Action-Level Backdoor Attacks against Deep Reinforcement Learning Systems via Adaptive Reward Exploration

Oubo Ma, Linkang Du, Yang Dai, Chunyi Zhou, Qingming Li, Yuwen Pu, Shouling Ji, *Senior Member, IEEE*

***Abstract*—Deep Reinforcement Learning (DRL) has demonstrated remarkable capabilities in domains such as robotics, finance, and autonomous systems. With the increasing cost of training, DRL models are increasingly shared and reused via model marketplaces, cloud platforms, and open-source repositories. This trend exposes DRL systems to action-level backdoor attacks, where adversaries inject malicious behaviors into models during the training phase and later manipulate their action outputs at deployment, potentially endangering safety-critical applications (e.g., causing vehicle collisions or drone crashes). Despite demonstrating feasibility, such attacks typically rely on handcrafted, task-specific backdoor rewards, which severely limit their generality and practicality, thereby causing practitioners to underestimate the true threat they pose. In this paper, we propose ADAPDOOR, the first general attack framework for adaptive backdoor reward exploration. ADAPDOOR coarsely initializes the backdoor reward from benign reward statistics and iteratively fine-tunes it based on the performance feedback from both benign and backdoor tasks. This enables stable action-level backdoor injection across diverse tasks without relying on extensive trial-and-error and expert knowledge. We conduct comprehensive experiments across 3 DRL algorithms, 11 environments, and 53 backdoor tasks, showing that ADAPDOOR outperforms existing baselines by 42.0% to 144.2%, under varying agent counts, action spaces, and reward signal types. Our findings reveal that action-level backdoors pose a more severe and realistic threat to DRL systems than previously recognized. Furthermore, we evaluate three potential defenses to explore pathways for mitigating this threat. The source code and task configurations of our evaluation are publicly available to ensure reproducibility and facilitate further research.**

***Index Terms*—Deep reinforcement learning, backdoor attacks, adaptive reward exploration, transition tampering.**

This work was partly supported by the New Generation Artificial Intelligence-National Science and Technology Major Project under No. 2025ZD0123503, NSFC under No. U2441239, U24A20336, 62402379, 62502432, 62502433, and 62402425, the China Postdoctoral Science Foundation under No. 2024M762829, 2025M781523, 2025M781522, and 2026M791724, Zhejiang Key Laboratory of Decision Intelligence under No. 2025E10006, Zhejiang Provincial Natural Science Foundation Exploration of China under No. LMS26F020003, State Key Laboratory of Cryptography and Digital Economy Security under No. KFYB2504, the Zhejiang Provincial Natural Science Foundation under No. LD24F020002, and the "Pioneer and Leading Goose" R&D Program of Zhejiang under No. 2025C02033 and 2025C01082. *(Corresponding author: Shouling Ji.)*

Oubo Ma, Chunyi Zhou, Qingming Li, and Shouling Ji are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310027, China (e-mail: mob@zju.edu.cn; zhouchunyi@zju.edu.cn; liqm@zju.edu.cn; sji@zju.edu.cn).

Linkang Du is with Xi'an Jiaotong University, Xi'an 710049, China (e-mail: linkangd@gmail.com).

Yang Dai is with National University of Defense Technology, Changsha 410072, China (e-mail: daiyang2000@163.com).

Yuwen Pu is with the School of Big Data & Software Engineering at Chongqing University, Chongqing 400044, China (e-mail: yw.pu@cqu.edu.cn).

This article has supplementary downloadable material available at xxx.

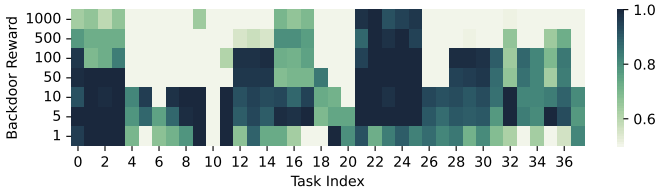


Fig. 1. The impact of backdoor reward and task variations on the attack effectiveness of action-level backdoors. Color intensity indicates the harmonic mean of the victim agent's performance on both benign and backdoor tasks.

I. INTRODUCTION

DEEP Reinforcement Learning (DRL) has achieved superhuman performance in various sequential decision-making tasks, such as robotic control [1], autonomous driving [2], and algorithmic trading [3]. As DRL agents grow more powerful and costly to train, pre-trained models are increasingly shared and reused on third-party platforms such as Hugging Face and Databricks. Against this backdrop, an adversary can release DRL models injected with backdoors. These backdoors remain dormant under normal conditions, but can be activated by particular triggers to hijack targeted actions, leading to severe consequences such as traffic congestion [4], vehicle collisions [5], or drone crashes [6].

Existing action-level backdoor attacks have demonstrated the feasibility of this adversarial vector [7]–[12]. However, these methods predominantly rely on manually designed, task-specific backdoor rewards, which require substantial domain expertise and fail to generalize across environments and tasks. In particular, fixed reward designs are ill-suited to the diversity and dynamics of real-world DRL systems, where reward signals, action spaces, and learning progress vary drastically. We validate this claim through the following empirical study and further identify the backdoor reward as a key catalyst in enabling effective action-level backdoor attacks.

Empirical Study. We adopt TrojDRL [7] as a representative DRL backdoor attack—where the backdoor reward is fixed at 1—to investigate how task characteristics affect the attack effectiveness. Specifically, we select four classic and two physics-based control tasks [13], and design 38 action-level backdoor tasks. Further implementation details of this empirical study are provided in Section V-A.

Fig. 1 shows that when the backdoor reward is fixed at 1, TrojDRL achieves a harmonic mean performance of 0.737 with a standard deviation of 0.207 across all benign and backdoor tasks. The results exhibit substantial variation in performance across tasks. For instance, it achieves 0.351 on `Task17` but reaches 0.999 on `Task19`. We further investigate the impact of varying the backdoor reward and identify two

key observations: (1) Even when the benign and backdoor tasks are fixed, varying the backdoor reward can introduce significant instability in attack performance. For instance, in `Task35`, increasing the backdoor reward from 1 to 5 improves the attack performance from 0.401 to 0.997, whereas further increasing it to 10 causes a drop to 0.831. (2) It is infeasible to predefine a backdoor reward that consistently yields stable attack performance across diverse DRL tasks.

Motivation. Building on the above observations, we find that existing methods for determining backdoor rewards often do not meet the practical attack requirements. Manual trial-and-error approaches, as used in the previous empirical study, are inefficient and lack scalability. *Therefore, this paper aims to enable cross-task adaptive backdoor reward exploration and to highlight that the threat posed by action-level backdoors to DRL systems has been significantly underestimated.*

Challenges. Achieving the aforementioned goals involves addressing the following three key challenges. *Task Discrepancy* **(C1)**: Substantial variations in Markov Decision Process (MDP) formulations across DRL tasks—particularly in benign reward structures—pose difficulties for generalized backdoor reward design. *Distraction Dilemma* **(C2)**: Jointly training benign and backdoor tasks within a shared policy model may lead to task interference, where one task dominates the learning process and suppresses the other. *Limited Trial-and-Error Search* **(C3)**: The inherent non-stationarity of DRL limits the feasibility of frequent backdoor reward tuning, as such adjustments may cause performance instability or even irreversible degradation in both benign and backdoor tasks.

Our Proposal. We propose ADAPDOOR, a novel attack framework for action-level backdoor injection. ADAPDOOR formulates the backdoor attack as a multi-task learning paradigm and enables the adaptive exploration of backdoor rewards. Specifically, ADAPDOOR comprises four main components: *Performance Monitoring*, *Initial Freezing*, *Backdoor Injection*, and *Adaptive Exploration*. *Performance Monitoring* tracks the agent's performance on both benign and backdoor tasks, and employs Exponential Weighted Averaging (EWA) and normalization to mitigate **C1**. *Initial Freezing* avoids the dominance of backdoor tasks during the agent's early training phase, thereby mitigating **C2**. Once *Initial Freezing* terminates, *Backdoor Injection* is performed to inject action-level backdoors, which supports both discrete and continuous action scenarios. In parallel, *Adaptive Exploration* first coarsely initializes the backdoor reward based on the statistical properties of the benign reward signals, and then iteratively refines it based on performance feedback. To address **C3**, the adversary adopts conservative exploration to limit the adjustment frequency.

Evaluations. We conduct extensive evaluations to demonstrate the effectiveness of ADAPDOOR across 3 mainstream DRL algorithms and 11 prominent environments (covering classic control, physics-based control, robotic, and multi-agent competition). A total of 53 backdoor tasks are designed to capture representative combinations of key factors—including agent counts, action spaces, reward signal types, and multi-backdoor configurations. Furthermore, we discuss potential defenses from multiple perspectives (e.g., state distribution, action drift, and reward reduction), exploring pathways to mitigate the threat posed by ADAPDOOR.

Contributions. This paper makes the following contributions:

- To the best of our knowledge, ADAPDOOR is the first general framework for adaptive backdoor reward exploration. It reveals that adversaries can achieve cross-task action-level backdoor injection without relying on extensive trial-and-error and expert knowledge.
- We observe that benign task performance is inversely correlated with the backdoor reward, whereas backdoor task performance is positively correlated. Based on this, we implement adaptive reward exploration guided by real-time performance monitoring.
- We conduct a comprehensive evaluation demonstrating that ADAPDOOR outperforms state-of-the-art baselines across diverse attack scenarios. To promote reproducibility, we release the source code and detailed backdoor designs at https://github.com/maoubo/ADAPDOOR.

II. BACKGROUND

A. Deep Reinforcement Learning

DRL is a machine learning paradigm in which an agent learns to make optimal sequential decisions by interacting with an environment and maximizing cumulative rewards. This process is typically modeled as an MDP, which is represented as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, which returns the immediate reward received when the agent takes action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition function, specifying the probability of transitioning to state $s' \in \mathcal{S}$ after taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. $\gamma \in [0, 1)$ is the discount rate, which determines the present value of future rewards. The agent makes decisions based on the deep neural network policy π_θ, where $\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A})$ maps state $s \in \mathcal{S}$ to a specific action or a probability distribution over the action space $\mathcal{A}$. As the agent interacts with the environment, it collects experiences in the form of trajectories $\tau = \{\tau_0, \tau_1, ..., \tau_T\}$, where each transition $\tau_t = (s_t, a_t, r_t)$ records the state, action, and immediate reward signal at time step t, and T denotes the time horizon. The goal of DRL is to learn the optimal parameters θ that maximize the expected cumulative reward, i.e., $\theta^* = \arg\max_\theta \mathbb{E}_{\tau \sim \pi_\theta}[\mathcal{G}(\tau)]$, where $\mathcal{G}(\tau) = \sum_{t=0}^{T} \gamma^t r_t$ is the discounted return.

B. DRL Backdoor Attacks

DRL backdoor attacks seek to force the victim agent to follow specific policies or actions. These attacks can be broadly classified into two main categories.

Policy-Level Backdoor. This type of backdoor focuses on coarse-grained manipulation of the victim agent, with each trigger mapping to a target policy. For instance, an adversary can activate the backdoor to redirect an autonomous driving agent's destination from a school to a hospital, regardless of the specific route taken. Attack techniques used to inject these backdoors include environment switching [14], policy combination [15], and trajectory poisoning [16].

