BehaviorGuard: Online Backdoor Defense for Deep Reinforcement Learning

Yinbo Yu[1], **Xueyu Yin**[2], **Jiadai Wang**[2], **Chunwei Tian**[3], **Sai Xu**[4*], **Qi Zhu**[1], **Daoqiang Zhang**[1]
[1]College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics
[2]School of Cybersecurity, Northwestern Polytechnical University
[3]School of Computer Science and Technology, Harbin Institute of Technology
[4]Department of Electronic and Electrical Engineering, University College London
yinboyu@nuaa.edu.cn, xueyuy@mail.nwpu.edu.cn, wangjiadai@nwpu.edu.cn, chunweitian@hit.edu.cn, sai.xu@ucl.ac.uk, zhuqinuaa@163.com, dqzhang@nuaa.edu.cn

Abstract

Backdoor attacks pose a serious threat to deep reinforcement learning (DRL). Current defenses typically rely on reward anomalies to reverse-engineer triggers and model finetuning to remove backdoors. However, complex trigger patterns undermine their robustness, and fine-tuning entails high costs, limiting practical utility. Therefore, we shift defense concerns to trigger-agnostic backdoor output behaviors and propose BehaviorGuard, an online behavior-based backdoor detection and mitigation framework for DRL. Specifically, we find that regardless of attacks, backdoored policies induce consistent shifts in action distributions to ensure reliable activation, leaving detectable traces in high-quantile regions and distribution tails, even in the absence of triggers. Based on this, we design a novel metric that captures behavioral drift in action distributions to identify and suppress backdoor actions at runtime. To our knowledge, this is the first online backdoor defense that counters attacks both in single- and multi-agent DRL. Evaluated across diverse benchmarks with different backdoor attacks, BehaviorGuard consistently surpasses prior methods in both efficacy and efficiency.

1 Introduction

Deep reinforcement learning (DRL) has shown strong performance in complex sequential decision-making tasks and is starting to be used in safety-critical domains such as games [Vinyals *et al.*, 2019], autonomous driving [Wang *et al.*, 2021b], and robotics [Tang *et al.*, 2025]. As DRL moves into such settings, security and safety become central concerns [Gu *et al.*, 2024]. Recent studies [Kiourti *et al.*, 2019; Wang *et al.*, 2021a] have revealed that DRL policies are highly vulnerable to backdoor attacks, where an adversary implants a hidden backdoor during policy training that causes the agent to behave maliciously once the backdoor is activated, while maintaining benign performance otherwise. Compared to backdoors in supervised learning [Gu *et al.*, 2017; Liu *et al.*, 2018], backdoors in DRL are particularly dangerous due to the sequential nature of decision making and complex environment dynamics, which can amplify small behavior shifts into catastrophic outcomes.

This threat is further amplified in multi-agent reinforcement learning (MARL) settings. In competitive environments, an adversary may trigger a victim's backdoor purely through its own actions or interaction patterns [Wang *et al.*, 2021a], whereas in cooperative environments, poisoning a single agent can compromise the entire team through coordination failure and cascading effects [Chen *et al.*, 2022]. The inherent non-stationarity and inter-agent coupling in MARL significantly increase the difficulty of detecting and mitigating backdoors, making defenses developed for single-agent settings unreliable when transferred across environments.

Existing defenses against DRL backdoors primarily rely on reward anomalies, trigger reconstruction, or prior knowledge of trigger patterns [Guo *et al.*, 2023; Chen *et al.*, 2023; Yuan *et al.*, 2024]. Although these methods show high defensive capabilities, they still face several challenges: (1) reward signals may appear normal when the backdoor is active [Rathbun *et al.*, 2025], resulting in a high detection false-negative rate; (2) pattern-dependent methods fail against triggers based on sequential behaviors with spatial or temporal dependence [Yu *et al.*, 2024a]; (3) most methods lack a unified treatment that generalizes across both single- and multi-agent scenarios; and (4) many methods require expensive policy retraining to mitigate backdoors.

To address these challenges, in this work, we shift our defense focus from recognizing diverse and attack-specific trigger patterns targeted by prior work to identifying the unified backdoor output manifestation. Specifically, we analyze the output behavioral differences between clean and backdoor DRL policies. Our key observation is that, regardless of trigger patterns or environments, backdoored policies inevitably induce subtle but persistent behavioral drift in the action distribution. Even under trigger-free inputs, the policy must bias its action probabilities to ensure successful activation, leaving detectable signatures in high quantiles and tails of the distribution. This insight enables trigger-agnostic backdoor defense without relying on reward signals or trigger priors.

Building on this finding, we propose **BehaviorGuard**[1], a

*Corresponding Author

[1]https://github.com/c0d818/BehaviorGuard

unified framework that jointly detects and mitigates backdoors online for different DRL scenarios. We first design a backdoor detection method based on the signature of behavioral drift, which quantifies action-distribution deviations using the Behavioral Drift Score (BDS) and density-based statistics. Second, we propose a lightweight, retraining-free, drift-constrained backdoor mitigation strategy to mitigate backdoor actions online at runtime. Once backdoor outputs are detected, BehaviorGuard suppresses backdoor behaviors by projecting the abnormal action distribution back to a low-drift region, achieving practical protection without retraining. To the best of our knowledge, this is the first work that can detect and mitigate backdoor attacks against single- and multi-agent DRL at runtime.

Our contributions are summarized as follows:

- We identify behavioral drift as an attack- and environment-agnostic Trojan signature.
- We propose a unified backdoor detection framework for single- and multi-agent environments.
- We introduce a lightweight drift-constrained mitigation method that suppresses backdoor behaviors at inference time without retraining, enabling online defense.
- Extensive experiments across single-, cooperative, and competitive multi-agent benchmarks demonstrate that BehaviorGuard achieves powerful detection performance while effectively mitigating backdoor impact with minimal degradation on clean performance.

