# Supplementary Materials for PolicyCleanse: Backdoor Detection and Mitigation for Competitive Reinforcement Learning

Junfeng Guo[1] Ang Li[2] Lixu Wang[3] Cong Liu[4]

[1]University of Maryland, College Park [2]Simular Research [3]Northwestern University [4] UC Riverside

{junfeng.guo}@utdallas.edu {me@angli}.ai {lixuwang2025}@u.northwestern.edu

{congl}@ucr.edu

## 1. Broader Impact and Limitation

Our work aims to address the problem of backdoor detection and mitigation in reinforcement learning. We believe that our work provide deep reinforcement learning practitioners additional protection against backdoor attacks thus contributes positively to the human society and addresses a critical safety problem for reinforcement learning. Our method has certain limitations such as sensitive to the adaptive attack. However, as we mentioned in the paper, the adaptive attack would make the backdoor attack less effective and less stealthy, which is impractical in the real world.

## 2. Future Work.

Our work manages to solve application issues for competitive reinforcement learning from the security perspective (*e.g.*, Trojan Attack). There are still several potential issues for the real-world application of multi-agent reinforcement learning, such as its robustness [17, 18, 16, 9], fairness [20, 19], efficiency [4, 2], *etc.* We will focus on addressing potential issues for the real-world application of multi-agent reinforcement learning in the future work.

## 3. Detailed Description of Each Environment.

1. *Run to Goal:* Two agents are initialized on a flat place with two parallel finish lines. The agent that first reaches the finish line on its opposite side is determined as the winner. Two types of agents are experimented in this environment: ant agents and human agents.
2. *You Shall Not Pass:* A red agent and a blue agent are initialized face-to-face near a finish line. The blue agent aims to pass the finish line while the red tries to prevent it from passing the line. The blue agent wins if it passes the finish line; otherwise, the red wins.
3. *Sumo:* Two agents are set on a limited and circular area facing one another. The agent which touches the other and stands till the other falls becomes the winner. Consistent with [14], we only use human agents in this environment.

Each game is provided by OpenAI [1] and supported by Mujoco [12]. The reward for each agent is set according to the configurations of [1]. We illustrate each game in Fig. 1. The dimensions of observations and actions for each agent and environment is shown as below.

## 4. Calculation of Anomaly Index for $T$

Following previous work on backdoor detection [13, 5, 3, 15], we apply MAD outlier detection MAD(·) on $R_{\text{sum}}$ to determine whether (pseudo) trigger actions are found. Specifically, we firstly collect the negation of the target agent's accumulated reward against a dummy opponent agent within $M$ steps for 500 times as an array $R_{\text{arr}}$. Then we calculate the value of $T$ based on $R_{\text{arr}}$, where the $T$ can be just tagged as anomalous (*i.e.*, anomaly index $= 4$ [5]) against $R_{\text{arr}}$ using MAD outlier detectors. The anomaly index for a given $R_{\text{sum}}$ is computed as [8, 13]:

$$\text{Anomaly Index: } \frac{R_{\text{sum}} - \texttt{Median}[R_{\text{arr}}]}{C \cdot \texttt{Median}[|R_{\text{arr}} - \texttt{Median}[R_{\text{arr}}]|]} \tag{1}$$

,

where $C$ represents a constant value and is typically set as 1.4826 with the assumption that $R_{\text{sum}}$ fits Gaussian distribution [13, 5, 3]. Therefore, we set $T$ as:

$$T \leftarrow 4 \cdot C \cdot \texttt{Median}[|R_{\text{arr}} - \texttt{Median}[R_{\text{arr}}]|] + \texttt{Median}[R_{\text{arr}}] \tag{2}$$

Accordingly, for each reversed actions, if its corresponding $R_{\text{sum}} \geq T$, we determine the actions as trigger actions.