Action-Level Backdoor. This type of backdoor focuses on fine-grained manipulation of the victim agent, with each trigger mapping to a target action. For instance, an adversary can activate the backdoor to force an autonomous driving agent to execute a sharp turn, causing traffic congestion or collisions. Transition tampering is the main injection technique for these backdoors [7], while finding in-distribution triggers [8], [9] and trigger optimization [10], [17], [18] improve their stealthiness and effectiveness. Rathbun *et al.* [11] employed Monte Carlo estimation to ensure that the target action appears theoretically optimal. However, we find that the attack performance of these methods is sensitive to the backdoor reward and requires task-specific, manual tuning to maintain effectiveness. This constraint limits their scalability and obscures the broader threat potential of action-level backdoors.

Appendix A provides a comprehensive comparison between policy-level and action-level backdoor attacks across six criteria. Motivated by the potential for precise manipulation and flexible activation, this work focuses on exposing the threat posed by action-level backdoors in DRL, with a particular emphasis on the pivotal role of backdoor rewards as a catalyst for attack effectiveness.

III. Threat Model

Aligned with current trends in DRL model commercialization and sharing—facilitated through open-source repositories, cloud platforms, and third-party model marketplaces—we consider a threat model where the adversary is the developer or distributor of the pre-trained DRL agents.

In modern DRL deployment ecosystems, model providers are often involved in downstream deployment, including runtime configuration and technical support. This integrated role inherently endows the adversary with the capabilities of a supply-chain insider threat. Leveraging this concomitant operational privilege during the deployment phase, the adversary can manipulate environmental interfaces or instrument sensory input pipelines to seamlessly insert malicious triggers that activate the target agent's action-level backdoor.

A backdoor attack is defined as $(\mathcal{T}, \mathcal{S}^\dagger, \mathcal{A}^\dagger, \mathcal{F}_s, \mathcal{F}_a, \mathcal{R}^\dagger)$, where $\mathcal{T}$ is the trigger space, $\mathcal{S}^\dagger \subseteq \mathcal{S}$ is the backdoor state space, and $\mathcal{A}^\dagger \subseteq \mathcal{A}$ is the target action space. $\mathcal{F}_s : \mathcal{S} \times \mathcal{T} \to \mathcal{S}^\dagger$ is a trigger-state mapping function used to insert a trigger into the original state. $\mathcal{F}_a : \mathcal{T} \to \mathcal{A}^\dagger$ is a trigger-action mapping function that defines the target action corresponding to each trigger. $\mathcal{R}^\dagger$ is a backdoor reward function designed to reinforce the association between triggers and their target actions.

Adversary's Capability. The adversary predefines $\mathcal{T}$, $\mathcal{A}^\dagger$, and $\mathcal{F}_a$, after which ADAPDOOR is invoked to generate $\mathcal{S}^\dagger$, $\mathcal{F}_s$, and $\mathcal{R}^\dagger$. ADAPDOOR automatically explores $\mathcal{R}^\dagger$ without requiring the adversary to perform extensive trial-and-error or possess expert knowledge. During the agent's training phase, the adversary is allowed to insert triggers into the states and manipulate the transition data stored by the agent. During the agent's deployment phase, the adversary can activate the backdoor by inserting triggers into the states, forcing the agent to execute the target actions. The above setting is aligned with the majority of previous work [7]–[9], while relaxing the assumption of manually specifying $\mathcal{R}^\dagger$.

Adversary's Objective. When the action space $\mathcal{A}$ is discrete, the adversary aims to maximize the probability that the backdoored policy $\pi_{\theta^\dagger}$ selects the target action $a^\dagger = \mathcal{F}_a(\delta)$ when a trigger $\delta \in \mathcal{T}$ is inserted into the input state:

$$\max_{\theta^\dagger} \mathbb{E}_{s\sim\mathcal{S},\delta\sim\mathcal{T}} \left[\pi_{\theta^\dagger}(a^\dagger|\mathcal{F}_s(s,\delta))\right]. \tag{1}$$

When the action space $\mathcal{A}$ is continuous, the adversary aims to steer the backdoored policy toward outputting actions that minimize the deviation from the target action when a trigger is inserted into the input state:

$$\min_{\theta^\dagger} \mathbb{E}_{s\sim\mathcal{S},\delta\sim\mathcal{T}} \left[||\pi_{\theta^\dagger}(\mathcal{F}_s(s,\delta)) - a^\dagger||\right]. \tag{2}$$

In addition, the adversary seeks to minimize the discrepancy in benign task performance between $\pi_{\theta^\dagger}$ and π_θ:

$$\min_{\theta^\dagger} \mathbb{E}_{s\sim\mathcal{S}\setminus\mathcal{S}^\dagger} \left[||U^{\pi_{\theta^\dagger}}(s) - U^{\pi_\theta}(s)||\right], \tag{3}$$

where $U^\pi(s)$ denotes the expected return when following the policy π from the state s to the terminal state.

IV. Methodology

This section outlines the overall workflow of ADAPDOOR and provides the design details of its key components.

A. Attack Workflow

Fig. 2 illustrates the workflow of ADAPDOOR, outlining the main roles and interactions of each component:

Performance Monitoring (Section IV-B). ADAPDOOR tracks the agent's performance on both benign and backdoor tasks during the training phase, referred to as the Benign Task Performance (BTP) and Attack Success Rate (ASR), respectively. Both indicators are processed via EWA and normalization to mitigate **C1**.

Initial Freezing (Section IV-C). ADAPDOOR postpones backdoor injection based on the number of trajectories and the monitored BTP to mitigate **C2**.

Backdoor Injection (Section IV-D). Once *Initial Freezing* terminates, ADAPDOOR probabilistically inserts triggers into the environment and tampers with the transition data.

Adaptive Exploration (Section IV-E). Simultaneously with *Backdoor Injection*, the backdoor reward is coarsely initialized based on the statistical characteristics of the benign reward signals, and then refined guided by monitored performance. To address **C3**, ADAPDOOR adopts conservative exploration to limit the adjustment frequency of the backdoor reward.

Backdoor Activation. During the deployment phase of the agent, the adversary can insert a trigger into the environment, thereby activating the backdoor and inducing the agent to execute the target action. The activation strategy is flexible and can be tailored to the adversary's practical requirements.

B. Performance Monitoring

ADAPDOOR conceptualizes an action-level backdoor attack as a multi-task learning problem and separately monitors the agent's performance on benign and backdoor tasks. Unlike conventional DRL multi-task learning [19], [20], both tasks

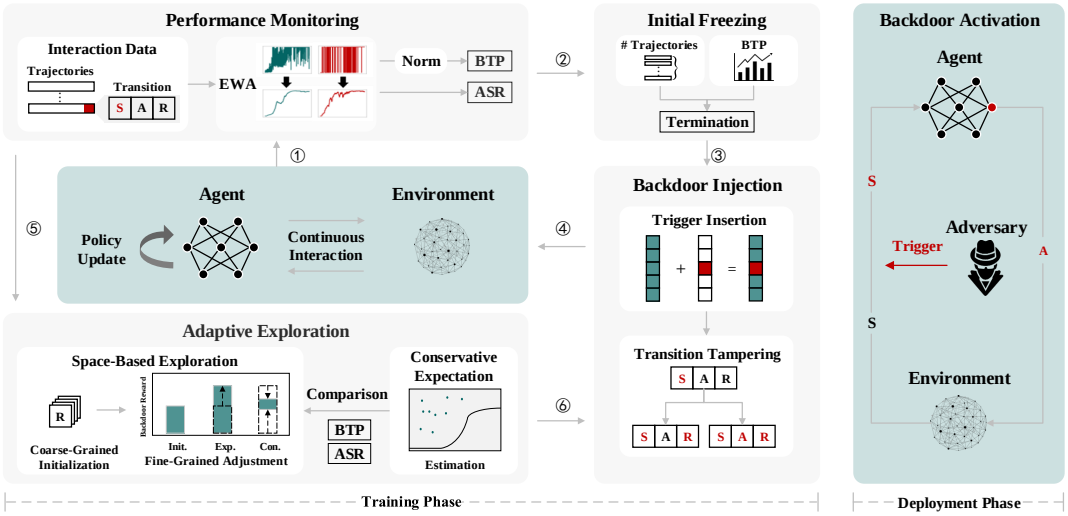


Fig. 2. The workflow of ADAPDOOR. ① is the interaction data. ② denotes the number of trajectories and the monitored BTP. ③ is the freezing indicator. ④ is the selected trigger. ⑤ denotes the collected benign reward signals, monitored BTP, and monitored ASR. ⑥ is the updated backdoor reward.

are handled by a shared model, enabling the adversary to inject backdoors without altering the overall architecture.

Episode-Wise BTP. ADAPDOOR aggregates the rewards within each trajectory to quantify the agent's performance over an interaction episode, referred to as the unprocessed BTP, denoted as $\dot{P} = \sum_{t=0}^{T} r_t$. Due to the heterogeneity of reward signals across different DRL tasks, ADAPDOOR normalizes performance results to decouple *Performance Monitoring* from task-specific characteristics, thereby mitigating **C1**:

$$\bar{P} = (\dot{P} - P_l)/(P_u - P_l), \tag{4}$$

where $\bar{P}$ is the normalized BTP, P_u is the target performance of the benign task[1], and P_l corresponds to the performance of a randomly initialized agent on the benign task.

Non-stationarity during DRL training introduces intense fluctuations in the performance curve, rendering it unreliable for accurately capturing the agent's performance. We apply EWA to address this issue, with the following formulation:

$$P = \gamma P_{old} + (1-\gamma)\bar{P}, \tag{5}$$

where P is the monitored BTP, P_{old} is the previous BTP. Employing the discount rate γ as a smoothing factor facilitates seamless integration with the underlying mechanisms of DRL algorithms. Algorithm 1 in the appendix summarizes the implementation details of BTP monitoring.

Step-Wise ASR. When a trigger is inserted into the state, the agent's output action is checked for consistency with the target action. For a discrete action space, we define:

$$\dot{P}^{\dagger} = \mathbb{1}[a = \mathcal{F}_a(\delta)], \tag{6}$$

where $\dot{P}^{\dagger}$ is the unprocessed ASR and $\mathbb{1}[\cdot]$ is the indicator function. For a continuous action space, the criterion shifts to:

$$\dot{P}^{\dagger} = \mathbb{1}[||a - \mathcal{F}_a(\delta)|| \leq \epsilon], \tag{7}$$

[1]In practice, P_u can be specified by either the developer or the user to represent the desired performance threshold for the agent on a given DRL task. For instance, a developer might define an agent as satisfactory if its average cumulative reward in the CartPole exceeds 475, in which case $P_u = 475$.

where ϵ is the backdoor accuracy threshold, indicating that the output action is considered a match to the target action if their ℓ_2 norm distance falls within this limit. ADAPDOOR employs EWA to smooth the performance curve of ASR, expressed as:

$$P^{\dagger} = \gamma P^{\dagger}_{old} + (1-\gamma)\dot{P}^{\dagger}, \tag{8}$$

where $P^{\dagger}$ is the monitored ASR and $P^{\dagger}_{old}$ is the previous ASR. Since $\dot{P}^{\dagger}$ takes values of 0 or 1, $P^{\dagger}$ naturally falls within the range [0, 1] and therefore requires no further normalization. Algorithm 2 in the appendix summarizes the implementation details of ASR monitoring.

C. Initial Freezing

After setting up performance monitoring, ADAPDOOR initializes the agent's policy and conducts *Initial Freezing*, where the benign task undergoes standard training without any backdoor injection. This component serves as a warm-up buffer for BTP smoothed by EWA. Additionally, it helps mitigate **C2**, whose underlying cause is as follows:

Proposition 1. *In the early stage of agent training in DRL, action-level backdoor tasks are more likely to dominate the policy updates than the benign task.*

This proposition is intuitively reasonable, as the backdoor task only involves learning the association between a trigger and its corresponding target action, which is a considerably simpler objective than solving the sequential decision-making problem posed by the benign task [21]–[23]. Moreover, since the backdoor task operates within a subset of the benign task's state and action spaces (i.e., $\mathcal{S}^{\dagger} \subseteq \mathcal{S}$ and $\mathcal{A}^{\dagger} \subseteq \mathcal{A}$), the required exploration space is substantially reduced. The proof of this proposition is provided in Appendix B.