2 Background and Related Work

2.1 Deep Reinforcement Learning

Single-agent RL (SARL) is commonly modeled as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where S and A denote the state and action spaces, $P(s'|s,a)$ is the transition kernel, $r(s,a)$ is the immediate reward, and $\gamma \in (0,1)$ is the discount factor. At time t, the agent observes a state (or observation) x_t and samples an action $a_t \sim \pi_\theta(\cdot|x_t)$; the environment then transitions to s_{t+1} and returns reward r_t. The policy π_θ is typically parameterized by DNN, and is trained to maximize the expected discounted return $\mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

Beyond the single-agent setting, we also consider multi-agent RL (MARL) [Busoniu *et al.*, 2008]. In competitive MARL (com-MARL), agents' rewards are competitive or conflicting, and the environment evolves under the joint action $a_t = (a_t^1, \ldots, a_t^n)$ [Lowe *et al.*, 2017]. In cooperative MARL (coo-MARL), agents share a team objective (e.g., a shared team reward), where coordination among policies is essential [Rashid *et al.*, 2020; Foerster *et al.*, 2018].

2.2 Backdoor Attack Formalization in DRL

Backdoor attacks aim to implant a hidden malicious behavior into a policy network such that it behaves similarly to a clean policy under normal inputs, while reliably switching to an attacker-specified behavior when a trigger is present. Formally, let δ denote the trigger and $f_\delta(\cdot)$ be the trigger injection function that applies the trigger pattern to an input: $x_t^\delta = f_\delta(x_t)$. Backdoor triggers can be classified into two types: **instant** and **sequential** triggers. The former can be instantiated as single-step state perturbations (e.g., visual patches [Kiourti *et al.*, 2019] or specific states [Wang *et al.*, 2021b]). The latter can be temporal state patterns [Yu *et al.*, 2022] or interaction-based conditions among agents [Wang *et al.*, 2021a; Fang *et al.*, 2025] spanning multiple steps. The attacker obtains a backdoored policy π^{bd} via poisoning the training process such that π^{bd} mimics a clean policy π^{cl} on trigger-free inputs but shifts toward a target behavior π^{tar} on triggered inputs:

$$\pi^{bd}(\cdot|x) \approx \pi^{cl}(\cdot|x), \quad \pi^{bd}(\cdot|f_\delta(x)) \approx \pi^{tar}(\cdot|f_\delta(x)). \quad (1)$$

In SARL, this switch often manifests as a sharp return drop or persistent target actions; in com-MARL, the attack may reduce the victim's win rate or induce systematic failures in key adversarial states; in coo-MARL, deviations of one agent can further disrupt coordination and degrade team performance.

2.3 Backdoor Attacks in DRL

Backdoor Attacks in SARL. TrojDRL [Kiourti *et al.*, 2019] is the first to show backdoor attacks on deep RL agents, demonstrating that a small amount of poisoned data and reward tweaks could implant malicious behavior. Yu et al. [Yu *et al.*, 2022] introduce a new DRL backdoor whose trigger is a set of sequential observations, making it stealthy and persistent. BadRL [Cui *et al.*, 2024] uses highly sparse poisoned samples during training and testing, maintaining a high attack success rate while reducing cost. BAFFLE [Gong *et al.*, 2024] implants a backdoor by poisoning offline RL datasets.

Backdoor Attacks in com-MARL. BACKDOORL [Wang *et al.*, 2021a] uses a sequence of actions by the opponent as a trigger, so that the adversary can activate the victim agent's backdoor through its own actions. SCAB [Liu *et al.*, 2025] injects backdoors via pre-trained agents interacting with legitimate actions without the need for observation perturbations or reward modification.

Backdoor Attacks in coo-MARL. MARNet [Chen *et al.*, 2022] successfully implements backdoor attacks on coo-MARL systems by designing trigger, action poisoning, and reward hacking modules, causing the attacker's agent to behave abnormally in a malicious environment. One4all[Zheng *et al.*, 2023] presents two poisoning attack techniques (SNPA and TAPA), which covertly poison the coo-MARL system by manipulating a single agent's observations or reward functions. BLAST [Yu *et al.*, 2024b; Fang *et al.*, 2025] introduces a unilateral filter that enables a single backdoor agent to quickly affect other clean agents, thereby enabling a backdoor leverage attack on the entire multi-agent system.

2.4 Backdoor Defenses in DRL

[Acharya *et al.*, 2023] detects backdoor by analyzing discrepancies in how the policy evaluates state observations. [Vyas *et al.*, 2024] presents a runtime detection method based on neural activation patterns, effectively detecting subtly hidden backdoor triggers. These methods are effective, but cannot mitigate detected backdoors. PolicyCleanse [Guo *et al.*, 2023] detects Trojan agents in com-MARL according to abnormal accumulated rewards and mitigates the backdoor via machine unlearning. BIRD [Chen *et al.*, 2023] and SHINE [Yuan *et al.*, 2024] leverage trigger restoration and policy

finetuning to detect and remove backdoors in DRL policies without requiring attack specifications. These methods all rely on reward anomalies and expensive policy retraining to detect and mitigate backdoors, which results in potential missed detection risks and high defense overhead. Bharti et al. [Bharti *et al.*, 2022] proposed a provable, retraining-free backdoor defense mechanism that projects observations onto a safe subspace to suppress backdoor activation. However, it only supports the single-agent scenario.

Different from current methods, our BehaviorGuard is based on a novel behavioral drift metric without trigger priors, reward signals, or policy retraining, and can be deployed to defend against backdoors online. Moreover, it is effective against different trigger patterns, instant or sequential behaviors, environments (SARL, com-MARL, or coo-MARL), and retains defense power even when rewards appear normal.

3 Method

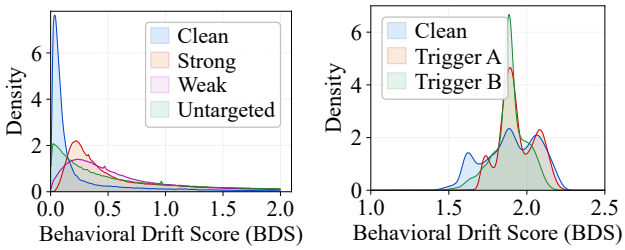
3.1 Threat Model

Attacker's Capabilities and Goal. The attacker has complete control over the training process, data, and reward functions, and can inject any backdoors with instant or sequential triggers into policies. These policies can strive to achieve their benign objectives, but once the backdoor is activated, they will immediately perform backdoor objectives.