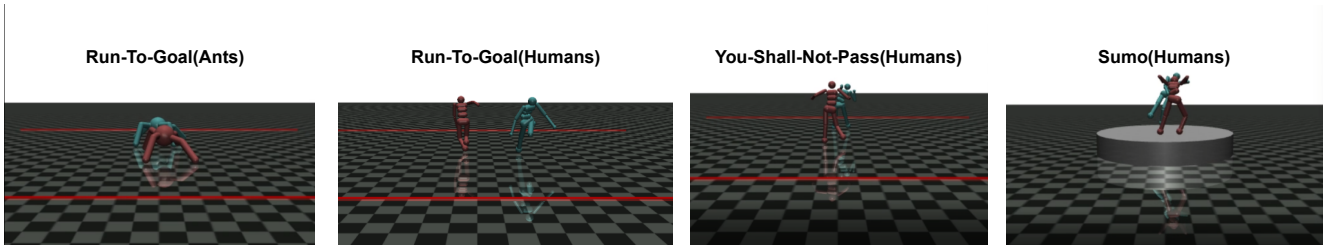


Figure 1. The illustration of Each Mujoco Game.

| Environment | Ants | Humans |
|---|---|---|
| | observations / actions | |
| Run-To-Goal | 122 / 8 | 380 / 17 |
| You-Shall-Not-Pass | ———- | 380 / 17 |
| Sumo | 122 / 8 | 395 / 17 |

Table 1. The dimensions of observation and action spaces for each agent and environment

## 5. The Details of Model Architectures and Implementation Configurations

Following previous work [14], the Trojan agent is built with Long Short-Term Memory (LSTM) architecture [7] to achieve both attack efficacy and stealth. The trigger length is set as 25 with 20% probability by default. The benign agents are built using multi-layer perceptions (MLP) or LSTM following previous work [1]. Consistent with previous work [1], we adopt two layers MLP with 128 neurons per hidden-layer for training benign agents for Run-To-Goal and You-Shall-Not-Pass. As for Sumo, we implement two layers LSTM with 128 neurons per hidden-layer. For trojan agents, we leverage a two-layer LSTM with 128 neurons per hidden-layer. We implement benign and `PolicyCleanse` using PPO [11] with stable baselines [10]. The default parameters for PPO is policy clip range $\epsilon = 0.2$, discounting factor $\gamma = 0.995$ and generalized advantage estimate parameter $\lambda = 0.95$. For each Trojan model, we inject $\geq 20\%$ poisonous trajectories to achieve the optimal attack efficacy. PolicyCleanse policy $\pi_S(s|\theta_S)$ is built with two-layer MLP and each layer has 64 neurons. PolicyCleanse has the same observation and input spaces as the Trojan agent.

## 6. The Performance of backdoor attacks for Sumo(Ants) and Sumo(Humans) Games

We conduct dozens of experiments for implementing BackdooRL for Sumo (Ants) task. However, we observe that the trojan agent mostly stay still against the opponent agent, as shown in Fig. 2, which leads to a very long game time and tie rate. We also issue this to the authors of BackdooRL. They attribute such observations to that the agent for Sumo (Ant) is rather stable thus both agents remain still during the game.

**Algorithm 1** PolicyCleanse

**Input:** Target Agent $\pi_T(\cdot)$; Environment with a random seed; number of steps in the Performing ($N$) and Observing ($M$) Phases.
**Initialize** PolicyCleanse policy $\pi_S(\cdot;\theta_S)$
**for** iteration=1,2,... **do**
  Run PolicyCleanse $\pi_S(\cdot;\theta_S)$ against $\pi_T(\cdot)$ in environment for $(M+N)$ timesteps for collecting trajectories$\{s_{\mathrm{S}}^t, a_{\mathrm{S}}^t, s_{\mathrm{T}}^t, a_{\mathrm{T}}^t, R_{\mathrm{T}}(s_{\mathrm{S}}^t, s_{\mathrm{T}}^t, a_{\mathrm{T}}^t)\}_{t=0}^{N+M}$;
  for $t \in [0, N)$ Calculate $R_S(s_{\mathrm{S}}^{(t)}, s_{\mathrm{T}}^{(t)}, a_{\mathrm{S}}^{(t)}) = -R_{\mathrm{T}}(s_{\mathrm{S}}^{(t)}, s_{\mathrm{T}}^{(t)}, a_{\mathrm{T}}^{(t)})$
  Calculate $R_{\text{sum}} = -\sum_{t=N}^{N+M} R_{\mathrm{T}}(s_{\mathrm{S}}^{(t)}, s_{\mathrm{T}}^{(t)}, a_{\mathrm{T}}^{(t)})$
  **if** $R_{\text{sum}}$ is deemed as anomalous based on Section.4.2 **then**
    $R_{\mathrm{S}}(t=N) = 10^3$
  **else**
    $R_{\mathrm{S}}(t=N) = -10^3$
  **end if**
  **Updating** PolicyCleanse $\pi_{\mathrm{S}}(\cdot;\theta_{\mathrm{S}})$ using PPO [11] through maximizing: $\sum_{t=0}^{N} \gamma^t R_{\mathrm{S}}(s_{\mathrm{S}}^{(t)}, s_{\mathrm{T}}^{(t)}, a_{\mathrm{S}}^{(t)})$
**end for**