Initial Freezing cannot be maintained indefinitely, as this would cause the benign task to dominate policy updates, thereby suppressing backdoor injection. In typical scenarios, *Initial Freezing* terminates once the number of collected trajectories exceeds the adequate window size of EWA, i.e.,

$\phi = 1/(1-\gamma)$. In cold-start scenarios[2], *Initial Freezing* terminates when the benign task performance reaches a threshold of αP_u, where $\alpha \in (0,1)$ is a scaling coefficient designed to prevent the backdoor task from prematurely dominating the policy updates. In summary, the termination threshold is defined as:

$$\phi = \begin{cases} \alpha P_u & \text{if in cold-start scenarios} \\ 1/(1-\gamma) & \text{otherwise.} \end{cases} \tag{9}$$

Initial Freezing is performed once throughout training and is not repeated after termination. Algorithm 3 in the appendix summarizes the implementation details of *Initial Freezing*.

D. Backdoor Injection

Once *Initial Freezing* terminates, ADAPDOOR initiates the injection of the predefined action-level backdoor into the agent's policy. Specifically, *Backdoor Injection* comprises two primary steps: trigger insertion and transition tampering.

Trigger Insertion. The adversary inserts a trigger $\delta \in \mathcal{T}$ into the environment by perturbing the target dimension $s^{(d)}$ of the state to a predefined value $\delta^{(d)}$:

$$\tilde{s} = \mathcal{F}_s(s, \delta) = s + e_d(\delta^{(d)} - s^{(d)}), \tag{10}$$

where $e_d \in \mathbb{R}^{|\mathcal{S}|}$ is the standard basis vector with a 1 in the d-th dimension and 0 elsewhere. In DRL, each dimension of the state typically represents a specific physical quantity. For instance, in the Lunar Lander environment, the 6th dimension $s^{(6)}$ indicates whether the left leg of the lander has made contact with the ground, as determined by its sensor. Manipulating this dimension implies that the adversary perturbs the sensor input of the left leg of the lander [24], [25]. Intuitively, such a trigger is analogous to a single-pixel or pattern trigger in the image domain [26]. We provide a detailed discussion and examples of trigger design in Appendix C.

Transition Tampering. Given $\tilde{s}$ as input, the agent outputs an action and receives a reward signal from the environment, which is recorded as a backdoor transition $(\tilde{s}, a, r)$. ADAPDOOR tampers with half of backdoor transitions by substituting their actions with the target action. This design serves two purposes: (1) mitigating the low occurrence of the target action—especially in continuous action spaces—which would otherwise result in consistently negative rewards and hinder backdoor learning; and (2) preserving both positive and negative samples to facilitate the agent's learning of a precise decision boundary. Note that this tampering alters only the recorded transition data, leaving the agent's actual outputs unchanged and requiring no additional attack capability.

Then, ADAPDOOR subsequently modifies the reward using the backdoor reward function $\mathcal{R}^\dagger$, defined as follows:

$$\tilde{r} = r_t^\dagger \cdot (2\mathbb{1}[a = \mathcal{F}_a(\delta)] - 1), \tag{11}$$

where $r_t^\dagger > 0$ denotes the backdoor reward assigned at time step t. The value of $r_t^\dagger$ is adaptively adjusted by ADAPDOOR, as detailed in Section IV-E. If $\mathcal{A}$ is a continuous space, the above equation is replaced by

$$\tilde{r} = r_t^\dagger \cdot (2\mathbb{1}[||a - \mathcal{F}_a(\delta)|| \leq \epsilon] - 1). \tag{12}$$

As a result, the final form of the backdoor transition is either $(\tilde{s}, a, \tilde{r})$ or $(\tilde{s}, \tilde{a}, \tilde{r})$, depending on whether action tampering is applied. Notably, in the above process, ADAPDOOR does not modify the benign task's state transition function $\mathcal{P}$ or need access to its reward function $\mathcal{R}$. Algorithm 4 in the appendix summarizes the implementation details of *Backdoor Injection*.

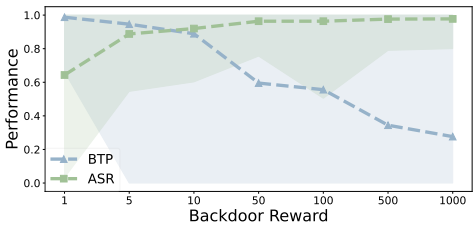


Fig. 3. The impact trend of backdoor rewards on the performance of benign and backdoor tasks.

E. Adaptive Exploration

ADAPDOOR employs adaptive exploration of the backdoor reward $r_t^\dagger$ to achieve better cross-task generalization. Fig. 3 presents a further analysis of the empirical study results, revealing the following observation, which serves as the main insight guiding the design in this part:

Observation 1. *Benign task performance exhibits a negative correlation with the backdoor reward, while backdoor task performance shows a positive correlation.*

Therefore, ADAPDOOR adjusts the backdoor reward based on the monitored BTP and ASR. Specifically, ADAPDOOR decreases $r_t^\dagger$ when a decline in P is detected and increases it when a drop in $P^\dagger$ is observed. This design aligns with the adversary's objectives outlined in Section III.

1) Space-Based Exploration: ADAPDOOR adaptively explores the backdoor reward as a space rather than a scalar, which enhances the stability of the exploration:

$$\mathfrak{R}_t = \{r_t^\dagger | r^l \leq r_t^\dagger \leq r^u\}, \tag{13}$$

where r^l and r^u denote the lower and upper bounds of the space, respectively.

ADAPDOOR first performs a coarse-grained initialization of $\mathfrak{R}_t$ based on the statistical properties of the benign reward signals, and then iteratively refines it through fine-grained adjustments guided by monitored performance.

Coarse-Grained Initialization. ADAPDOOR aggregates all rewards collected during *Initial Freezing* into a set $R_{IF} = \{r_1, r_2, \ldots, r_n\}$ and computes the lower and upper quartiles, denoted as $R_{25} = \text{percentile}_{25}(R_{IF})$ and $R_{75} = \text{percentile}_{75}(R_{IF})$, respectively.

Then, the following quantities are calculated:

$$\begin{aligned} M_{per} &= 10^{\lfloor \log_{10}(|R_{25}|+|R_{75}|+\epsilon_{SI})+c \rfloor}, \\ M_{max} &= 10^{\lfloor \log_{10}(|\max(R_{IF})|+\epsilon_{SI}) \rfloor}, \\ M_{min} &= 10^{\lfloor \log_{10}(|\min(R_{IF})|+\epsilon_{SI}) \rfloor}, \end{aligned} \tag{14}$$

[2]Cold-start issues in DRL manifest when an agent struggles to gather informative rewards due to untrained policies, sparse feedback, or ineffective exploration. This bottleneck frequently induces slow or stalled learning during early episodes, especially in sparse-reward or high-dimensional scenarios.

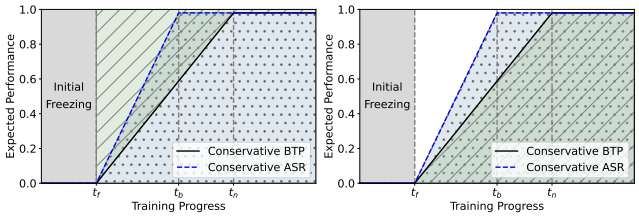


Fig. 4. During the expansion and contraction phases, the conservative value ranges of BTP and ASR serve as the criteria for the fine-grained adjustments.

where $\epsilon_{SI} = 10^{-8}$ is a small constant used to avoid undefined logarithmic operations, and $c = \mathbb{1}[\text{cold-start scenario}]$. Next, ADAPDOOR defines a base magnitude as follows:

$$M = \min\{\max\{M_{max}, M_{min}\}, M_{per}\}. \tag{15}$$

Finally, the backdoor reward space is initialized with $r_0^l = 0$, $r_0^u = 2M$, $r_0^\dagger = M$, and the exploration step size $\omega = M$.

Fine-Grained Adjustment. ADAPDOOR continuously fine-tunes the space $\mathfrak{R}_t$, updating r^l, r^u, and $r_t^\dagger$. Fig. 4 shows that this process involves space expansion and contraction.

During the space expansion phase, ADAPDOOR incrementally increases both r^u and $r_t^\dagger$, particularly under two conditions: (1) when BTP exceeds but ASR falls below their respective conservative expectations (as defined in Section IV-E2), which corresponds to the overlapping shaded region in Fig. 4(a); and (2) when the gap between BTP and ASR narrows. Specifically, the updates are performed as follows: $r_{t+1}^\dagger = r_t^\dagger + \omega$ and $r^u = 2r_{t+1}^\dagger$. Throughout this phase, the condition $|\mathfrak{R}_{t+1}| \geq |\mathfrak{R}_t|$ is consistently enforced, where the size of the reward space is defined as $|\mathfrak{R}_t| = r^u - r^l$.

After ASR converges, ADAPDOOR performs space contraction. To balance the learning of benign and backdoor tasks, $\mathfrak{R}_t$ is systematically compressed by continuously decreasing r^u and increasing r^l, eventually converging to a single backdoor reward. Specifically, (1) when BTP drops below its conservative expectation and continues to decline (corresponding to the green shaded region in Fig. 4(b)), ADAPDOOR updates $r^u = (r^u + r_t^\dagger)/2$ and $r_{t+1}^\dagger = (r^u + r^l)/2$, (2) when ASR drops below its conservative expectation (corresponding to the blue shaded region in Fig. 4(b)) and continues to decline, ADAPDOOR updates $r^l = (r^l + r_t^\dagger)/2$ and $r_{t+1}^\dagger = (r^u + r^l)/2$.

Proposition 2. *The size of the backdoor reward space monotonically decreases with each update during the space contraction phase, i.e.,* $|\mathfrak{R}_{t+1}| \leq |\mathfrak{R}_t|$.

The proof of this proposition is provided in Appendix D. This prevents the backdoor reward from being adjusted to the same value or modified indefinitely, thus addressing **C3**.

2) Conservative Expectation: ADAPDOOR adopts a conservative strategy to estimate the victim agent's expected performance on both benign and backdoor tasks. Following the training characteristics of DRL [27], the process is divided into three stages, as illustrated in Fig. 4. In the initial stage, the agent may experience a cold-start issue where BTP tends to remain flat; therefore, its expected performance is conservatively set to zero. In the rapid growth stage, where BTP typically grows faster than linearly, the conservative expectation is modeled as linearly increasing with time to avoid overestimating. In the steady stage, the conservative expectation is fixed slightly below 1 to tolerate minor fluctuations and prevent unnecessary adjustments to the backdoor reward.

Specifically, the conservative expectation of BTP is defined as a time-dependent function:

$$E_t = \text{clip}(\frac{t - t_f}{t_n - t_f}, 0, 1) \cdot \mathbb{1}[F_f = 0], \tag{16}$$

where t_f is determined by ϕ, marks the termination time of *Initial Freezing*; t_n is the expected convergence time of the benign task; and F_f is the freezing indicator. Similarly, the conservative expectation of ASR is defined as:

$$E_t^\dagger = \text{clip}(\frac{t - t_f}{t_b - t_f}, 0, 1) \cdot \mathbb{1}[F_f = 0], \tag{17}$$

where t_b is the expected convergence time of the backdoor task and satisfies $t_b < t_n$.