Defender's Capabilities and Goal. The defender can only download or outsource the final policy, which may potentially contain a backdoor. We assume the defender does not have the resources to retrain policies. This enables higher practicality. The defender aims to detect whether the policy contains a backdoor and, if so, mitigate its impact.

3.2 Key Intuition

One key observation across different backdoor attacks (e.g., TrojDRL[Kiourti *et al.*, 2019] and BACKDOORL[Wang *et al.*, 2021a]) is that their policies systematically reshape the action distribution, leaving detectable *behavioral drifts*. As shown in Fig. 1, which compares the BDS distribution of clean policies and those injected with 3 types of TrojDRL attacks in SARL and 2 types of BACKDOORL attacks in com-MARL, these drifts manifest in distinct patterns. In SARL, the clean BDS distribution is highly concentrated, owing to the stability of the single-agent environment. Backdoor attacks undermine this consistency, resulting in dispersed distributions with heavier tails. In contrast, in the non-stationary environment of com-MARL, a clean policy naturally exhibits a wider BDS distribution due to adversarial interactions. Intriguingly, BACKDOORL counteracts this by enforcing a fixed, cooperative strategy to achieve its attack goal, leading to an abnormally concentrated BDS distribution. In summary, despite variations in its directional pattern across environments, *the behavioral drift induced by the inherent conflict between primary and backdoor tasks serves as a robust, universal clue for backdoor defense*.



(a) SARL: Breakout (b) com-MARL: sumo-humans

Figure 1: Behavioral drift comparison of clean and backdoor policies in single- and multi-agent environments.

3.3 Behavioral Drift Detection for Backdoor Detection

Based on the above observation, in BehaviorGuard, we design a behavioral drift detection (BDD) method based on BDS to detect backdoors against both SARL and MARL. However, their underlying mechanisms and observable characteristics differ substantially. Hence, to maintain conceptual clarity and analytical focus, here, we mainly describe our designs of BDD for SARL and leave its extensions for coo- and com-MARL in our full version [Yu *et al.*, 2026].

In the single-agent scenario, once the backdoor is activated, it will induce systematic shifts in the policy's action distributions under attack conditions, while the policy remains indistinguishable from a clean model on benign inputs. BDD aims to identify these attacks by quantifying deviations between the policy's runtime behavior and its expected benign statistics. Given a trajectory of length T_s, BDS at each timestep t can be calculated as:

$$\mathrm{BDS}_t = \frac{1}{2}\sum_{k=1}^{K}\left(\frac{p_{t,k}-\mu_k}{\sigma_k+\epsilon}\right)^2, \tag{2}$$

where $p_{t,k}$ denotes the predicted probability of the k-th action at timestep t, and (μ_k, σ_k) are the corresponding mean and standard deviation estimated from clean executions. The constant ϵ is used for numerical stability.

To characterize the distribution of behavioral drift over the entire execution, we collect the sequence $\{\mathrm{BDS}_t\}_{t=1}^{T_s}$ and estimate its probability density function using KDE:

$$\hat{f}(\mathrm{BDS}_t) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{\mathrm{BDS}_t - \mathrm{BDS}_i}{h}\right), \tag{3}$$

where $K(\cdot)$ denotes a Gaussian kernel, h is the bandwidth parameter, and $\{\mathrm{BDS}_i\}_{i=1}^{N}$ are the observed drift samples used for density estimation. Based on the estimated density, we define a *tail statistic* that measures the proportion of timesteps exhibiting rare, low-density deviations:

$$\mathrm{tail} = \frac{1}{T_s}\sum_{t=1}^{T_s}\mathbf{1}\left[\hat{f}(\mathrm{BDS}_t) < \tau_{\mathrm{tail}}\right], \tag{4}$$

where τ_{tail} is defined as the density value corresponding to the low-density tail of the BDS distribution. It is calibrated

Algorithm 1 Behavioral Drift Detection of BehaviorGuard

Require: Action probabilities $\{p_{t,k}\}_{t=1}^{T_s}$; benign stats (μ_k, σ_k); KDE (K, h); density threshold τ_{tail}; weights (w_1, w_2); decision threshold τ
Ensure: Detection result
Initialize $\mathcal{B} \leftarrow [\,]$
for $t = 1$ to T_s **do**
  $\text{BDS}_t \leftarrow \frac{1}{2}\sum_{k=1}^{K}\left(\frac{p_{t,k}-\mu_k}{\sigma_k+\epsilon}\right)^2$
  Append BDS_t to $\mathcal{B}$
end for
for $t = 1$ to T_s **do**
  $\hat{f}(\mathcal{B}[t]) \leftarrow \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{\mathcal{B}[t]-\mathcal{B}[i]}{h}\right)$
end for
tail $\leftarrow \frac{1}{T_s}\sum_{t=1}^{T_s} \mathbf{1}\left[\hat{f}(\mathcal{B}[t]) < \tau_{\text{tail}}\right]$
$\overline{\text{BDS}} \leftarrow \frac{1}{T_s}\sum_{t=1}^{T_s}\mathcal{B}[t]$
Compute $\widehat{\text{BDS}}$ according to Eq. (6)
if $\widehat{\text{BDS}} > \tau$ **then**
  return "Potential Backdoor Detected"
else
  return "No Backdoor Detected"
end if

by fitting a KDE to the clean baseline BDS_t values and setting τ_{tail} to the 5th percentile of the estimated densities $\hat{f}(\text{BDS}_t)$.