Regarding Sumo (Humans), the Trojan agents would fail much possibly even though seeing actions different from benign actions sent by the trigger agent. For examples, the actions produced by initialized PolicyCleanse would also possibly trigger the Trojan agent. It is possibly caused by that the Trojan agent for Sumo (Humans) significantly overfits the benign actions from the opponent to preserve its performance when no trigger actions present.

## 7. Algorithm for PolicyCleanse

The algorithm for PolicyCleanse is detailed in Algorithm 1.

## 8. The Details of Mitigation Process

Our mitigation approach requires to interact with the environment for re-optimizing the Bellman equation, which is consistent with our threat model. The optiomization process should let the Trojan agent interact with the environment to search the $\hat{a}_T^{(n)}$ which can lead the optimal performance of Trojan agent under $\{\ \hat{s}_T^{(t=n)}, \ldots \hat{s}_T^{(t=\infty)}\}$,and replace the

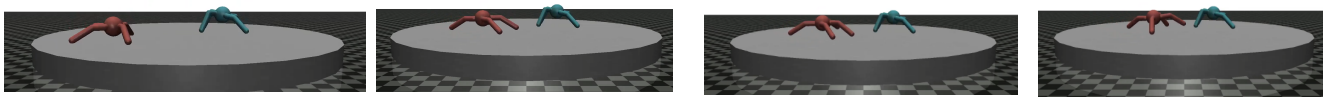

Figure 2. The illustration of backdoor attacks for Sumo(Ants).

$a_T^{(n)}$. For each $a_T^{(t=n)}$ to be replaced, the $\hat{s}_T^{(t=n)}$ is set to be $s_T^{(t=n)}$ and $\hat{s}_T^{(t=n+1,n+2,\ldots)}$ is given by the interation with environment during the optimization procedure. The $\hat{a}_T^{(t=n,n+1,n+2,\ldots)}$ is taken to optimize **??**, then we replace $a_T^{(t=n)}$ with $\hat{a}_T^{(t=n)}$

## 9. Computation Cost for `PolicyCleanse`

We still measure the computation cost for implementing `PolicyCleanse`. We test `PolicyCleanse` with NVIDIA 2080 Ti GPU and Intel Xeon E Processors. `PolicyCleanse` will take around 357 and 381 minutes for Ants and Humanoid agents with 2500 optimization iterations. In general, `PolicyCleanse` would detect at least one pseudo action within 1000 iterations, which would cost around 138 and 154 minutes for Ants and Humanoid agents. As for the mitigation, `PolicyCleanse` costs close to 2 hours in total for each agent with parallel computation. Compared with training a CRL agent from scratch [1, 6] that would take millions of iterations for optimization, we think the cost for `PolicyCleanse` is acceptable.

## 10. More results on comparison of accumulated reward for Trojan agent against PolicyCleanse and trigger agents

We here present the comparison of accumulated reward for different types of Trojan agent (*i.e.*, Dummy and Random agents) against PolicyCleanse and trigger agents, as shown in Fig. 3 and Fig. 4.

## 11. Additional visualization comparison for actual and reversed (pseudo) triggers

We here conduct additional visualization comparison for actual and reversed (pseudo) triggers with Humanoid agents since Humanoid's actions are more distinguishable than other agents (*i.e.*, ants). The results are shown in Fig. 5.