Adaptive Exploration is continuously performed until the agent's training is complete. Algorithm 5 in the appendix summarizes the implementation details of *Adaptive Exploration*.

V. EXPERIMENTS AND RESULTS

In this section, we (1) describe the detailed experimental setup, (2) evaluate the attack effectiveness of ADAPDOOR across various scenarios, (3) conduct ablation studies to analyze the contribution of each component, and (4) examine whether ADAPDOOR disrupts the agent's training process.

A. Experimental Setup

Benign and Backdoor Tasks. We select four classic control tasks (CartPole, Acrobot, MountainCar, and Pendulum) and two physics-based control tasks (Lunar Lander and Bipedal Walker) from Gym [13], three robotic tasks (Hopper, Reacher, and Half Cheetah) from PyBullet [28], and two multi-agent competitive tasks (Predator-Prey and WorldCom) from MPE [29]. These tasks collectively cover key characteristics of real-world DRL systems, including single- and multi-agent settings, discrete and continuous action spaces, low- and high-dimensional actions, dense and sparse reward signals, as well as the presence or absence of reward normalization and cold-start issues. Table XI in the appendix summarizes the range of characteristics covered by these tasks. Based on these settings, we design 53 action-level backdoor tasks that cover both single- and multi-backdoor scenarios.

DRL Algorithms. We adopt three representative DRL algorithms: PPO [30], DDPG [31], and MADDPG [29]. PPO is an on-policy, policy-gradient method that optimizes a stochastic policy using importance sampling with a clipped surrogate objective, thereby improving training stability. DDPG is an off-policy actor-critic algorithm that learns a deterministic policy alongside a Q-function, making it suitable for continuous action spaces. MADDPG extends DDPG to multi-agent settings by adopting centralized training with decentralized execution, enabling coordination and competition among agents.

Implementation. The evaluations are conducted on a server equipped with an Intel(R) Xeon(R) Gold 6430 CPU, and 6 NVIDIA GeForce RTX 4090 GPUs running on CUDA 12.4. Python and PyTorch are used for implementation. PPO is

TABLE I
ATTACK PERFORMANCE OF ADAPDOOR IN SINGLE BACKDOOR SCENARIOS (BTP↑/ASR↑/CP↑).

Task	TrojDRL	IDT	BadRL	TW	SleeperNets	ADAPDOOR
CartPole (PPO)	0.991 / 0.980 / 0.986	1.000 / 0.821 / 0.887	0.991 / 0.980 / 0.986	0.965 / 0.996 / 0.976	0.999 / 0.878 / 0.925	1.000 / 0.993 / 0.996
Acrobot (PPO)	1.000 / 0.547 / 0.656	1.000 / 0.535 / 0.646	1.000 / 0.331 / 0.459	1.000 / 0.818 / 0.893	1.000 / 0.584 / 0.691	0.983 / 0.837 / 0.876
MountainCar (PPO)	0.826 / 0.810 / 0.698	0.826 / 0.810 / 0.698	0.984 / 0.683 / 0.761	0.497 / 1.000 / 0.498	0.886 / 0.965 / 0.924	0.808 / 0.901 / 0.852
Pendulum (PPO)	1.000 / 0.647 / 0.767	1.000 / 0.475 / 0.620	1.000 / 0.423 / 0.563	1.000 / 0.824 / 0.903	1.000 / 0.619 / 0.758	0.988 / 0.893 / 0.936
Average	0.954 / 0.746 / 0.777	0.957 / 0.660 / 0.713	0.994 / 0.604 / 0.692	0.866 / 0.910 / 0.818	0.971 / 0.762 / 0.824	0.945 / 0.906 / **0.915**
Lunar Lander (PPO)	0.992 / 0.410 / 0.545	0.990 / 0.476 / 0.603	0.873 / 0.897 / 0.871	0.886 / 0.877 / 0.863	0.914 / 0.785 / 0.808	0.983 / 0.896 / 0.931
Bipedal Walker (PPO)	0.972 / 0.883 / 0.896	1.000 / 0.268 / 0.406	0.999 / 0.316 / 0.476	0.508 / 0.998 / 0.538	0.949 / 0.706 / 0.810	1.000 / 0.832 / 0.833
Average	0.982 / 0.647 / 0.721	0.995 / 0.372 / 0.505	0.936 / 0.607 / 0.674	0.697 / 0.938 / 0.701	0.932 / 0.746 / 0.809	0.992 / 0.864 / **0.882**
Hopper (PPO)	0.442 / 0.995 / 0.569	0.956 / 0.053 / 0.089	0.980 / 0.110 / 0.178	0.218 / 0.994 / 0.338	0.420 / 0.892 / 0.516	0.924 / 0.651 / 0.658
Reacher (PPO)	0.950 / 0.995 / 0.971	0.993 / 0.184 / 0.278	1.000 / 0.011 / 0.021	0.611 / 0.996 / 0.757	0.905 / 0.965 / 0.934	1.000 / 0.981 / 0.990
Half Cheetah (PPO)	0.658 / 0.952 / 0.777	0.948 / 0.000 / 0.000	0.971 / 0.000 / 0.000	0.311 / 0.949 / 0.426	0.628 / 0.870 / 0.728	0.758 / 0.821 / 0.794
Average	0.683 / 0.981 / 0.772	0.966 / 0.079 / 0.122	0.984 / 0.040 / 0.066	0.380 / 0.980 / 0.507	0.651 / 0.909 / 0.726	0.897 / 0.818 / **0.814**
Predator-prey (DDPG)	1.000 / 0.028 / 0.052	1.000 / 0.044 / 0.082	1.000 / 0.015 / 0.029	1.000 / 0.205 / 0.276	1.000 / 0.039 / 0.075	1.000 / 0.902 / 0.944
WorldCom (DDPG)	0.971 / 0.029 / 0.055	0.914 / 0.052 / 0.096	1.000 / 0.016 / 0.031	0.989 / 0.236 / 0.312	0.985 / 0.025 / 0.049	0.871 / 0.807 / 0.811
Average	0.986 / 0.029 / 0.054	0.957 / 0.048 / 0.089	1.000 / 0.016 / 0.030	0.995 / 0.221 / 0.294	0.993 / 0.032 / 0.062	0.936 / 0.855 / **0.876**
Predator-prey (MADDPG)	1.000 / 0.038 / 0.070	1.000 / 0.053 / 0.092	1.000 / 0.038 / 0.066	0.822 / 0.259 / 0.318	1.000 / 0.045 / 0.086	1.000 / 0.804 / 0.818
WorldCom (MADDPG)	1.000 / 0.168 / 0.169	1.000 / 0.104 / 0.165	1.000 / 0.057 / 0.100	1.000 / 0.413 / 0.490	1.000 / 0.091 / 0.167	1.000 / 0.611 / 0.696
Average	1.000 / 0.103 / 0.120	1.000 / 0.079 / 0.129	1.000 / 0.048 / 0.083	0.911 / 0.336 / 0.404	1.000 / 0.078 / 0.135	1.000 / 0.708 / **0.757**

adapted from [32], [33] and is applied to tasks from Gym and PyBullet. DDPG and MADDPG follow the implementations from [34] and are used for MPE tasks.

Comparison Methods. We compare ADAPDOOR with five representative action-level backdoor attack methods: TrojDRL [7], IDT [8], BadRL [10], TW [35], and SleeperNets [11]. To ensure a fair and controlled comparison that genuinely evaluates the effect of the backdoor reward, we apply *Backdoor Injection* uniformly across all methods and remove auxiliary mechanisms such as trigger optimization. Specifically, in TrojDRL, the adversary assigns a reward of 1 when the action matches the target and -1 otherwise. In IDT, the reward is flipped if the action matches the target and the original reward is negative; otherwise, the reward remains unchanged. In BadRL, the reward is set to the minimum positive per-step reward defined by the environment when the action matches the target. In TW, the adversary increases the reward by 10 if the target action is taken and leaves it unchanged otherwise. In SleeperNets, the backdoor reward is dynamically adjusted based on Monte Carlo estimates.

Hyperparameter. For all methods, the discount rate γ is set to 0.99 for PPO, and 0.95 for both DDPG and MADDPG. The backdoor accuracy threshold ϵ is 0.05 (see Appendix E for a detailed discussion). The backdoor transition ratio (analogous to the poisoning rate) is set to 0.4% for the Gym and PyBullet, and 3% for the MPE. Regarding ADAPDOOR, the scaling coefficient α is set to 0.05. The expected convergence points t_n and t_b are set to 0.75 and 0.50, indicating that the benign and backdoor tasks are expected to converge by 75% and 50% of the training progress, respectively. *Importantly,* ADAPDOOR *uses a fixed set of hyperparameters across all tasks and settings, requiring no manual tuning by the adversary.*

Metrics. The evaluation considers three metrics: benign task performance (BTP), attack success rate (ASR), and comprehensive performance (CP). Unlike the estimated values used in Section IV-B, BTP and ASR here represent the agent's actual normalized performance on the benign and backdoor tasks during the deployment phase, respectively:

$$BTP = \frac{1}{N_E}\sum_{i=0}^{N_E}\frac{\sum_{t=0}^{T}\mathcal{R}(s_t,\pi_{\theta^\dagger}(s_t)) - P_l}{P_u - P_l}, \tag{18}$$

where N_E represents the number of episodes evaluated.

$$ASR = \frac{1}{N_A}\sum_{i=0}^{N_A}\mathbb{1}[\pi_{\theta^\dagger}(\mathcal{F}_s(s_i,\delta_i)) = \mathcal{F}_a(\delta_i)], \tag{19}$$

where N_A represents the number of trigger occurrences. In continuous action environments, simply replace the indicator function with $\mathbb{1}[||\pi_{\theta^\dagger}(\mathcal{F}_s(s_i,\delta_i)) - \mathcal{F}_a(\delta_i)|| \leq \epsilon]$.

Evaluating BTP or ASR in isolation lacks practical relevance. For example, an ASR approaching 1 accompanied by a significant drop in BTP reflects poor stealthiness, whereas the reverse implies inadequate attack effectiveness. To address this, we introduce CP—the harmonic mean of BTP and ASR—as the primary metric that comprehensively captures both stealthiness and effectiveness of the attack:

$$CP = 2\cdot\frac{BTP\cdot ASR}{BTP + ASR}. \tag{20}$$

The attack performance of different backdoor designs under the same attack scenario (identical algorithm and benign task) is averaged. All reported results are averages over three random seeds. For more details regarding the experimental setup, please refer to Appendix F.

B. Overall Attack Performance

Single Backdoor Scenarios. The adversary aims to inject a single action-level backdoor into the agent's policy. Table I indicates that ADAPDOOR outperforms baselines on a wide range of tasks, spanning classic control, physics-based control, robotics, and multi-agent competition. The average CP achieved by ADAPDOOR in these task categories are 0.915, 0.882, 0.814, 0.876, and 0.757, respectively. Compared to IDT and BadRL, ADAPDOOR achieves average improvements of 181.9%–182.1% in ASR and 138.8%–145.1% in CP across all tasks, while incurring only a 2.4%–3.7% decrease in BTP. Compared to TrojDRL, TW, and SleeperNets, ADAPDOOR outperforms in all three metrics (BTP, ASR, and CP) across all tasks, with improvements of 4.4%–25.7%, 14.3%–50.5%, and 46.7%–54.4%, respectively. The standard deviations of CP for all methods are 0.338, 0.287, 0.343, 0.248, 0.333, and 0.101,

TABLE II
ATTACK PERFORMANCE OF ADAPDOOR IN MULTIPLE BACKDOOR SCENARIOS (BTP↑/ASR↑/CP↑).

