslav Pajic. Robust reinforcement learning through efficient adversarial herding. *https://arxiv.org/abs/2306.07408*, 2023.

John C Doyle, Bruce A Francis, and Allen R Tannenbaum. Feedback control theory. *Courier Corporation*, 2013.

Meire Fortunato and et al. Mohammad Gheshlaghi Azar. Noisy networks for exploration. In *arXiv preprint arXiv:1706.10295*, 2017.

Qitong Gao, Michael Naumann, Ilija Jovanov, Vuk Lesi, K. Kumaravelu, Warren Grill, and Miroslav Pajic. Model-based design of closed-loop deep brain stimulation controllers using reinforcement learning. In *11th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 108–118, 2020.

Qitong Gao, Steven L. Schmidt, Karthik Kamaravelu, Dennis A. Turner, Warren M. Grill, and Miroslav Pajic. Offline policy evaluation for learning-based deep brain stimulation controllers. In *13th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 80–91, 2022a.

Qitong Gao, Dong Wang, Joshua D Amason, Siyang Yuan, Chenyang Tao, Ricardo Henao, Majda Hadziahmetovic, Lawrence Carin, and Miroslav Pajic. Gradient importance learning for incomplete observations. *International Conference on Learning Representations*, 2022b.

Qitong Gao, Stephen L Schmidt, Afsana Chowdhury, Guangyu Feng, Jennifer J Peters, Katherine Genty, Warren M Grill, Dennis A Turner, and Miroslav Pajic. Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pages 44–55, 2023.

Adrian Goldwaser and Michael Thielscher. Deep reinforcement learning for general game playing. *Proceedings of the AAAI conference on artificial intelligence*, 34:1701–1708, 2020.

Zhaoyuan Gu, Zhenzhong Jia, and Howie Choset. Adversary a3c for robust reinforcement learning. *arXiv preprint arXiv:1912.00330*, 2019.

Hao-Lun Hsu, Qiuhua Huang, and Sehoon Ha. Improving safety in deep reinforcement learning using unsupervised action planning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5567–5573, May 2022. doi: 10.1109/ICRA46639.2022.9812181.

Hao-Lun Hsu, Haocheng Meng, Shaocheng Luo, Juncheng Dong, Vahid Tarokh, and Miroslav Pajic. Reforma: Robust reinforcement learning via adaptive adversary for drones flying under disturbances. In *2024 International Conference on Robotics and Automation (ICRA)*, 2024a.

Hao-Lun Hsu, Weixin Wang, Miroslav Pajic, and Pan Xu. Randomized exploration in cooperative multi-agent reinforcement learning. *https://arxiv.org/abs/2404.10728*, 2024b.

Haque Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin F. Yang. Randomized exploration for reinforcement learning with general value function approximation. *International Conference on Machine Learning*, pages 4607–4616, 2021.

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 *https://arxiv.org/abs/2305.18246*, 2023.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Ilija Jovanov, Michael Naumann, Karthik Kumaravelu, Warren M. Grill, and Miroslav Pajic. Platform for model-based design and testing for deep brain stimulation. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 263–274, April 2018. doi: 10.1109/ICCPS.2018.00033.

Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8127–8138. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/5cb0e249689cd6d8369c4885435a56c2-Paper.pdf.

Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. *Advances in Neural Information Processing Systems*, 33:8127–8138, 2020b.

Sehwan Kim, Qifan Song, and Faming Liang. Stochastic gradient langevin dynamics algorithms with adaptive drifts. In *arXiv preprint arXiv:2009.09535*, 2020.

Nikki Lijing Kuang, Ming Yin, Mengdi Wang, Yu-Xiang Wang, and Yian Ma. Posterior sampling with delayed feedback for reinforcement learning with linear function approximation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=RiyH3z7oIF.

Ziniu Li, Yingru Li, Yushun Zhang, Tong Zhang, and Zhi-Quan Luo. Hyperdqn: A randomized exploration method for deep reinforcement learning. *International Conference on Learning Representations*, 2021.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

Thodoris Lykouris, Max Simchowitz, Alex Slivkins, and Wen Sun. Corruption-robust exploration in episodic reinforcement learning. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3242–3245. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/lykouris21a.html.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *arXiv preprint arXiv:1312.5602*, 2013.

Arnab Nilim and Laurent Ghaoui. Robustness in markov decision problems with uncertain transition matrices. *Advances in neural information processing systems*, 16, 2003.

Michael S Okun. Deep-brain stimulation for parkinson's disease. In *New England Journal of Medicine*, pages 1529–1538, 2012.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2017.

Daniel Russo. Worst-case regret bounds for exploration via randomized value functions. *Advances in neural information processing systems*, pages 14410–14420, 2019.

Parisa Sarikhani, Hao-Lun Hsu, and Babak Mahmoudi. Automated tuning of closedloop neuromodulation control systems using bayesian optimization. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1734–1737, 2022.

Stephen L Schmidt, Afsana H Chowdhury, Kyle T Mitchell, Jennifer J Peters, Qitong Gao, Hui-Jie Lee, Katherine Genty, Shein-Chung Chow, Warren M Grill, Miroslav Pajic, and Dennis A Turner. At home adaptive dual target deep brain stimulation in Parkinson's disease with proportional control. *Brain*, 147(3):911–922, 12 2023. ISSN 0006-8950. doi: 10.1093/brain/awad429. URL https://doi.org/10.1093/brain/awad429.

David Silver, Aja Huang, and Chris J. Maddison et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

Laura M. Smith, J. Chase Kew, Tianyu Li, Linda Luu, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Learning and adapting agile locomotion skills by transferring experience. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.051. URL https://doi.org/10.15607/RSS.2023.XIX.051.

Maks Sorokin, Jie Tan, C. Karen Liu, and Sehoon Ha. Learning to navigate sidewalks in outdoor environments. *IEEE Robotics and Automation Letters*, 7(2):3906–3913, 2022. doi: 10.1109/LRA.2022.3145947.

Malcolm Strens. A bayesian framework for reinforcement learning. In *International Conference on Machine Learning*, pages 943–950, 2020.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust reinforcement learning using adversarial populations, 2020.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, pages 681–688. PMLR, 2011.

Zhihan Xiong, Ruoqi Shen, Qiwen Cui, Maryam Fazel, and Simon Shaolei Du. Near-optimal randomized exploration for tabular markov decision processes. *Advances in neural information processing systems*, 2022.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021.

Kemin Zhou, John Comstock Doyle, and et al Keith Glover. Robust and optimal control. *Prentice hall New Jersey*, 40, 1996.