Accelerating Action-Robust Deep Deterministic Policy Gradient via Parallel Optimization

Extended Abstract

SeongIn Kim
University of Tsukuba
Tsukuba, Japan
kim@fz.iit.tsukuba.ac.jp

Takeshi Shibuya
University of Tsukuba
Tsukuba, Japan
shibuya@iit.tsukuba.ac.jp

ABSTRACT

The Noisy Action Robust MDP (NR-MDP) framework is effective for training robust policies but suffers from significant computational inefficiency due to the iterative, alternating training of a protagonist and an adversary. While reducing the frequency of adversary updates can shorten training time, it typically results in a degradation of robust performance. To address this trade-off, we propose Efficient Action Robust DDPG (EAR-DDPG), which eliminates the need for explicit adversarial policy learning by generating adversarial actions on-the-fly. We show that the approximately optimal adversarial action lies at a vertex of the hypercube. This insight allows us to replace conventional iterative optimization with instantaneous parallel computation over these vertex candidates.

KEYWORDS

Robust reinforcement learning; Adversarial learning; Parallel computing

ACM Reference Format:
SeongIn Kim and Takeshi Shibuya. 2026. Accelerating Action-Robust Deep Deterministic Policy Gradient via Parallel Optimization: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/VKQY3141

1 INTRODUCTION

Reinforcement Learning (RL) has achieved remarkable success in complex control tasks, ranging from robotics to autonomous driving [10]. However, standard RL relies on the assumption that training and deployment environments are identical. In real-world scenarios, environmental parameter mismatches—such as varying road friction due to weather conditions—often lead to severe performance degradation [2, 7, 13]. To address this, the Robust Markov Decision Process (R-MDP) framework formulates a max-min optimization problem in which a protagonist learns to maximize performance against an adversary that perturbs the environment. Among R-MDP variants, the Noisy Action Robust MDP (NR-MDP) [11] is particularly advantageous for real-world deployment, as it models perturbations on actions, which are physically easier to implement than manipulating state transitions. Also, various methods have been proposed to solve the NR-MDP optimization problem.[3, 4, 11].

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/VKQY3141

Despite its practical benefits, the NR-MDP framework suffers from significant computational inefficiency. Solving the max-min objective typically requires the iterative training of both a protagonist and an adversarial policy network. This extensive training time poses a challenge for "Green AI" [1, 8, 9, 12] and hinders rapid model deployment. Conventional acceleration methods, such as early stopping of the adversary, often fail to obtain an optimal adversary, resulting in a decline in the protagonist's robustness.

To resolve this trade-off between training efficiency and robust performance, we propose the Efficient Action Robust DDPG (EAR-DDPG). Unlike traditional approaches that learn a separate adversarial policy, EAR-DDPG solves the NR-MDP objective by generating optimal adversarial actions on-the-fly. We transform the adversary's policy-level optimization problem into a continuous action-level optimization problem and subsequently approximate it as a discrete optimization problem. This allows us to evaluate adversarial action candidates in parallel, thereby obtaining an approximately optimal adversarial action at each step. Experiments using MuJoCo demonstrate that the proposed method is effective in reducing training time while maintaining robust performance.

2 PRELIMINARIES

2.1 Markov Decision Process

An MDP [6] is defined as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ is the transition function, $R(\mathbf{s}_t, \mathbf{a}_t)$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. In this study, we consider an N-dimensional continuous action space defined as $\mathcal{A} = [-B, B]^N$.

Given a deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$, the action $\mathbf{a}_t = \pi(\mathbf{s}_t)$ is selected. The action-value function $Q_\pi(\mathbf{s}, \mathbf{a})$ is defined as:

$$Q_\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a} \right]$$

respectively. In reinforcement learning, the goal is to learn an optimal policy π^* that satisfies $\pi^*(\mathbf{s}) = \arg\max_{\mathbf{a}} Q^*(\mathbf{s}, \mathbf{a})$, where $Q^*(\mathbf{s}, \mathbf{a}) = \max_\pi Q_\pi(\mathbf{s}, \mathbf{a})$ holds for all $\mathbf{s} \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$.

2.2 Noisy Action Robust Markov Decision Process

An NR-MDP [11] is also defined by the same five components $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ as an MDP. In NR-MDPs, the protagonist's action $\mathbf{a} = \pi(\mathbf{s})$ is convexly combined with an adversarial action $\bar{\mathbf{a}} = \bar{\pi}(\mathbf{s})$. The resulting mixed action $\mathbf{a}^m$ is given by:

$$\mathbf{a}^m = (1 - \alpha)\mathbf{a} + \alpha\bar{\mathbf{a}},$$

Table 1: Wall-clock training time per 1,000 steps under Hopper-v4 environment.