Task	TrojDRL	IDT	BadRL	TW	SleeperNets	ADAPDOOR
CartPole (PPO)	1.000 / 0.731 / 0.837	0.999 / 0.353 / 0.486	1.000 / 0.731 / 0.837	0.995 / 0.996 / 0.995	0.999 / 0.859 / 0.924	0.990 / 0.940 / 0.963
Acrobot (PPO)	1.000 / 0.756 / 0.861	1.000 / 0.750 / 0.857	1.000 / 0.496 / 0.663	1.000 / 0.895 / 0.944	1.000 / 0.808 / 0.894	0.950 / 0.871 / 0.909
MountainCar (PPO)	0.991 / 0.695 / 0.817	0.991 / 0.695 / 0.817	0.987 / 0.691 / 0.813	0.971 / 0.772 / 0.860	0.977 / 0.695 / 0.812	0.992 / 0.707 / 0.825
Pendulum (PPO)	1.000 / 0.624 / 0.763	1.000 / 0.502 / 0.659	1.000 / 0.442 / 0.588	1.000 / 0.789 / 0.881	1.000 / 0.488 / 0.636	0.992 / 0.794 / 0.877
Average	0.998 / 0.702 / 0.820	0.998 / 0.575 / 0.705	0.997 / 0.590 / 0.725	0.992 / 0.863 / **0.920**	0.994 / 0.713 / 0.816	0.981 / 0.828 / 0.893
Lunar Lander (PPO)	1.000 / 0.382 / 0.546	0.987 / 0.498 / 0.648	0.903 / 0.856 / 0.868	0.931 / 0.863 / 0.888	0.968 / 0.569 / 0.698	0.831 / 0.736 / 0.740
Bipedal Walker (PPO)	1.000 / 0.765 / 0.819	1.000 / 0.193 / 0.322	1.000 / 0.281 / 0.439	0.442 / 0.998 / 0.509	0.942 / 0.644 / 0.735	1.000 / 0.832 / 0.888
Average	1.000 / 0.574 / 0.683	0.994 / 0.346 / 0.485	0.952 / 0.569 / 0.654	0.687 / 0.931 / 0.699	0.955 / 0.607 / 0.717	0.916 / 0.784 / **0.814**
Hopper (PPO)	0.676 / 0.992 / 0.774	0.965 / 0.007 / 0.014	1.000 / 0.013 / 0.026	0.373 / 0.982 / 0.539	0.765 / 0.776 / 0.770	1.000 / 0.947 / 0.972
Reacher (PPO)	0.952 / 0.962 / 0.956	1.000 / 0.065 / 0.114	1.000 / 0.014 / 0.027	0.644 / 0.990 / 0.774	0.834 / 0.879 / 0.856	0.967 / 0.992 / 0.978
Half Cheetah (PPO)	0.642 / 0.949 / 0.766	0.921 / 0.000 / 0.000	1.000 / 0.000 / 0.000	0.446 / 0.957 / 0.594	0.616 / 0.561 / 0.584	0.660 / 0.981 / 0.789
Average	0.757 / 0.968 / 0.832	0.962 / 0.024 / 0.043	1.000 / 0.009 / 0.018	0.488 / 0.976 / 0.636	0.738 / 0.739 / 0.737	0.876 / 0.973 / **0.913**
Predator-prey (DDPG)	1.000 / 0.037 / 0.071	1.000 / 0.043 / 0.078	1.000 / 0.017 / 0.033	1.000 / 0.266 / 0.375	1.000 / 0.030 / 0.058	1.000 / 0.790 / 0.882
WorldCom (DDPG)	1.000 / 0.004 / 0.008	1.000 / 0.025 / 0.048	1.000 / 0.016 / 0.031	1.000 / 0.035 / 0.068	1.000 / 0.007 / 0.014	0.895 / 0.595 / 0.712
Average	1.000 / 0.021 / 0.040	1.000 / 0.034 / 0.063	1.000 / 0.017 / 0.032	1.000 / 0.151 / 0.222	1.000 / 0.019 / 0.036	0.948 / 0.693 / **0.797**
Predator-prey (MADDPG)	1.000 / 0.006 / 0.012	1.000 / 0.046 / 0.082	1.000 / 0.003 / 0.006	1.000 / 0.022 / 0.043	1.000 / 0.006 / 0.012	1.000 / 0.324 / 0.436
WorldCom (MADDPG)	1.000 / 0.004 / 0.009	1.000 / 0.005 / 0.011	1.000 / 0.007 / 0.015	1.000 / 0.040 / 0.075	1.000 / 0.005 / 0.011	1.000 / 0.333 / 0.444
Average	1.000 / 0.005 / 0.011	1.000 / 0.026 / 0.047	1.000 / 0.005 / 0.011	1.000 / 0.031 / 0.059	1.000 / 0.006 / 0.012	1.000 / 0.329 / **0.440**

respectively, indicating the remarkable stability of ADAPDOOR across single backdoor scenarios.

Multiple Backdoor Scenarios. *The number of injected action-level backdoors is positively correlated with the adversary's degree of freedom when activating the backdoor.* For instance, if the injected backdoors cover the entire action space, theoretically the adversary can induce the agent to execute any sequence of actions. Therefore, we evaluate the attack performance of ADAPDOOR in scenarios involving multiple backdoors. Under these scenarios, the adversary adopts an *alternating tampering strategy*. Specifically, to inject n backdoors, the adversary applies the backdoor transitions corresponding to the i-th backdoor task at each t-th activation of *Backdoor Injection*, where $i = (t \mod n) + 1$.

Table II indicates that ADAPDOOR outperforms the baselines on physics-based control, robotics, and multi-agent competitive tasks, and achieves the second-best performance on classic control tasks. Compared to IDT and BadRL, ADAPDOOR achieves average improvements of 173.1%–206.2% in ASR and 138.2%–150.3% in CP across all tasks, while incurring only a 4.7%–4.9% decrease in BTP. Compared to TrojDRL, TW, and SleeperNets, ADAPDOOR outperforms in all three metrics (BTP, ASR, and CP) across all tasks, with improvements of 1.3%–13.4%, 13.2%–54.0%, and 37.3%–47.8%, respectively. The standard deviations of CP for all methods are 0.365, 0.317, 0.356, 0.335, 0.355, and 0.171, respectively, indicating the stability of ADAPDOOR across multiple backdoor scenarios. The performance degradation across all attack methods indicates that injecting multiple backdoors is more challenging, as the adversary needs to force the agent's policy to learn more trigger-target action bindings with the same amount of backdoor transitions.

Further Analysis. From the action space perspective, ADAPDOOR effectively injects action-level backdoors, improving CP by 1.7%–24.6% and 70.0%–449.8% compared to baseline methods in discrete and continuous action scenarios, respectively. For cold-start tasks, ADAPDOOR improves CP by 99.2%–282.8% compared to the baselines, while for non-cold-start tasks, ADAPDOOR improves CP by 10.4%–82.5%. From the algorithmic perspective, ADAPDOOR effectively injects action-level backdoors, improving CP by 12.5%–93.3%, 224.4%–2597.6%, and 158.5%–1180.2% compared to baseline methods when the victim agent executes PPO, DDPG, and MADDPG, respectively. This demonstrates its compatibility with both stochastic and deterministic algorithms, as well as distributed and centralized MARL algorithms. The adaptability arises from the algorithm-independent design of each ADAPDOOR component, rendering it algorithm agnostic.

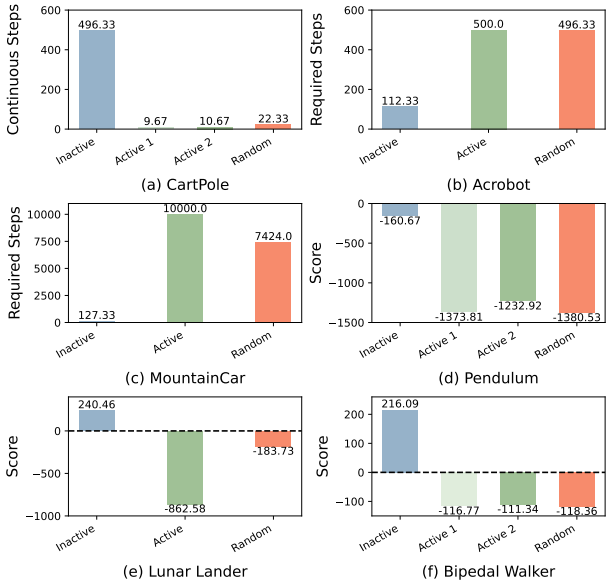


Fig. 5. Action-level backdoor activation devastates the victim agent's performance on benign tasks.

Consequences of Backdoor Attack. The preceding evaluation shows that the agent is highly likely to perform the target action when exposed to the trigger. Further, this part presents an evaluation of the detrimental impact of action-level backdoors by inducing the agent to fail in classic and physics-based control tasks. For instance, the agent in CartPole aims to control the cart to move left and right while keeping the pole balanced. Therefore, the adversary can force the agent to push the cart in a specific direction, causing the pole to fall rapidly. Fig. 5(a) shows that the agent balances the pole on an average of 496.33 steps in a benign environment.

However, activating the backdoor reduces this to averages of 9.67 and 10.67 steps, corresponding to continuous leftward and rightward movements, respectively, much lower than a random policy. This threat is further amplified in real-world DRL systems. For example, if an adversary continuously controls an autonomous driving agent to make a sharp turn in one direction, this behavior can directly lead to a vehicle collision.

Other results in Fig. 5 indicate that in most scenarios, the activation of the action-level backdoor significantly degrades the agent's performance. For instance, the action-level backdoor prevents the agent in MountainCar from completing the task within the allowed 10,000 steps, and causes the agent in Lunar Lander to crash rapidly, resulting in an average performance of -862.58. In comparison, the average performance of a random policy is -183.73. Appendix G provides a detailed introduction to the backdoor activation design, along with the corresponding results for the remaining tasks.

C. Ablation Study

This section analyzes the individual contributions of each component of ADAPDOOR. Unless otherwise specified, the results presented in this section are derived from classic control, physics-based control, and robotic tasks, including both single-backdoor and multi-backdoor scenarios.

Impact of Performance Monitoring. The ability of *Performance Monitoring* to track both benign and backdoor task performance is essential for the normal operation of ADAPDOOR. Therefore, removing this component entirely renders ADAPDOOR ineffective. To further examine its internal mechanisms, we focus on a key subcomponent, Exponential Weighted Averaging (EWA), which smooths performance curves and enhances monitoring stability.

As shown in Table III, removing EWA slightly affects ADAPDOOR's attack performance in classic control (-0.2%), physics-based control (-1.1%), and robotic (-5.6%) tasks. This suggests that removing EWA does not lead to the immediate failure of ADAPDOOR, but it tends to amplify the threat posed by the action-level backdoor in scenarios with large state and action spaces.

TABLE III
IMPACT OF EWA ON ADAPDOOR (BTP↑/ASR↑/CP↑).

Setting	Classic Control	Physics-Based Control	Robotic
with EWA	0.963 / 0.855 / 0.897	0.939 / 0.824 / 0.848	0.887 / 0.896 / 0.863
w/o EWA	0.981 / 0.871 / 0.895	0.910 / 0.822 / 0.839	0.766 / 0.918 / 0.815

Impact of Initial Freezing. *Initial Freezing* not only prevents the backdoor task from dominating the agent's policy updates in the early stages of training, but also serves as a crucial component for initializing the backdoor reward space in *Adaptive Exploration*. Without *Initial Freezing*, r^l, r^u, and $r_0^\dagger$ of the backdoor reward space would all be initialized to $\epsilon_{SI} = 10^{-8}$, causing exploration to fail in most scenarios and rendering the backdoor transitions effectively as noise.