We further aggregate the drift magnitude over the trajectory by computing the mean drift score $\overline{\text{BDS}} = \frac{1}{T_s}\sum_{t=1}^{T_s}\text{BDS}_t$. Both $\overline{\text{BDS}}$ and tail are normalized using Z-scores:

$$z_{\text{BDS}} = \frac{\overline{\text{BDS}} - \mu_{\text{BDS}}}{\max(\sigma_{\text{BDS}}, \epsilon)}, \quad z_{\text{tail}} = \frac{\text{tail} - \mu_{\text{tail}}}{\max(\sigma_{\text{tail}}, \epsilon)}. \tag{5}$$

Finally, they are combined into a composite detection score:

$$\widehat{\text{BDS}} = \frac{1}{\sqrt{2}}(z_{\text{BDS}} + z_{\text{tail}}), \tag{6}$$

where we use $\frac{1}{\sqrt{2}}$ to keep $\widehat{\text{BDS}}$'s range being consistent with BDS. A policy execution is flagged as potentially backdoored when $\widehat{\text{BDS}}$ exceeds a predefined threshold τ. We set τ as the 95th percentile of the composite score distribution computed from clean baseline executions, and fix it for all subsequent backdoor detection. The complete detection procedure is summarized in Algorithm 1.

The above detection formulation is designed for single-agent policies, and extending it to multi-agent settings requires additional considerations. In *competitive* multi-agent environments, backdoor behaviors often emerge through interaction-dependent deviations, where an agent's abnormal actions are amplified by the opponent's responses. In contrast, *cooperative* multi-agent scenarios exhibit correlated behavior shifts across agents, in which backdoor activation may manifest as synchronized or role-specific deviations. These characteristics render direct application of single-agent BDD insufficient and require dedicated designs that account for inter-agent dependency and joint behavioral dynamics. Those details of the extended BDD are described in [Yu *et al.*, 2026].

Algorithm 2 Drift-Constrained Backdoor Action Mitigation

Require: Victim policy π_θ; drift detector producing score BDS_t; threshold τ_D;
reference action sequence $\mathcal{A}^{\text{ref}} = \{a_t^{\text{ref}}\}_{t=1}^{T}$; window length L; guard probability p
Ensure: Executed action $\tilde{a}_t$
Initialize drift indicator buffer $\{z_i\}$ with length L to zeros
for $t = 1, 2, \ldots$ **do**
  Observe current state s_t
  Compute raw action $a_t \leftarrow \arg\max_a \pi_\theta(a \mid s_t)$
  Obtain reference action $a_t^{\text{ref}} \leftarrow \mathcal{A}^{\text{ref}}[t \bmod T]$
  Obtain drift score BDS_t from detector
  $z_t \leftarrow \mathbb{I}(\text{BDS}_t > \tau_D)$
  Update buffer and compute $c_t \leftarrow \sum_{i=\max(1,t-L+1)}^{t} z_i$
  $g_t \leftarrow \mathbb{I}(c_t \geq L)$
  if $g_t = 1$ **and** $\text{Unif}(0,1) < p$ **then**
    $\tilde{a}_t \leftarrow a_t^{\text{ref}}$
  else
    $\tilde{a}_t \leftarrow a_t$
  end if
  Execute action $\tilde{a}_t$ and observe next state
end for

3.4 Behavioral-Drift–Constrained Mitigation

In the spirit of runtime shielding for RL [Alshiekh *et al.*, 2018], we further design an online backdoor mitigation method in BehaviorGuard based on behavioral drift, which constrains policy execution by leveraging the output of BDD. This method is summarized in Algorithm 2. Without requiring trigger priors or policy retraining, our method intervenes online to suppress abnormal actions induced by persistent deviations from normal policy execution.

In the single-agent environment, we assume that typical action distribution characteristics under benign conditions can be approximated from limited offline interactions or historical execution logs, and use them as a normal behavior prior that remains stable under similar environment configurations. Formally, the prior is represented as a time-indexed set of reference actions:

$$\mathcal{A}^{\text{ref}} = \{a_t^{\text{ref}}\}_{t=1}^{T}, \tag{7}$$

where a_t^{ref} denotes the most representative action at timestep t under benign conditions (e.g., the mode of the action distribution). During policy inference, the reference action is retrieved via modular indexing

$$a_t^{\text{ref}} = \mathcal{A}^{\text{ref}}[t \bmod T], \tag{8}$$

which provides a lightweight behavioral reference without requiring exact state alignment.

At each timestep, BDD outputs a score BDS_t. When it exceeds a predefined threshold τ_D, the output is considered to deviate from normal behavior. τ_D is calibrated offline as the 95th percentile of the clean baseline executions' BDS distribution, thus filtering out innocuous noise. To reduce false triggers caused by stochasticity or transient noise, we smooth the detection results over time by defining a drift indicator:

$$z_t = \mathbb{I}(\text{BDS}_t > \tau_D), \tag{9}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. Drift occurrences are then accumulated within a temporal window of length L:

$$c_t = \sum_{i=\max(1,t-L+1)}^{t} z_i, \tag{10}$$

where L controls how many consecutive drift detections are required before mitigation is triggered. When $c_t \geq L$, the policy is regarded as entering a persistent abnormal behavior regime, and a mitigation gate $g_t = \mathbb{I}(c_t \geq L)$ is activated.

At timestep t, the policy produces a raw action $a_t = \arg\max_a \pi_\theta(a \mid s_t)$. If mitigation is not triggered ($g_t = 0$), the raw action is executed. Otherwise, when persistent behavioral drift is detected ($g_t = 1$), the action is corrected with probability p by projecting it back to the reference action:

$$\tilde{a}_t = \begin{cases} a_t^{\text{ref}}, & \text{if } g_t = 1 \text{ and } u < p, \\ a_t, & \text{otherwise,} \end{cases} \qquad u \sim \text{Unif}(0,1). \tag{11}$$

p is a mitigation probability. This probabilistic correction avoids deterministic intervention at all timesteps and allows the mitigation strength to be controlled via parameters p and L, thereby suppressing abnormal behavior while preserving benign policy performance.