AR-DDPG (10:1)	AR-DDPG (1:1)	Ours
3.67 sec	6.54 sec	3.69 sec

where $\alpha \in [0, 1]$ is a coefficient representing the strength of the perturbation. The mixed action $\mathbf{a}^m$ is applied to the environment to determine the reward.

Let $\pi_\alpha^m(\pi, \bar{\pi})$ denote the mixed policy that outputs $\mathbf{a}^m$ given state $\mathbf{s}$. In this framework, the protagonist policy π is trained to output an action which maximizes performance under the worst-case adversary policy $\bar{\pi}$, formulated as:

$$\pi^*(\mathbf{s}) = \arg\max_{\mathbf{a}} \min_{\bar{\pi}} Q_{\pi_\alpha^m(\pi^*,\bar{\pi})}(\mathbf{s}, (1-\alpha)\mathbf{a} + \alpha\bar{\pi}(\mathbf{s})).$$

The AR-DDPG algorithm, which is a DDPG-based method [5], learns the protagonist and the adversary by repeatedly solving the following optimization problems:

$$\pi_k = \arg\max_{\pi} Q_{\pi_\alpha^m(\pi,\bar{\pi}_k)}(\mathbf{s}, (1-\alpha)\pi(\mathbf{s}) + \alpha\bar{\pi}_k(\mathbf{s}))$$
$$\bar{\pi}_{k+1} = \arg\min_{\bar{\pi}} Q_{\pi_\alpha^m(\pi_k,\bar{\pi}_k)}(\mathbf{s}, (1-\alpha)\pi_k(\mathbf{s}) + \alpha\bar{\pi}(\mathbf{s})).$$

In this process, the robustness–performance trade-off is controlled by adjusting the update frequencies of the maximization (protagonist) and minimization (adversary) steps.

3 PROPOSED METHOD: EAR-DDPG

3.1 Overview

We propose EAR-DDPG (Efficient Action-Robust DDPG), an action-robust off-policy actor–critic algorithm that eliminates adversary policy learning. Instead of training an explicit adversary actor network, EAR-DDPG generates an approximately worst-case adversarial action on-the-fly at each visited state by solving a per-state constrained minimization of the learned critic.

3.2 Action-Level Reformulation of the Adversary Update

Conventional action-robust methods implement the inner minimization by learning an adversary policy $\bar{\pi}$ through repeated gradient updates. EAR-DDPG instead treats the inner minimization as a direct action selection problem. Given a state $\mathbf{s}$ and protagonist action $\mathbf{a} = \pi(\mathbf{s})$, define the adversary subproblem:

$$\bar{\mathbf{a}}^* = \arg\min_{\bar{\mathbf{a}} \in \mathcal{A}} Q_{\pi_\alpha^m(\pi_k,\bar{\pi}_k)}\Big(\mathbf{s},\ (1-\alpha)\pi_k(\mathbf{s}) + \alpha\bar{\mathbf{a}}\Big). \tag{1}$$

If we can compute $\bar{\mathbf{a}}^*$, then we no longer need an adversary actor network at all.

3.3 Local Quadratic Model of the Critic

Directly solving Equation (1) is a continuous box-constrained optimization, potentially expensive if done iteratively at every step. To enable fast computation, we analyze the local structure of $Q(\mathbf{s}, \cdot)$ near optimal action $\mathbf{a}^* = (1-\alpha)\pi_k(\mathbf{s}) + \alpha\bar{\pi}_k(\mathbf{s})$. Then, Equation (1) can be rewritten as

$$\arg\min_{\bar{\mathbf{a}} \in \mathcal{A}} Q_{\pi_\alpha^m(\pi_k,\bar{\pi}_k)}\Big(\mathbf{s},\ \mathbf{a}^* + \alpha(\bar{\mathbf{a}} - \bar{\pi}_k(\mathbf{s}))\Big). \tag{2}$$

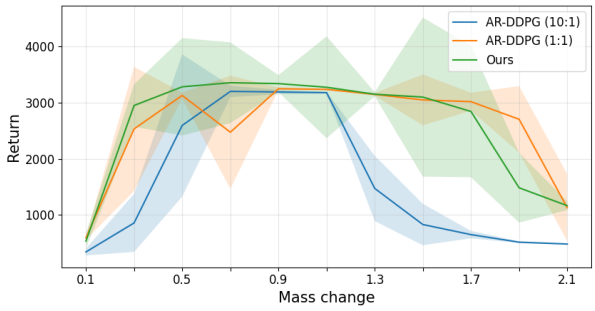

Figure 1: Robust performance heatmaps across environment parameter sweeps in Hopper-v4, comparing AR-DDPG (10:1), AR-DDPG (1:1), and EAR-DDPG.