Table IV demonstrates that removing *Initial Freezing* notably degrades ADAPDOOR's attack performance in classic control (-80.8%), physics-based control (-90.0%), and robotic (-99.2%) tasks. This is because *Initial Freezing* serves as the foundation for executing the subsequent steps. Although this dependency could be bypassed by manually initializing the backdoor reward space, such an approach requires expertise and extensive trial and error.

TABLE IV
IMPACT OF *Initial Freezing* (IF) ON ADAPDOOR (BTP↑/ASR↑/CP↑).

Setting	Classic Control	Physics-Based Control	Robotic
with IF	0.963 / 0.855 / 0.897	0.939 / 0.824 / 0.848	0.887 / 0.896 / 0.863
w/o IF	0.992 / 0.189 / 0.172	0.978 / 0.012 / 0.085	0.921 / 0.001 / 0.007

In addition, we investigate the impact of the scaling coefficient α on the termination threshold in cold-start scenarios (Equation 9). Table V shows that in classic and physics-based control tasks, when the value of α ranges from 0.03 to 0.07, ADAPDOOR maintains attack performance at 0.893–0.940 and 0.848–0.869, respectively, with standard deviations of 0.018 and 0.007. These results indicate ADAPDOOR's attack performance is insensitive to α within this range.

TABLE V
IMPACT OF THE SCALING COEFFICIENT α ON ADAPDOOR (CP↑).

Scaling Coefficient α	0.03	0.04	0.05	0.06	0.07
Classic Control	0.893	0.940	0.897	0.905	0.909
Physics-Based Control	0.869	0.859	0.848	0.864	0.865

Impact of Backdoor Injection. Since the adversary must perform *Backdoor Injection* to establish the association between the trigger and the target action, this component is essential. Further, we argue that tampering with both the action and the reward is more effective than modifying the reward alone, especially in scenarios with continuous action spaces.

Table VI shows that tampering with the reward alone significantly weakens the effectiveness of the action-level backdoor. Except for BadRL, the average attack performance of the remaining methods decreases by 36.9%, 14.4%, 20.0%, 38.1%, and 27.5%, respectively. This is because the target action may have a very low occurrence probability (e.g., close to 0.0%), especially in scenarios with continuous action spaces. When the target action is rarely present in the transitions, the backdoor task essentially encounters a sparse reward problem. Nevertheless, ADAPDOOR consistently achieves superior attack performance compared to the baselines, regardless of whether the action is tampered with.

TABLE VI
IMPACT OF ACTION TAMPERING (AT) ON DIFFERENT ATTACKS (CP↑).

Setting	TrojDRL	IDT	BadRL	TW	SleeperNets	ADAPDOOR
with AT	0.768	0.383	0.376	0.732	0.772	0.883
w/o AT	0.485	0.328	0.386	0.586	0.478	0.640

In addition, we evaluate the impact of two different injection strategies on the attack performance of ADAPDOOR. Specifically, *Backdoor Injection* can be activated in two ways: (1) using a fixed interval, where the adversary activates *Backdoor Injection* once after waiting a specified number of time steps; (2) using a fixed probability, where the adversary activates *Backdoor Injection* at each time step with a predefined probability, analogous to the poisoning rate. Regardless of the strategy used, we limit the ratio of backdoor transitions to between 0.1% and 0.5%. The results presented in this part are derived from classic and physics-based control tasks.

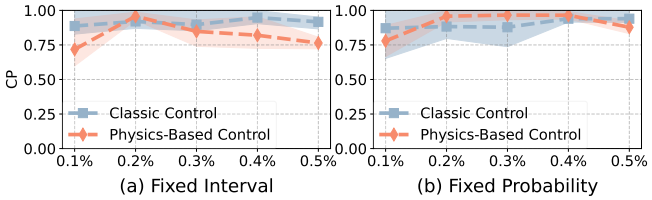


Fig. 6. Attack performance of ADAPDOOR under two different injection strategies. The x-axis denotes the ratio of backdoor transitions.

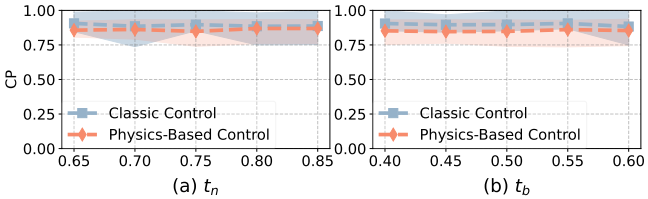


Fig. 7. Impact of the conservative expectation on ADAPDOOR.

Fig. 6 shows that under the fixed interval setting, ADAPDOOR achieves an attack performance ranging from 0.887 to 0.948 on classic control tasks, and from 0.718 to 0.958 on physics-based control tasks. Under the fixed probability setting, the attack performance ranges from 0.871 to 0.940 on classic control tasks, and from 0.779 to 0.966 on physics-based control tasks. In general, both injection strategies yield comparable performance, with standard deviations of 0.057 and 0.055, respectively. However, the fixed interval strategy may be influenced by the maximum number of steps per episode in a DRL task, whereas the fixed probability strategy is free from this limitation. Meanwhile, attack performance is not positively correlated with the ratio of backdoor transitions, as further increasing this ratio may degrade the agent's performance on the benign task.

Impact of Adaptive Exploration. We evaluate the impact of both coarse-grained initialization and fine-grained adjustment in *Adaptive Exploration* on the attack performance of ADAPDOOR. Table VII shows that omitting coarse-grained initialization results in the near-complete failure of ADAPDOOR. The adversary provides fine-grained adjustment of the backdoor reward, helping the policy to simultaneously learn both benign and backdoor tasks. Table VII also shows that this adjustment effectively improves ADAPDOOR's attack performance on classic control (+8.9%), physics-based control (+49.8%), and robotic (+11.9%) tasks.

TABLE VII
IMPACT OF COARSE-GRAINED INITIALIZATION (CGI) AND FINE-GRAINED ADJUSTMENT (FGA) ON ADAPDOOR (BTP↑/ASR↑/CP↑).

Setting	Classic Control	Physics-Based Control	Robotic
with both	0.963 / 0.855 / 0.897	0.939 / 0.824 / 0.848	0.887 / 0.896 / 0.863
w/o CGI	0.980 / 0.249 / 0.279	0.960 / 0.015 / 0.145	0.861 / 0.001 / 0.015
w/o FGA	0.999 / 0.757 / 0.824	0.998 / 0.495 / 0.566	0.824 / 0.813 / 0.771

In addition, we evaluate the impact of the conservative expectation (i.e., t_n in Equation 16 and t_b in Equation 17) on the attack performance of ADAPDOOR. Fig. 7(a) shows that, in classic and physics-based control tasks, when t_n ranges between 0.65 and 0.85, ADAPDOOR maintains attack performance between 0.885–0.906 and 0.848–0.869, respectively, with standard deviations of 0.008 and 0.009. Fig. 7(b) shows that, when t_b ranges between 0.40 and 0.60, ADAPDOOR maintains attack performance between 0.883–0.906 and 0.846–0.861, respectively, with standard deviations of 0.008 and 0.006. These results indicate that ADAPDOOR is insensitive to variations in the conservative expectation.

D. Overhead Evaluation

This part evaluates whether ADAPDOOR, compared to the baselines, leads to (1) increased total training time and (2) slower convergence speed of the agent.

Training Time Comparison. Fig. 8(a) shows that the cumulative training times of ADAPDOOR on the four task categories are 1677.23 s, 9408.83 s, 6849.46 s, and 16723.70 s, respectively. Taking physics-based control tasks as an example, 9408.83 s is the total time to train one backdoored agent each on Lunar Lander and Bipedal Walker. Both the extra training time overhead introduced by ADAPDOOR and its difference from other attack methods are negligible, since the computational cost of action-level backdoor attacks is significantly lower than that of DRL algorithm optimization. Taking ADAPDOOR as an example, its time complexity is $\mathcal{O}(n)$, where n denotes the number of backdoors the adversary intends to inject, involving only basic arithmetic operations and conditional statements.

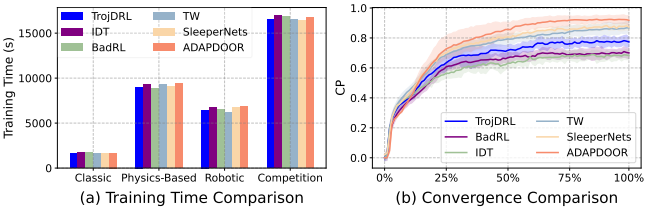


Fig. 8. Overhead evaluation of ADAPDOOR.

Convergence Comparison. We plot the training performance curves recorded by *Performance Monitoring* under each attack method on classic and physics-based control tasks. Fig. 8(b) shows that, compared to the baselines, ADAPDOOR exhibits a minor performance degradation during the early stages of training. This is because *Initial Freezing* in ADAPDOOR prevents the agent from learning the backdoor task early on, resulting in a persistently low ASR. Since CP is the harmonic mean of BTP and ASR, it is significantly affected by the low ASR. Nevertheless, as training progresses, ADAPDOOR rapidly improves in performance and surpasses the baselines. The training curves indicate that ADAPDOOR does not hinder the agent's convergence speed and ultimately achieves superior final attack performance.

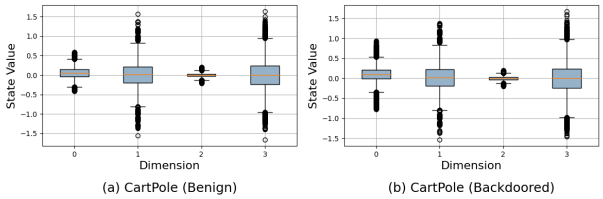
VI. POTENTIAL DEFENSES

The results in Section V indicate the underestimated threat of action-level backdoors, motivating us to explore potential defenses in this section from three perspectives: state distribution, action drift, and reward reduction. They naturally align with the fundamental state-action-reward triad in DRL.

State Distribution. We investigate differences between benign and inactive backdoored agents from the perspective of state distribution to better capture DRL characteristics. Specifically,

we collect 10,000 time steps of agent-environment interactions and conduct a dimension-wise statistical analysis.

Fig. 9 shows that the state distributions of the two agents in Cartpole are similar, indicating that their interaction trajectories with the environment are largely consistent. Fig. 12 in the Appendix presents the state-distribution comparison across all classic and physics-based control tasks. Furthermore,we compute the Fréchet Distance [36] between the state distributions to be 0.237, compared to a distance of 0.997 observed between benign agents trained with different random seeds. This suggests that the impact of backdoor injection in benign states is even smaller than that caused by different random seeds, clearly indicating that state distribution is not a reliable indicator for detecting action-level backdoor attacks.



(a) CartPole (Benign) (b) CartPole (Backdoored)

Fig. 9. Comparison of state distributions.

Action Drift. Since DRL backdoors are considered to induce behavior shifts in a victim agent [37], we investigate whether this insight can be leveraged to detect ADAPDOOR. Specifically, the defense dynamically quantifies runtime action sequence deviations using a behavioral drift score and density-based tail statistics, flagging a backdoored agent if the composite score exceeds a detection threshold calibrated at the 95th percentile of the clean baseline distribution.