## 12. The pseudo trigger is not a consequence of the presence of natural Trojans

According to **??**, for benign agents, we observe there is no natural Trojan trigger according to our detection criteria. Specifically, no Trigger actions can be learned to cause the catastrophic failure for benign agents (green dotted line). Moreover, according to **??**, most reversed trigger actions stay close to the action trigger actions but stay away from the benign actions. Therefore, we do not think that the pseudo Trigger action is a consequence of the presence of natural Trojans.

## 13. Ablation study: The robustness against adaptive attacks.

To further investigate the robustness of `PolicyCleanse`, we evaluate `PolicyCleanse` under the worst scenario where the attacker is aware of our defense mechanism. We consider the attacker aims to bypass `PolicyCleanse` through making the activated Trojan agent's performance degrade slowly, thus performs stealthy against `PolicyCleanse`. Specifically, the attacker can manipulate the $\pi_{fail}$ in BackdooRL by performing Algorithm 2 instead of minimizing Eq.(3). We conduct experiments using an agent for each game and its architecture and trigger actions are consistent with Section.5.1. We test adaptive BackdooRL across four games and find that the adaptive backdooRL can successfully learn a Trojan agent $\pi_T$ being able to bypass the detection of PolicyCleanse. The entire learning procedure of adaptive BackdooRL typically takes 10-13 iterations across each game.

However, we also conduct experiments to verify the efficacy of the proposed adaptive BackdooRL, the results is shown in Fig. 6. We find that even though the adaptive attack can bypass PolicyCleanse, its attack efficacy significantly decreases ($\geq 19.6\%$) comparing with BackdooRL across four games, which means that `PolicyCleanse` can significantly alleviate the attack efficacy for current Trojan attack(*i.e.*,BackdooRL) across four games. And for Run-To-Goal(Ants), You-Shall-Not-Pass and Sumo(Human) these games, the adaptive BackdooRL's efficacy degrades close to the benign agents(The trigger agent don't send triggers). Such results may be attributed to two reasons: First, the Trojan policy $\pi_{fail}$ learned by the Trojan agent through our considered adaptive attack performs less effective compared with that for BackdooRL, which can be revealed by its accumulated reward is larger according to Line.6 in Algorithm 2. Secondly, adaptive BackdooRL would make the Trojan policy $\pi_{fail}$ fail slower compared with BackdooRL. As discussed in BackdooRL [14], the neural network for Trojan agents(*e.g.*, LSTM [7]) has limited memory to remember the Trojan policy, therefore BackdooRL proposes to make $\pi_{fail}$ fail as quickly as possible. However, the adap-

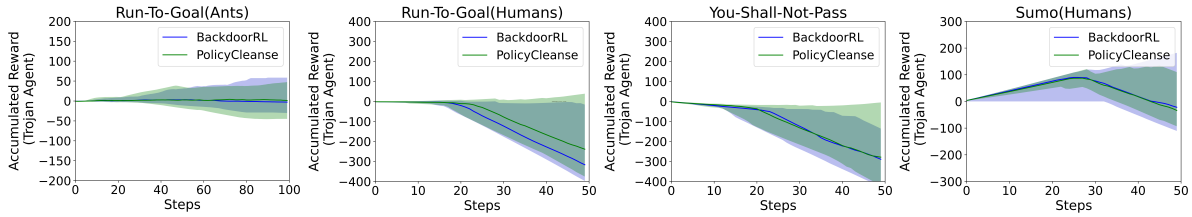


Figure 3. The comparison of accumulated reward for Trojan agent (Dummy Agent) against `PolicyCleanse` and trigger agents.

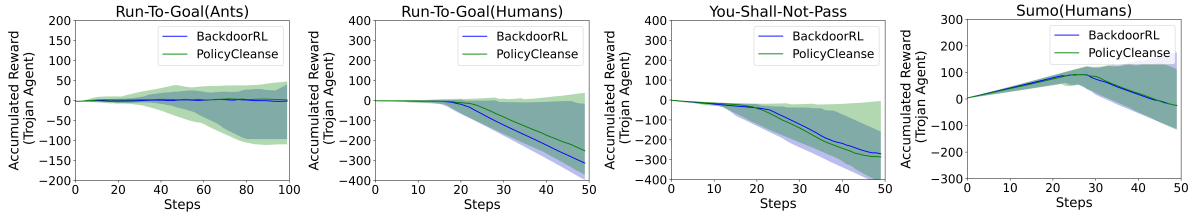


Figure 4. The comparison of accumulated reward for Trojan agent (Random Agent) against `PolicyCleanse` and trigger agents.