Overall, BehaviorGuard forms a unified closed loop between detection and mitigation: BDD determines whether the policy has entered an abnormal regime, while action constraints intervene at the execution level only when persistent drift is confirmed. This design enables effective and controllable backdoor suppression with a low computational overhead, without policy retraining or trigger knowledge. Extensions to multi-agent environments based on the same principle are provided in [Yu *et al.*, 2026].

4 Evaluation

4.1 Experiment Setup

Evaluation Environments. We evaluate the performance of BehaviorGuard against backdoor attacks in discrete-action settings (SARL and coo-MARL) and a continuous-action setting (com-MARL).

SARL. We follow TrojDRL [Kiourti *et al.*, 2019] and consider three attack variants: Strong-Targeted, Weak-Targeted, and Untargeted. Experiments are conducted on four Atari games from OpenAI Gym [Brockman *et al.*, 2016] and ALE [Bellemare *et al.*, 2013]: Breakout, Pong, Seaquest, and SpaceInvaders. These tasks exhibit diverse dynamics and action-selection patterns, allowing us to evaluate detection behavior across heterogeneous state distributions.

com-MARL. We adopt BACKDOORL [Wang *et al.*, 2021a], which targets adversarial interaction settings, and evaluate on three tasks from the competitive benchmark suite [Bansal *et al.*, 2017]: Sumo-Humans, Run-To-Goal, and You-Shall-Not-Pass. The strategic coupling and non-stationarity induced by opponent behaviors make this setting particularly challenging for trigger-agnostic detection.

coo-MARL. We evaluate against the MARNet attack [Chen *et al.*, 2022] on two representative coo-MARL algorithms (QMIX [Rashid *et al.*, 2018] and COMA [Foerster *et al.*, 2018]), which use specific textures as their trigger.

We set the mitigation probability $p = 0.5$ and the window length $L = 5$ to default across all environments. For each task, we evaluate multiple clean and backdoored policies to account for training variability.

Metrics. We assess the proposed approach using key metrics. We use the average episode reward in SARL tasks and the winning rate in MARL tasks to compare different policies. We use the False Positive Rate (FPR) and its AUC, which reflects the proportion of false positives. We also introduce a mitigation rate (MR) to compare defense performance across heterogeneous environments, which is a normalized recovery ratio of a defended policy from the poisoned performance toward the clean one. These metrics provide a comprehensive evaluation of the method's performance and allow for comparison with baseline approaches. All evaluations were conducted with an RTX 5090 GPU.

Baselines. We compare against representative backdoor defense baselines. First, we implement direct retraining via PPO-based retraining [Schulman *et al.*, 2017] under the same environment and training budget for a fair comparison. Second, we include Neural Cleanse (NC) [Wang *et al.*, 2019], STRIP [Gao *et al.*, 2019], and FeatureRE [Wang *et al.*, 2022] as classic trigger-oriented defenses, adapted to RL policies by operating on state observations (as the model input) and flagging abnormal responses. We further include RL-specific defenses, including BIRD[Chen *et al.*, 2023], SHINE[Yuan *et al.*, 2024], and PD[Bharti *et al.*, 2022].

4.2 Experiment Results

Overall Defense Effectiveness

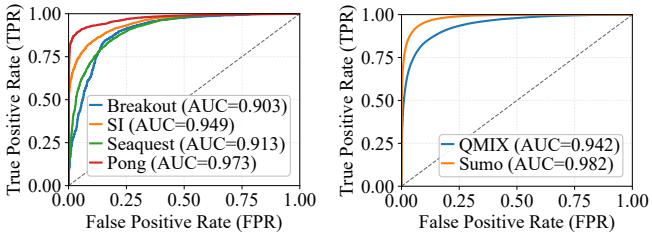
Table 1 demonstrates the defense effectiveness of our BehaviorGuard and 7 baselines under the clean and poisoned environments. Among the 3 classic defenses, STRIP outperforms NC and FeatureRE, mainly because the latter two suffer from low-fidelity trigger reconstruction. Direct retraining can partially mitigate backdoors in SARL and coo-MARL, but fails in all com-MARL environments. BIRD and SHINE defend against backdoors via RL-specific trigger restoration and fine-tuning, but SHINE has stronger defense performance due to its high-fidelity trigger reconstruction. PD, as a retraining-free defense, performs poorly in both clean and poisoned environments due to its defective projection on all input states. BehaviorGuard does not introduce negative effects on the policy in the clean environment and achieves defense performance comparable to SHINE in the poison environment. While SHINE can accurately restore spatial triggers in SARL and coo-MARL, it struggles with complex sequential triggers used in com-MARL. In contrast, BehaviorGuard is trigger-agnostic and does not rely on trigger recognition, enabling consistent robustness across diverse RL settings.

Detection Performance

We evaluated whether BehaviorGuard can reliably identify trigger activations. Fig. 2 demonstrates its detection quality for SARL and MARL via ROC curves and AUC values. Fig. 2(a) shows that BehaviorGuard achieves consistently strong discrimination on SARL Atari tasks, with ROC curves concentrated toward the upper-left region and AUC values above 0.90 across games, indicating reliable separa-