Define $\bar{\mathbf{a}}' := (\bar{\mathbf{a}} - \bar{\pi}_k(\mathbf{s}))/2$. For sufficiently small α, Equation (2) can be approximated as

$$Q(\mathbf{s}, \mathbf{a}^* + 2\alpha\bar{\mathbf{a}}') \approx Q(\mathbf{s}, \mathbf{a}^*) + 2\alpha\mathbf{g}^\top\bar{\mathbf{a}}' + 2\alpha^2\bar{\mathbf{a}}'^\top H\bar{\mathbf{a}}', \tag{3}$$

where $\mathbf{g} = \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}^*)$ and $H = \nabla_{\mathbf{a}}^2 Q_{\pi_\alpha^m}(\mathbf{s}, \mathbf{a}^*)$. If $\mathbf{a}^*$ lies in the interior of $\mathcal{A}$, then $\mathbf{g} = \mathbf{0}$, so the second-order term becomes dominant and we treat Equation (3) as a convex function. In contrast, if $\mathbf{a}^*$ lies on the boundary of $\mathcal{A}$, the first-order term dominates and we treat Equation (3) as a linear function. For a linear objective over a hypercube, the optimum is attained at a vertex. For a convex objective over a hypercube, the optimum is also attained at a vertex.

Therefore, EAR-DDPG replaces continuous minimization with discrete search

$$\bar{\mathbf{a}}^* = \arg\min_{\bar{\mathbf{a}} \in \mathcal{V}} Q_{\pi_\alpha^m(\pi_k,\bar{\pi}_k)}\Big(\mathbf{s},\ (1-\alpha)\pi_k(\mathbf{s}) + \alpha\bar{\mathbf{a}}\Big),$$

where $\mathcal{V} = \{-B, +B\}^N$ and $|\mathcal{V}| = 2^N$.

4 EXPERIMENTS

4.1 Setup

We evaluate the proposed method on Hopper-v4 to focus on two practical questions: (i) does it maintain robust performance under body mass shifts (0.1 to 2.1times), and (ii) does it reduce wall-clock training time by eliminating adversary policy learning? We compared AR-DDPG with a 10:1 protagonist-to-adversary update ratio, AR-DDPG with a 1:1 update ratio, and EAR-DDPG. We set the action-perturbation coefficient to $\alpha = 0.1$ for all robust-training methods.

4.2 Results

The experimental results are presented in Table 1 and Figure 1. First, Table 1 shows that EAR-DDPG reduces the training time by approximately 44%. In addition, Figure 1 indicates that AR-DDPG (1:1) and EAR-DDPG achieve nearly the same level of robustness performance. Therefore, EAR-DDPG can substantially reduce training time while maintaining robustness performance comparable to that of AR-DDPG.

ACKNOWLEDGMENTS

This work was supported by JST SPRING, Grant Number JPMJSP2124.

REFERENCES

[1] Verónica Bolón-Canedo, Laura Morán-Fernández, Brais Cancela, and Amparo Alonso-Betanzos. 2024. A review of green artificial intelligence: Towards a more sustainable future. *Neurocomputing* 599 (2024), 128096.

[2] Yuchuan Du, Chenglong Liu, Yang Song, Yishun Li, and Yu Shen. 2019. Rapid estimation of road friction for anti-skid autonomous driving. *IEEE transactions on intelligent transportation systems* 21, 6 (2019), 2461–2470.

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

[4] SeongIn Kim and Takeshi Shibuya. 2024. Action Robust Reinforcement Learning with Highly Expressive Policy. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 5385–5390.

[5] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[6] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[7] Stefania Santini, Nicola Albarella, Vincenzo Maria Arricale, Renato Brancati, and Aleksandr Sakhnevych. 2021. On-board road friction estimation technique for autonomous driving vehicle-following maneuvers. *Applied sciences* 11, 5 (2021), 2197.

[8] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM* 63, 12 (2020), 54–63.

[9] Ghada Sokar, Elena Mocanu, Decebal Constantin Mocanu, Mykola Pechenizkiy, and Peter Stone. 2021. Dynamic sparse training for deep reinforcement learning. *arXiv preprint arXiv:2106.04217* (2021).

[10] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

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

[12] Pierre Thodoroff, Wenyu Li, and Neil D Lawrence. 2022. Benchmarking real-time reinforcement learning. In *NeurIPS 2021 Workshop on Pre-registration in Machine Learning*. PMLR, 26–41.

[13] Shizhe Zang, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar. 2019. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE vehicular technology magazine* 14, 2 (2019), 103–111.