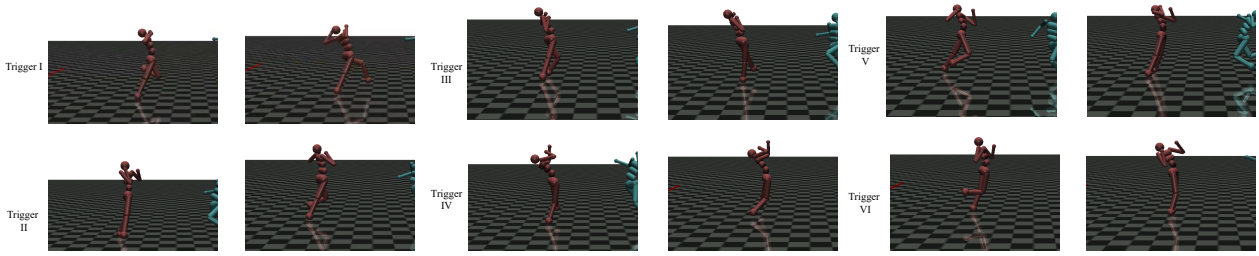


Figure 5. Additional visualization comparison for actual and reversed (pseudo) triggers. The left figure within each row is the actual trigger while the right one represents the corresponding reversed (pseudo) trigger. Triggers I and II are evaluated under Run-To-Goal(Humans), Trigger III and IV are evaluated under You-Shall-Not-Pass; Trigger V and VI are evaluated under Sumo(Humans).

**Algorithm 2** Adaptive BackdooRL

**Input:** PolicyCleanse algorithm with default parameters; Benign policy $\pi_{win}(s)$
**Output:** Adaptive BackdooRL $\pi_T(s)$
**Initialize** the minimum and maximum accumulated value for Trojan policy as $R_{min} = -1000$ and $R_{max} = 1000$
**while** $R_{max} - R_{min} < 1$: **do**
    Initialize Trojan policy: $\pi_{fail}(s) \leftarrow \pi_{win}(s)$
    Use PPO [11] to learn $\pi_{fail}(s)$ by minimizing Eq.(3) until $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t (R(s^{(t)}, a_T^{(t)}; \pi_{fail}))] \leq \frac{R_{max}+R_{min}}{2}$ against a dummy agent; ▷ When the $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t (R(s^{(t)}, a_T^{(t)}; \pi_{fail}))]$ drops close to the threshold, we stop optimizing.
    Use BackdooRL to learn $\pi_T(s)$ based on $\pi_{fail}$ and $\pi_{win}$;
    Implement PolicyCleanse on $\pi_T(s)$ as default;
    **if** PolicyCleanse finds a pseudo trigger action within 5,000 iterations **then**
        $R_{min} \leftarrow \frac{R_{max}+R_{min}}{2}$
    **else**
        $R_{max} \leftarrow \frac{R_{max}+R_{min}}{2}$
    **end if**
**end while**

tive BackdooRL may make the entire Trojan agent hard to imitate the Trojan policy.

Last but not least, we also summarize the failing speed for the adaptive BackdooRL and BackdooRL in Fig. 7. We can see that adaptive BackdooRL would require the Trojan agent to take significantly more steps for failing compared with BackdooRL. In the real world, failing speed would also affect the stealth of Trojan attacks against reinforcement

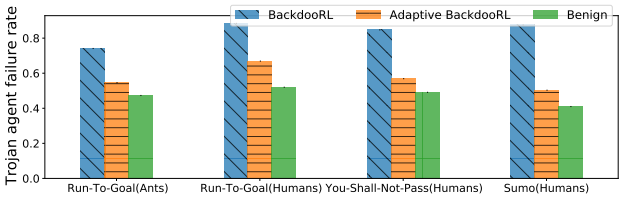


Figure 6. The comparison on the attack efficacy of BacdooRL and our proposed adaptive attack. The failure rate is reported over 500 games. The benign means that the performance of a benign agent. The value is reported as the median value.