Environment	Methods	SARL				com-MARL			coo-MARL	
		Breakout	SpaceInvaders	Seaquest	Pong	YSNP(%)	SH(%)	RTGA(%)	QMIX(%)	COMA(%)
Clean	**Original**	445.3±125.2	722.2±248.2	1662.8±101.3	16.9±1.9	48.7±1.2	33.0±2.4	49.9±1.2	99.1±0.9	95.1±0.6
	Direct retraining	435.1±129.7	762.1±261.9	-	7.1±0.9	37.3±3.4	27.2±4.3	48.9±1.7	98.6±1.4	95.7±1.4
	NC	252.5±74.8	317.8±109.3	-	3.4±0.5	-	-	-	**98.8**±1.1	**96.1**±1.0
	FeatureRE	450.1±131.4	741.8±255.1	-	7.3±1.0	-	-	-	**98.8**±1.2	95.7±1.0
	STRIP	476.8±138.3	707.3±243.1	-	15.2±2.0	-	-	-	87.1±1.1	79.3±1.7
	PD	212.2±80.9	281.8±154.1	901.5±122.3	-	-	-	-	-	-
	BIRD	478.2±156.0	626.3±260.9	**1684.1**±162.8	-	44.1±1.8	31.3±1.3	44.4±5.9	93.5±3.7	91.2±4.5
	SHINE	**505.5**±147.6	**880.1**±302.5	-	18.2±2.3	48.4±2.0	**37.2**±3.0	**50.8**±1.4	98.7±1.5	95.8±0.9
	BehaviorGuard	465.2±108.1	775.1±263.3	1672.8±86.5	**18.5**±1.6	**51.0**±3.1	33.3±4.3	49.4±0.5	98.5±1.2	95.3±0.8
Poisoned	**Original**	33.1±81.7	114.1±236.1	141.2±46.4	2.1±3.7	18.3±2.6	15.9±1.0	18.8±2.3	21.9±0.2	35.7±0.7
	Direct retraining	309.0±156.7	752.0±280.9	-	3.0±3.4	27.7±4.1	27.8±3.1	37.4±1.2	82.6±1.1	82.3±3.2
	NC	23.2±69.0	396.7±183.6	-	0.1±4.2	-	-	-	25.9±0.4	21.6±0.5
	FeatureRE	31.0±102.0	509.5±195.0	-	4.9±3.0	-	-	-	33.2±0.6	24.7±2.2
	STRIP	444.6±161.4	796.2±301.7	-	15.6±1.8	-	-	-	85.3±1.2	71.3±2.7
	PD	220.7±77.3	428.9±192.3	825.1±173.5	-	-	-	-	-	-
	BIRD	404.0±140.9	653.2±244.8	1630.8±162.9	-	43.6±3.9	29.3±4.0	42.4±6.6	95.1±0.6	93.6±2.5
	SHINE	**570.9**±144.1	**951.5**±380.5	-	17.9±2.2	47.2±2.0	35.8±3.0	45.2±1.4	**98.6**±1.5	88.1±2.3
	BehaviorGuard	517.8±118.7	783.2±252.1	**1656.1**±106.7	**18.2**±1.5	**47.8**±1.1	**36.6**±2.6	**49.7**±1.1	96.9±1.7	**94.2**±1.1

Table 1: Defense performance. We report the average episode return for SARL tasks and the average winning rate (%) for MARL tasks over 1,000 rounds (higher is better). The best result is bolded, and green shading marks competitive results close to the best.



(a) Single-agent (b) Multi-agent

Figure 2: ROC curves of BehaviorGuard's backdoor detection in (a) SARL Atari tasks and (b) multi-agent benchmarks.

tion between clean and backdoored policies under heterogeneous dynamics. Fig. 2(b) further demonstrates that BehaviorGuard remains effective in multi-agent benchmarks, where interaction-induced non-stationarity and strategic coupling introduce substantial benign behavioral variability. Overall, these results indicate that our BDS provides a stable detection signal across single- and multi-agent systems.

Robustness Across Advanced Attacks

We further evaluate BehaviorGuard's defense effectiveness under four different backdoor attacks: (i) **Normal attacks** (macro-averaged from Tab. 1), (ii) **Reward-constrained attack** [Rathbun *et al.*, 2025], which enforces bounded backdoor reward modifications to implant backdoors; (iii) **Sequential attack in SARL** [Yu *et al.*, 2022], in which we use a patch to appear clockwise on the four corners of input images in 4 consecutive timesteps as a sequential trigger and perform backdoor actions after the 4 timesteps, and (iv) **Sequential attack in coo-MARL** [Yu *et al.*, 2024b; Fang *et al.*, 2025], whose triggers are defined by agent-wise sequential actions with spatial and temporal dependences. We use MR to compare defense performance across different tasks. MR$=1$ indicates a clean-level recovery, and MR>1 allows slight over-recovery due to regularization effects.

Method	Normal	Reward-Constrained	S-Attack in SARL	S-Attack in coo-MARL
BIRD	0.883±0.08	0.641±0.05	0.737±0.05	0.817±0.12
SHINE	**1.074**±0.19	0.722±0.13	0.875±0.16	0.840±0.06
Ours	1.054±0.09	**1.090±0.06**	**0.981**±0.07	**0.953**±0.05

Table 2: Defense robustness across diverse backdoor attacks. We report the average MR over 1,000 evaluation episodes per environment. S-Attack means sequential attack.

Table 2 compares robustness across attack types. Besides normal attacks, the other 3 advanced attacks suppress reward anomalies or embed triggers into sequential/interaction-dependent states, leading to low performance of trigger identification in BIRD and SHINE. BehaviorGuard achieves the most consistent robustness across attacks, matching strong baselines under normal attacks while notably outperforming existing methods under advanced attacks.

Time Cost of Backdoor Defense

In Table 3, we evaluated the defense time cost of SHINE, BIRD, PD, and BehaviorGuard in 4 single-agent environments. In each environment, following the standard workflow of these methods, we first ran a set of episodes (10000, 1000, 10, and 1000) with a maximum of 2,000 steps per episode to perform their offline operations and then ran 1,000 episodes to evaluate their online operations and defended policies. We measured the time cost of dominant phases in these methods. Existing methods require hours due to their expansive operations, including trigger restoration in SHINE, policy fine-tuning in SHINE and BIRD, and subspace sanitization and state projection in PD. BIRD's trigger restoration can converge early, thereby being faster than SHINE. PD and BehaviorGuard perform backdoor mitigation online, but the high computational cost of PD's online state projection limits its applicability for policy inference. In contrast, BehaviorGuard finishes in 35.03 minutes (18.81m for offline baseline calibration and 16.22m for online detection and mitigation), contributed by its lightweight drift scoring and action correction. In summary, BehaviorGuard significantly outperforms existing defenses in terms of both efficacy and time cost, and its

Method	Offline cost	Online cost	Total
None	–	13.75m	13.75m
BIRD	12.0m (Trigger restoration) + 1.13h (Policy finetuning)	13.76m	1.32h
SHINE	4.80h (Trigger restoration) + 1.95h (Policy finetuning)	13.60m	6.75h
PD	0.93h (Subspace sanitization)	1.29h	2.22h
Ours	18.81m (Baseline calibration)	16.22m	**35.03m**

Table 3: Time cost comparison. 'None' denotes the basic inference time of a backdoor policy.

online defense capability ensures practical applicability.