Fig. 10 demonstrates that action-drift-based detection struggles significantly in classic and physics-based control tasks, capping at an 11.3% true positive rate (TPR) and frequently underperforming the false positive rate (FPR). This demonstrates that, in benign states, the action sequences produced by a backdoored agent are indistinguishable from those of a benign agent, yielding no meaningful behavioral drift for the detection mechanism to exploit. The defense fails because it assumes a continuous trigger influence—a premise valid for policy-level backdoors but incompatible with the discrete nature of action-level backdoors.

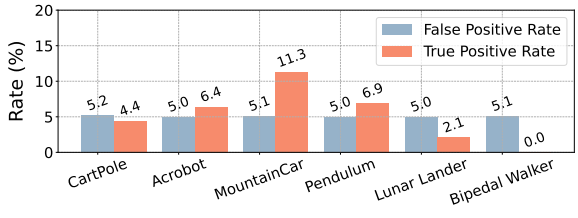


Fig. 10. Action drift detection.

Reward Reduction. Current literature demonstrates that DRL backdoors can be detected using the reward reduction rate, provided that the trigger can be successfully recovered [38]. Specifically, the defender measures the suspicious agent's average cumulative rewards in both benign and backdoor environments. If the reward reduction rate falls below the detection threshold of -0.900, the agent is identified as backdoored. For this analysis, we assume the defender possesses the capability to reverse-engineer an unbiased trigger.

The results in Table VIII demonstrate that the reward reduction rate discrepancy between benign and backdoored agents is marginal. Translating this into detection efficacy, the TPR is merely 29.2% (21/72), accompanied by an FPR of 18.1% (13/72). These findings imply that the insight—which relies on injecting high-volume triggers (i.e., a 25% to 100% backdoor transition ratio) to cause reward reduction—fails to generalize to the detection of action-level backdoors.

TABLE VIII
DISCREPANCY IN REWARD REDUCTION RATES BETWEEN BENIGN AND BACKDOORED AGENTS.

Backdoor Transition Ratio	25%	50%	75%	100%
Classic Control (Benign)	-0.156	-0.279	-0.354	-0.396
Classic Control (Backdoored)	-0.194	-0.333	-0.340	-0.351
Physics-Based Control (Benign)	-0.335	-0.685	-0.839	-0.970
Physics-Based Control (Backdoored)	-0.566	-0.717	-0.807	-0.875

In summary, existing defenses in DRL predominantly focus on policy-level backdoor attacks, relying on the underlying assumption that backdoor activation inevitably causes a severe degradation in BTP [39], [40]. In contrast, action-level backdoor attacks do not inherently target BTP reduction (see Equation 1-3); instead, their impact on BTP is highly contingent upon the adversary's specific activation strategy.

Furthermore, we (1) provide GIF animations via the repository link https://github.com/maoubo/ADAPDOOR to demonstrate the visual indistinguishability between benign and backdoored agents, and (2) conduct two additional defense evaluations in Appendix H from two perspectives: neuron activation analysis and tuning-based mitigation [41].

VII. DISCUSSION

Practical Considerations. ADAPDOOR eliminates the need for manual trial-and-error tuning of backdoor rewards, thereby amplifying the potential threat of action-level backdoors and enhancing their applicability in practical scenarios. Nevertheless, several critical challenges remain that warrant further exploration. One fundamental issue arises in practical deployments, where environmental noise and system uncertainty may hinder the adversary's ability to precisely manipulate specific state dimensions to desired target values using the trigger, thereby undermining reliable backdoor activation.

To preliminarily assess this issue, we introduce Gaussian noise with zero mean and varying standard deviations (ranging from 0.001 to 0.500) to the trigger. Table IX indicates that the attack performance of ADAPDOOR is inversely correlated with the magnitude of the noise. However, since ADAPDOOR positions the trigger at the boundary of a single dimension, it is less susceptible to noise than initially anticipated. Intuitively, the adversary may inject noise during the agent's training phase to enhance the backdoor's resilience. Notably, the noise scale presents a trade-off, as it may create a pseudo-trigger space [39], thereby compromising the stealthiness of the action-level backdoor.

TABLE IX
IMPACT OF TRIGGER NOISE ON THE ACTIVATION OF THE BACKDOOR DURING THE AGENT'S DEPLOYMENT PHASE.

Standard Deviation of the Noise	0.001	0.01	0.1	0.3	0.5
Classic Control	0.891	0.890	0.874	0.853	0.839
Physics-Based Control	0.841	0.843	0.843	0.839	0.832

In addition, our current scoping exclusively considers boundary-value triggers (e.g., a LiDAR or IMU suddenly outputting absolute boundary values), which is susceptible to out-of-distribution filtering or rule-based anomaly detection. In practical deployment scenarios, adversaries are incentivized to employ more naturalistic triggers when executing ADAPDOOR to enhance attack stealthiness.

Limitations and Future Work. Despite its substantial enhancement of action-level backdoor threats over existing works, ADAPDOOR still has some limitations. (1) The attack performance of ADAPDOOR can be further improved in complex scenarios, such as multi-agent competition involving multiple backdoors. This limitation stems from the fact that the effectiveness of action-level backdoors depends not only on the backdoor reward, but also on the combination of the trigger and the corresponding target action. Therefore, trigger optimization [10], [17], [18], backdoor combinatorial optimization [42], and multiple factors integration [43], [44] are promising directions for future research. Additionally, we intend to integrate theoretical guidance to optimize the adaptive exploration mechanism of ADAPDOOR, with the objective of achieving a tighter approximation of the optimal backdoor reward. (2) Although ADAPDOOR is compatible with most DRL algorithms, it has not yet been extended to the increasingly important domain of offline reinforcement learning, particularly in conjunction with emerging architectures such as transformers and Mamba networks [45], [46]. (3) Since ADAPDOOR is designed to operate during the agent's training phase, its attack efficacy degrades modestly (-9.3%) in post-training scenarios, where the adversary attempts to inject backdoors into agents that have already completed benign task training. A similar decline in post-training performance has also been observed in other attack methods. We infer that this degradation is due to limited policy plasticity [12], [47], [48], as the activation pathways used to represent the benign task are widespread, reducing the agent's parameter flexibility. Constructing backdoor pathways in such contexts is challenging. This difficulty is a key factor affecting action-level backdoor threats in the DRL supply chain scenario and warrants further investigation.

VIII. RELATED WORK

In addition to backdoor attacks, the deployment of DRL systems faces a wide range of security and privacy threats [49]–[51]. This section outlines the principal threat vectors in DRL and reviews representative studies tackling these issues.

Adversarial Perturbations. Inspired by adversarial examples [52], one of the most widely studied attack strategies in DRL involves injecting adversarial perturbations into the environment or the victim agent's observations [53]–[56], thereby disrupting the agent's sequential decision-making process. Furthermore, recent research has explored attacks that directly manipulate the victim's action outputs. For instance, Lee *et al.* [57] proposed two action manipulation attacks: the myopic action-space attack injects action perturbations based on current observations, while the look-ahead action-space attack considers future steps to maximize the attack's impact. However, directly manipulating the victim agent's action outputs is impractical; thus, adversarial policies have been introduced to overcome this limitation.

Adversarial Policies. Gleave *et al.* [58] first introduced the concept of adversarial policies in the context of zero-sum games, later referred to as *Victim-play*, in which the adversary gains control over the opponent and manipulates its actions to guide the victim into making suboptimal decisions. Guo *et al.* [59] extended this concept to general-sum games, while Wu *et al.* [60] integrated explainable AI techniques to enhance the stealthiness of adversarial policies. Wang *et al.* [61] delved into adversarial policies in discrete action scenarios and successfully beat superhuman-level Go AIs, showcasing that near-Nash or ϵ-equilibrium policies are exploitable. Further, Ma *et al.* [34] and Liu *et al.* [62] demonstrated that even when adversaries only have partial observation privileges over the victim or partial control over the opponent, adversarial policies still pose a significant threat to DRL systems.

Poisoning Attacks. Poisoning the environment and reward signal in DRL has been extensively studied due to its significant impact on learning dynamics. Since the reward signal encodes the long-term objectives of a DRL task and directly guides policy optimization, manipulating it can effectively mislead the victim agent's behavior. Prior work [63]–[65] has shown that adversaries can poison the reward signal to deviate from the intended objectives. This attack strategy has been extended to the safety alignment in RLHF [66], [67], highlighting its relevance in real-world applications.

Copyright Protection. With the increasing commercial deployment of DRL in real-world applications, copyright protection has become an increasingly important concern, particularly in safeguarding policies, trajectories, and training environments. Chen *et al.* [35] introduced a temporal-based watermarking scheme that verifies the ownership of DRL policies by analyzing action probability distributions. Du *et al.* [68] proposed a trajectory-level auditing mechanism for offline RL, using cumulative rewards as intrinsic and stable fingerprints of datasets. Ye *et al.* [69] proposed two reinforcement unlearning approaches that enable agents to selectively forget knowledge acquired from specific training environments.

IX. CONCLUSION

In this paper, we propose ADAPDOOR, the first general framework for adaptive backdoor reward exploration in action-level backdoor attacks. We demonstrate that an adversary can achieve cross-task backdoor injection without relying on extensive trial-and-error and expert knowledge. ADAPDOOR performs a coarse initialization of the backdoor reward, followed by fine-grained adjustments based on the performance feedback. Extensive evaluations show that ADAPDOOR sig-

nificantly improves the effectiveness of action-level backdoors while maintaining stealthiness. ADAPDOOR reveals that action-level backdoors pose a more severe threat to real-world DRL applications than that disclosed in previous works, thereby warranting extensive attention and further research.

REFERENCES

[1] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes," in *AAAI*, 2025.

[2] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[3] N. Majidi, M. Shamsi, and F. Marvasti, "Algorithmic Trading Using Continuous Action Space Deep Reinforcement Learning," *Expert Systems with Applications*, 2024.

[4] H. Zhang, J. Gu, Z. Zhang, L. Du, Y. Zhang, Y. Ren, J. Zhang, and H. Li, "Backdoor Attacks against Deep Reinforcement Learning Based Traffic Signal Control Systems," *Peer-to-Peer Networking and Applications*, 2023.

[5] X. Chen, S. Feng, Z. Xiong, S. An, Y. Mao, G. Tao, W. Guo, and X. Zhang, "Temporal Logic-Based Multi-Vehicle Backdoor Attacks against Offline RL Agents in End-to-end Autonomous Driving," in *NeurIPS*, 2025.

[6] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-Level Drone Racing Using Deep Reinforcement Learning," *Nature*, 2023.

[7] P. Kiourti, K. Wardega, S. Jha, and W. Li, "TrojDRL: Evaluation of Backdoor Attacks on Deep Reinforcement Learning," in *DAC*, 2020.

[8] C. Ashcraft and K. Karra, "Poisoning Deep Reinforcement Learning Agents with In-Distribution Triggers," *arXiv*, 2021.

[9] Y. Chen, Z. Zheng, and X. Gong, "MARNet: Backdoor Attacks against Cooperative Multi-Agent Reinforcement Learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[10] J. Cui, Y. Han, Y. Ma, J. Jiao, and J. Zhang, "BadRL: Sparse Targeted Backdoor Attack against Reinforcement Learning," in *AAAI*, 2024.

[11] E. Rathbun, C. Amato, and A. Oprea, "SleeperNets: Universal Backdoor Poisoning Attacks against Reinforcement Learning Agents," in *NeurIPS*, 2024.

[12] O. Ma, R. Lin, Y. Dai, J. Chen, C. Zhou, D. L., and S. Ji, "Angel or Demon: Investigating the Plasticity Interventions' Impact on Backdoor Threats in Deep Reinforcement Learning," in *ICML*, 2026.

[13] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv*, 2016.

[14] Z. Yang, N. Iyer, J. Reimann, and N. Virani, "Design of Intentional Backdoors in Sequential Models," *arXiv*, 2019.