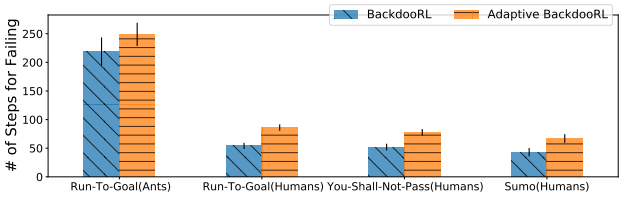


Figure 7. The comparison on the failing speed of BacdooRL and our proposed adaptive attack. The failing speed is measured by the number of steps for failing and is reported over 500 games. The failing speed for the adaptive BackdooRL is from $14\%$ to $59\%$ slower than BackdooRL across four games.

learning. This is because if a Trojan agent takes more steps to fail, it would be more likely to be observed and taken controlled by the human beings. From this perspective, adaptive BackdooRL would behave less stealth in the real world application compared with BackdooRL.

Considering the attack efficacy and failing speed for the adaptive BackdooRL, we think the attacker may not have much incentive to conduct the existing Trojan attack(*i.e.*, BackdooRL) and its variants against PolicyCleanse. However, there may be other advanced Trojan attacks to bypass PolicyCleanse while preserving the attack efficacy, which can be explored in the future.

## References


[1] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*, 2017. 1, 2, 3

[2] Jiahua Dong, Duzhen Zhang, Yang Cong, Wei Cong, Henghui Ding, and Dengxin Dai. Federated incremental semantic segmentation. In *CVPR*, pages 3934–3943, June 2023. 1

[3] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. *arXiv preprint arXiv:2103.13127*, 2021. 1

[4] Shangqian Gao, Burak Uzkent, Yilin Shen, Heng Huang, and Hongxia Jin. Learning to jointly share and prune weights for grounding based vision and language models. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[5] Junfeng Guo, Ang Li, and Cong Liu. AEVA: Black-box backdoor detection using adversarial extreme value analysis. In *International Conference on Learning Representations*, 2022. 1

[6] Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing. Adversarial policy learning in two-player competitive games. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3910–3919. PMLR, 18–24 Jul 2021. 3

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 3

[8] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology*, 49(4):764–766, 2013. 1

[9] Shahab Nikkhoo, Zexin Li, Aritra Samanta, Yufei Li, and Cong Liu. Pimbot: Policy and incentive manipulation for multi-robot reinforcement learning in social dilemmas, 2023. 1

[10] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. *GitHub repository*, 2019. 2

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2, 4

[12] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 1

[13] 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 *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 1

[14] 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. 1, 2, 3

[15] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 222–238. Springer, 2020. 1

[16] Yihan Wu, Aleksandar Bojchevski, and Heng Huang. Adversarial weight perturbation improves generalization in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10417–10425, 2023. 1

[17] Yihan Wu, Aleksandar Bojchevski, Aleksei Kuvshinov, and Stephan Günnemann. Completing the picture: Randomized smoothing suffers from the curse of dimensionality for a large family of distributions. In *International Conference on Artificial Intelligence and Statistics*, pages 3763–3771. PMLR, 2021. 1

[18] Yihan Wu, Hongyang Zhang, and Heng Huang. Retrievalguard: Provably robust 1-nearest neighbor image retrieval. In *International Conference on Machine Learning*, pages 24266–24279. PMLR, 2022. 1

[19] Yanfu Zhang, Runxue Bao, Jian Pei, and Heng Huang. Toward unified data and algorithm fairness via adversarial data augmentation and adaptive model fine-tuning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1317–1322. IEEE, 2022. 1

[20] Yanfu Zhang, Shangqian Gao, and Heng Huang. Recover fair deep classification models via altering pre-trained structure. In *European Conference on Computer Vision*, pages 481–498. Springer, 2022. 1