4.3 Ablation Studies

We conducted ablations to assess the effectiveness and robustness of BehaviorGuard under different configurations. Results in Fig. 3 are averaged over multiple independent runs in Atari environments.

Attack types. Fig. 3(a) compares BehaviorGuard's MR under 3 TrojDRL attacks. Its performance under targeted attacks is more stable than under untargeted attacks, especially under strong targeted attacks. This indicates that BehaviorGuard is more reliable when the abnormal behavior manifests as sustained drift targeted by most backdoor attacks, rather than isolated, stochastic deviations.

Guard probability p. Fig. 3(b) studies the guard probability p used in Eq. (11), which controls the strength of probabilistic action correction after persistent drift is detected. Setting $p = 0$ effectively disables mitigation and results in limited recovery, whereas moderate values (e.g., $p \in [0.25, 0.5]$) provide a clear improvement across environments. Further increasing p yields diminishing returns and may introduce unnecessary interventions.

Window length L. Fig. 3(c) studies the impact of the window length L used in Eq. (10), *i.e.*, the number of consecutive drift detections required to activate mitigation. Small L values tend to trigger overly early corrections, while overly large L delays intervention and reduces mitigation effectiveness. Intermediate L achieves the best trade-off, supporting the design choice of modeling *persistent* drift instead of reacting to instantaneous fluctuations.

Trigger size. Fig. 3(d) evaluates robustness to trigger size. Mitigation performance remains stable across different trigger magnitudes without catastrophic degradation, indicating that the strategy does not rely on a specific trigger scale and generalizes across attack configurations.

In summary, these results validate the necessity of combining temporal drift modeling with probabilistic action correction, and show that the key hyperparameters admit a reasonable operating range with consistent performance.

4.4 Discussion and Limitations

Our experimental results demonstrate behavioral drift as a trigger-agnostic signature for RL backdoor defense: it yields reliable separability across SARL and MARL and remains effective under interaction-induced non-stationary environments. BehaviorGuard's online guard achieves a favorable trade-off between utility and security, preserving clean performance while recovering poisoned performance without

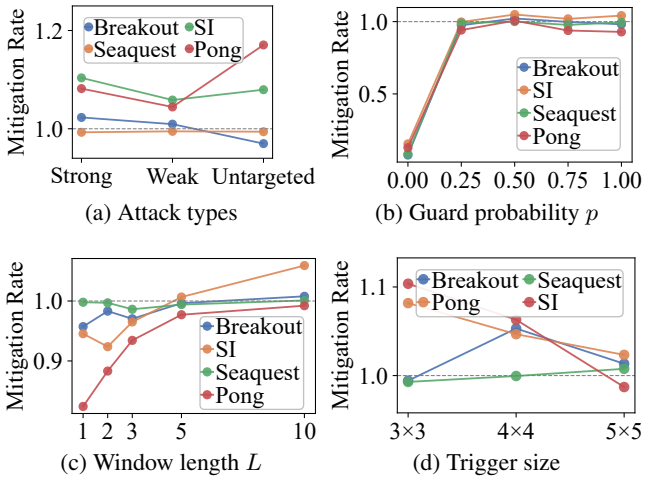


Figure 3: Ablation studies of BehaviorGuard. (a) Defense performance under different attack types, including strong targeted, weak targeted, and untargeted attacks. (b) Effect of p, which controls the strength of probabilistic action correction once persistent drift is detected. (c) Effect of L, determining how many consecutive drift detections are required to trigger mitigation. (d) Defense robustness with respect to trigger size. Results are averaged over 5 runs across Atari environments.

policy retraining or trigger reconstruction. The gains are especially clear for reward-constrained and sequential triggers where reward- or pattern-dependent defenses are weakened. BehaviorGuard still has a limitation. It is the reliance on benign baselines and a simple reference-action correction, which may require recalibration under distribution shift (e.g., new opponents or updated policies). Addressing this recalibration challenge and evaluating adaptive attackers that explicitly minimize drift are our future works.

5 Conclusion

We proposed a behavior-centric, online framework for backdoor detection and mitigation in DRL. Our key insight is that backdoored policies, despite differing trigger designs and objectives, tend to introduce persistent shifts in action distributions to ensure reliable activation, leaving detectable signatures in high-quantile regions and distribution tails even on trigger-free inputs. Based on this insight, we developed BehaviorGuard, which consists of a drift-based detector for backdoor activations without relying on reward anomalies, trigger priors, or trigger reconstruction, and a mitigator that suppresses backdoor actions at runtime and without requiring policy retraining. Extensive evaluations on SARL and MARL benchmarks demonstrate strong detection performance and a favorable utility-security trade-off: our defense substantially recovers poisoned performance while preserving clean-task performance, including under normal, reward-constrained, and sequential attacks. Future work will extend the drift constraint to continuous-action policies, improve calibration under distribution shift, and study adaptive attackers that explicitly optimize for drift evasion.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62202387), the Guangdong Basic and Applied Basic Research Foundation (2025A1515011112), and the supporting funds for talents of Nanjing University of Aeronautics and Astronautics.

References

[Acharya *et al.*, 2023] Manoj Acharya, Weichao Zhou, Anirban Roy, Xiao Lin, Wenchao Li, and Susmit Jha. Universal trojan signatures in reinforcement learning. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*, 2023.

[Alshiekh *et al.*, 2018] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, volume 32, 2018.

[Bansal *et al.*, 2017] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*, 2017.