[15] L. Wang, Z. Javed, X. Wu, W. Guo, X. Xing, and D. Song, "BACKDOORL: Backdoor Attack against Competitive Reinforcement Learning," in *IJCAI*, 2021.

[16] C. Gong, Z. Yang, Y. Bai, J. He, J. Shi, K. Li, A. Sinha, B. Xu, X. Hou, D. Lo *et al.*, "BAFFLE: Backdoor Attack in Offline Reinforcement Learning," in *S&P*, 2024.

[17] S. Li, M. Zhang, O. Ma, K. Wei, and S. Ji, "Toobadrl: Trigger optimization to boost effectiveness of backdoor attacks on deep reinforcement learning," *arXiv*, 2025.

[18] Y. Dai, O. Ma, L. Zhang, X. Liang, X. Cao, S. Ji, J. Zhang, J. Huang, and L. Shen, "TrojanTO: Action-Level Backdoor Attacks against Trajectory Optimization Models," *ICLR*, 2026.

[19] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, "Distral: Robust Multitask Reinforcement Learning," *NIPS*, 2017.

[20] N. Vithayathil Varghese and Q. H. Mahmoud, "A Survey of Multi-Task Deep Reinforcement Learning," *Electronics*, 2020.

[21] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-Backdoor Learning: Training Clean Models on Poisoned Data," in *NeurIPS*, 2021.

[22] W. Chen, B. Wu, and H. Wang, "Effective Backdoor Defense by Exploiting Sensitivity of Poisoned Samples," in *NeurIPS*, 2022.

[23] C. Li, Z. Li, M. Pan, and X. Li, "TRAP: Mitigating Poisoning-based Backdoor Attacks by Treating Poison with Poison," *IEEE Transactions on Dependable and Secure Computing*, 2025.

[24] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving," in *CCS*, 2019.

[25] Y. Xu, X. Han, G. Deng, J. Li, Y. Liu, and T. Zhang, "SoK: Rethinking Sensor Spoofing Attacks against Robotic Vehicles from a Systematic View," in *EuroS&P*, 2023.

[26] T. Gu, K. Liu, B. Dolan-Gavitt, J. Smith, and A. Johnson, "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks," *IEEE Access*, 2019.

[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2018.

[28] E. Coumans and Y. Bai, "PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning," http://pybullet.org, 2021.

[29] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, and P. Abbeel, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," in *NIPS*, 2017.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv*, 2017.

[31] T. P. Lillicrap, "Continuous Control with Deep Reinforcement Learning," in *ICLR*, 2016.

[32] S. Huang, R. F. J. Dossa, A. Raffin, A. Kanervisto, and W. Wang, "The 37 Implementation Details of Proximal Policy Optimization," in *ICLR Blog Track*, 2022.

[33] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, 2021.

[34] O. Ma, Y. Pu, L. Du, Y. Dai, R. Wang, X. Liu, Y. Wu, and S. Ji, "SUB-PLAY: Adversarial Policies against Partially Observed Multi-Agent Reinforcement Learning Systems," in *CCS*, 2024.

[35] K. Chen, S. Guo, T. Zhang, S. Li, and Y. Liu, "Temporal Watermarks for Deep Reinforcement Learning Models," in *AAMAS*, 2021.

[36] T. Eiter and H. Mannila, "Computing Discrete Fréchet Distance," 1994.

[37] Y. Yu, X. Yin, J. Wang, T. C., X. S., Z. Q., and Z. D., "BehaviorGuard: Online Backdoor Defense for Deep Reinforcement Learning," *arXiv*, 2026.

[38] X. Chen, W. Guo, G. Tao, X. Zhang, and D. Song, "BIRD: Generalizable Backdoor Detection and Removal for Deep Reinforcement Learning," in *NeurIPS*, 2023.

[39] J. Guo, A. Li, L. Wang, and C. Liu, "PolicyCleanse: Backdoor Detection and Mitigation for Competitive Reinforcement Learning," in *ICCV*, 2023.

[40] W. Guo, Z. Yuan, J. Jia, B. Li, and D. Song, "SHINE: Shielding Backdoors in Deep Reinforcement Learning," in *ICML*, 2024.

[41] Z. Sha, X. He, P. Berrang, M. Humbert, and Y. Zhang, "Fine-Tuning Is All You Need to Mitigate Backdoor Attacks," *arXiv*, 2022.

[42] B. H. Korte and J. Vygen, *Combinatorial Optimization*, 2011.

[43] X. Liu, O. Ma, W. Chen, Y. Xia, and Y. Zhou, "HDRS: A Hybrid Reputation System with Dynamic Update Interval for Detecting Malicious Vehicles in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[44] O. Ma, X. Liu, and Y. Xia, "ABM-V: An Adaptive Backoff Mechanism for Mitigating Broadcast Storm in VANETs," *IEEE Transactions on Vehicular Technology*, 2023.

[45] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision Transformer: Reinforcement Learning via Sequence Modeling," in *NeurIPS*, 2021.

[46] Y. Dai, O. Ma, L. Zhang, X. Liang, S. Hu, M. Wang, S. Ji, J. Huang, and L. Shen, "Is Mamba Compatible with Trajectory Optimization in Offline Reinforcement Learning?" in *NeurIPS*, 2024.

[47] S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, "Loss of Plasticity in Deep Continual Learning," *Nature*, 2024.

[48] T. Klein, L. Miklautz, K. Sidak, C. Plant, and S. Tschiatschek, "Plasticity Loss in Deep Reinforcement Learning: A Survey," *arXiv*, 2024.

[49] S. Maiti, A. Balabhaskara, S. Adhikary, I. Koley, and S. Dey, "Targeted Attack Synthesis for Smart Grid Vulnerability Analysis," in *CCS*, 2023.

[50] J. Yu, W. Guo, Q. Qin, G. Wang, T. Wang, and X. Xing, "AIRS: Explanation for Deep Reinforcement Learning based Security Applications," in *USENIX Security*, 2023.

[51] Y. Xia, X. Liu, J. Ou, and O. Ma, "RLID-V: Reinforcement Learning-Based Information Dissemination Policy Generation in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[52] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *S&P*, 2017.

[53] V. Behzadan and A. Munir, "Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks," in *MLDM*, 2017.

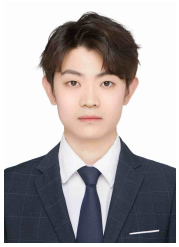
[54] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial Attacks on Neural Network Policies," *arXiv*, 2017.

[55] J. Sun, T. Zhang, X. Xie, L. Ma, Y. Zheng, K. Chen, and Y. Liu, "Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning," in *AAAI*, 2020.
[56] J. Tu, T. Wang, J. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Adversarial Attacks on Multi-Agent Communication," in *ICCV*, 2021.
[57] X. Y. Lee, S. Ghadai, K. L. Tan, C. Hegde, and S. Sarkar, "Spatiotemporally Constrained Action Space Attacks on Deep Reinforcement Learning Agents," in *AAAI*, 2020.
[58] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial Policies: Attacking Deep Reinforcement Learning," in *ICLR*, 2020.
[59] W. Guo, X. Wu, S. Huang, and X. Xing, "Adversarial Policy Learning in Two-Player Competitive Games," in *ICML*, 2021.
[60] X. Wu, W. Guo, H. Wei, and X. Xing, "Adversarial Policy Training against Deep Reinforcement Learning," in *USENIX Security*, 2021.
[61] T. T. Wang, A. Gleave, T. Tseng, K. Pelrine, N. Belrose, J. Miller, M. D. Dennis *et al.*, "Adversarial Policies Beat Superhuman Go AIs," in *ICML*, 2023.
[62] X. Liu, S. Chakraborty, Y. Sun, and F. Huang, "Rethinking Adversarial Policies: A Generalized Attack Formulation and Provable Defense in RL," in *ICLR*, 2024.
[63] A. Rakhsha, X. Zhang, X. Zhu, and A. Singla, "Reward Poisoning in Reinforcement Learning: Attacks against Unknown Learners in Unknown Environments," *arXiv*, 2021.
[64] M. Mohammadi, J. Nöther, D. Mandal, A. Singla, and G. Radanovic, "Implicit Poisoning attacks in Two-Agent Reinforcement Learning: Adversarial Policies for Training-Time Attacks," in *AAMAS*, 2023.
[65] J. Li, B. Zhang, and J. Wu, "Online Poisoning Attack against Reinforcement Learning under Black-box Environments," *arXiv*, 2024.
[66] P. Pathmanathan, S. Chakraborty, X. Liu, Y. Liang, and F. Huang, "Is Poisoning a Real Threat to LLM Alignment? Maybe More so Than You Think," in *AAAI*, 2025.
[67] T. Baumgärtner, Y. Gao, D. Alon, and D. Metzler, "Best-of-Venom: Attacking RLHF by Injecting Poisoned Preference Data," in *COLM*, 2024.
[68] L. Du, M. Chen, M. Sun, S. Ji, P. Cheng, J. Chen, and Z. Zhang, "ORL-AUDITOR: Dataset Auditing in Offline Deep Reinforcement Learning," in *NDSS*, 2024.
[69] D. Ye, T. Zhu, C. Zhu, D. Wang, K. Gao, Z. Shi, S. Shen, W. Zhou, and M. Xue, "Reinforcement Unlearning," in *NDSS*, 2025.

Oubo Ma received his B.E. degree from Wenzhou University in 2019, and the M.S. degree from HangZhou Normal University in 2022. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University. His research interests include LLM security and RL security.

Linkang Du received his B.E. and Ph.D. degrees from Zhejiang University in 2018 and 2023, respectively. He is currently an assistant professor at the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His research interests include privacy-preserving computing and trustworthy machine learning.

Yang Dai received his B.S. degree from Harbin Engineering University in 2022. He is currently pursuing the Ph.D. degree in National University of Defense Technology. His work primarily focuses on offline reinforcement learning, adversarial attacks, and backdoor attacks, aiming to evaluate and enhance the security of artificial intelligence systems in complex environments.



Chunyi Zhou received the Ph.D. degree with the School of Computer Science and Engineering from Nanjing University of Science and Technology in 2024. He is currently a Post-Doctor collaborating with Prof. Shouling Ji at the College of Computer Science and Technology, Zhejiang University. His research interests include AI security, federated learning, and machine unlearning.

Qingming Li received the B.S. degree from Xidian University in 2017, and the Ph.D. degree from Tsinghua University in 2022. She was a Research Assistant with the Research Institute of Artificial Intelligence, Zhejiang Lab, Hangzhou, China from 2022 to 2024. Since Apr. 2024, she has been a Postdoc with the College of Computer Science and Technology, Zhejiang University. Her research interests include deep learning security, large language models, and embodied agents.

Yuwen Pu received his Ph.D. in the School of Big Data and Software Engineering from Chongqing University in 2021. He is an associate professor in the School of Big Data and Software Engineering at Chongqing University. His research interests include AI security, data security, and privacy-preserving.

Shouling Ji (Senior Member, IEEE) is a Qiushi Distinguished Professor in the College of Computer Science and Technology at Zhejiang University. He received a Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology, and a Ph.D. degree in Computer Science from Georgia State University. His current research interests include AI Security, Software and System Security, and Data-driven Security and Privacy. He is a member of ACM, a senior member of IEEE, and a senior member of CCF. He is the recipient of the 2012 Chinese Government Award for Outstanding Self-Financed Students Abroad and 10 Best/Outstanding Paper Awards, including IEEE S&P 2025 Distinguished Paper Award and ACM CCS 2021 Best Paper Award.