[Bellemare *et al.*, 2013] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

[Bharti *et al.*, 2022] Shubham Bharti, Xuezhou Zhang, Adish Singla, and Jerry Zhu. Provable defense against backdoor policies in reinforcement learning. *NeurIPS*, 35:14704–14714, 2022.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[Busoniu *et al.*, 2008] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[Chen *et al.*, 2022] Yanjiao Chen, Zhicong Zheng, and Xueluan Gong. Marnet: Backdoor attacks against cooperative multi-agent reinforcement learning. *IEEE TDSC*, 20(5):4188–4198, 2022.

[Chen *et al.*, 2023] Xuan Chen, Wenbo Guo, Guanhong Tao, Xiangyu Zhang, and Dawn Song. Bird: generalizable backdoor detection and removal for deep reinforcement learning. *NeurIPS*, 36:40786–40798, 2023.

[Cui *et al.*, 2024] Jing Cui, Yufei Han, Yuzhe Ma, Jianbin Jiao, and Junge Zhang. Badrl: Sparse targeted backdoor attack against reinforcement learning. In *AAAI*, volume 38, pages 11687–11694, 2024.

[Fang *et al.*, 2025] Jing Fang, Saihao Yan, Xueyu Yin, Yinbo Yu, Chunwei Tian, and Jiajia Liu. Blast: A stealthy backdoor leverage attack against cooperative multi-agent deep reinforcement learning based systems. *arXiv preprint arXiv:2501.01593*, 2025.

[Foerster *et al.*, 2018] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI*, volume 32, 2018.

[Gao *et al.*, 2019] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *ACSAC*, pages 113–125, 2019.

[Gong *et al.*, 2024] Chen Gong, Zhou Yang, Yunpeng Bai, Junda He, Jieke Shi, Kecen Li, Arunesh Sinha, Bowen Xu, Xinwen Hou, David Lo, et al. Baffle: Hiding backdoors in offline reinforcement learning datasets. In *IEEE S&P*, pages 2086–2104, 2024.

[Gu *et al.*, 2017] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[Gu *et al.*, 2024] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories and applications. *IEEE TPAMI*, 2024.

[Guo *et al.*, 2023] Junfeng Guo, Ang Li, Lixu Wang, and Cong Liu. Policycleanse: Backdoor detection and mitigation for competitive reinforcement learning. In *ICCV*, pages 4699–4708, 2023.

[Kiourti *et al.*, 2019] Panagiota Kiourti, Kacper Wardega, Susmit Jha, and Wenchao Li. Trojdrl: Trojan attacks on deep reinforcement learning agents. *arXiv preprint arXiv:1903.06638*, 2019.

[Liu *et al.*, 2018] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*. Internet Soc, 2018.

[Liu *et al.*, 2025] Shijie Liu, Andrew C Cullen, Paul Montague, Sarah Erfani, and Benjamin IP Rubinstein. Fox in the henhouse: Supply-chain backdoor attacks against reinforcement learning. *arXiv preprint arXiv:2505.19532*, 2025.

[Lowe *et al.*, 2017] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *NeurIPS*, 30, 2017.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*, volume 80, pages 4292–4301, 2018.

[Rashid *et al.*, 2020] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

[Rathbun *et al.*, 2025] Ethan Rathbun, Alina Oprea, and Christopher Amato. Adversarial inception backdoor attacks against reinforcement learning. In *ICML*, 2025.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Tang *et al.*, 2025] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1):153–188, 2025.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Vyas *et al.*, 2024] Sanyam Vyas, Chris Hicks, and Vasilios Mavroudis. Mitigating deep reinforcement learning backdoors in the neural activation space. In *2024 IEEE Security and Privacy Workshops (SPW)*, pages 76–86. IEEE, 2024.

[Wang *et al.*, 2019] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, pages 707–723. IEEE, 2019.

[Wang *et al.*, 2021a] Lun Wang, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. Backdoorl: Backdoor attack against competitive reinforcement learning. *arXiv preprint arXiv:2105.00579*, 2021.

[Wang *et al.*, 2021b] Yue Wang, Esha Sarkar, Wenqing Li, Michail Maniatakos, and Saif Eddin Jabari. Stop-and-go: Exploring backdoor attacks on deep reinforcement learning-based traffic congestion control systems. *IEEE TIFS*, 16:4772–4787, 2021.

[Wang *et al.*, 2022] Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, and Shiqing Ma. Rethinking the reverse-engineering of trojan triggers. *NeurIPS*, 35:9738–9753, 2022.

[Yu *et al.*, 2022] Yinbo Yu, Jiajia Liu, Shouqing Li, Kepu Huang, and Xudong Feng. A temporal-pattern backdoor attack to deep reinforcement learning. In *IEEE GLOBECOM*, pages 2710–2715, 2022.

[Yu *et al.*, 2024a] Yinbo Yu, Jiajia Liu, Hongzhi Guo, Bomin Mao, and Nei Kato. A spatiotemporal backdoor attack against behavior-oriented decision makers in metaverse: From perspective of autonomous driving. *IEEE JSAC*, 42(4):948–962, 2024.

[Yu *et al.*, 2024b] Yinbo Yu, Saihao Yan, and Jiajia Liu. A spatiotemporal stealthy backdoor attack against cooperative multi-agent deep reinforcement learning. In *IEEE GLOBECOM*, pages 4280–4285, 2024.

[Yu *et al.*, 2026] Yinbo Yu, Xueyu Yin, Jiadai Wang, Chunwei Tian, Sai Xu, Qi Zhu, and Daoqiang Zhang. Behaviorguard: Online backdoor defense for deep reinforcement learning. *arXiv preprint arXiv:2605.05977*, 2026.

[Yuan *et al.*, 2024] Zhuowen Yuan, Wenbo Guo, Jinyuan Jia, Bo Li, and Dawn Song. Shine: Shielding backdoors in deep reinforcement learning. In *ICML*, 2024.

[Zheng *et al.*, 2023] Haibin Zheng, Xiaohao Li, Jinyin Chen, Jianfeng Dong, Yan Zhang, and Changting Lin. One4all: Manipulate one agent to poison the cooperative multi-agent reinforcement learning. *Computers & Security*, 124:103005, 2023.