

Boosting Transferable Adversarial Attacks against Deep Reinforcement Learning

Zexin Li¹ Ruili Yao² Yiming Zeng³ Xiaoxue Gao^{4*}

¹Nanyang Technological University

²University of California, Riverside ³University of Connecticut

⁴School of Artificial Intelligence,

The Chinese University of Hong Kong, Shenzhen

zexin.li@ntu.edu.sg

Abstract

Most adversarial attacks on deep reinforcement learning (DRL) assume white-box access to the victim policy, which rarely holds in practice. This paper studies *transfer-based black-box* attacks on DRL: the attacker crafts observation perturbations on a white-box surrogate agent and feeds them to an unknown victim. We formulate the attack as return minimization under a per-step perturbation budget. We first show that transplanting transferable image-classification attacks (FGSM, MI-FGSM, and NI-FGSM) with a per-step objective yields perturbations that transfer but are no stronger than random noise of the same budget. We then propose a *trajectory-level* attack that optimizes a sequence of perturbations over a receding horizon through a differentiable model of the environment and a temperature-smoothed surrogate policy, with the same optimizers. On CartPole-v1 with ten DQN and DDQN agents and 100 surrogate-victim pairs, the trajectory-level attack outperforms per-step attacks and random noise in the white-box, cross-model, and cross-algorithm settings.

1 Introduction

Deep reinforcement learning (DRL) has achieved remarkable success in learning complex control tasks and adapting to dynamic environments [12, 26, 29]. However, DRL policies inherit the adversarial fragility of deep neural networks: small, carefully crafted perturbations of the observations can steer an agent into catastrophic actions [11, 24, 34]. Such vulnerabilities have been demonstrated across modalities and tasks, including face recognition [20], speech recognition [8], dialogue generation [19], the efficiency of dynamic neural networks and large language models [4, 5], backdoors in vision-language models [6], and policy manipulation in multi-robot reinforcement learning [21, 27].

Most attacks on DRL assume white-box access to the victim’s parameters [11, 24, 30, 34]. In practice, a deployed

policy is usually inaccessible, so the attacker has to operate in a *black-box* setting [11, 28]. A practical strategy is to craft *transferable* adversarial examples on a white-box surrogate model and apply them to the victim. This strategy has proven highly effective for image classification [7, 20, 23] and speech recognition [8], and has also been demonstrated in DRL across policies and algorithms [11, 28]. In particular, prior work identifies shared adversarial features across MDPs [13] and shows that adversarial training can leave policies vulnerable to natural, black-box perturbations [14]. Building on this evidence, we study whether explicitly optimizing future state trajectories improves transfer over per-step attack objectives.

A natural first step is to apply strong transferable attacks from image classification independently at every time step. We find that, on CartPole, such perturbations transfer across seeds and algorithms but are no more damaging than random noise at the corners of the same budget. The per-step objective is myopic: it only targets the current action, whereas a well-trained policy quickly corrects a single wrong action.

We therefore lift the attack objective from single actions to trajectories. Our attack optimizes a sequence of perturbations over a receding horizon so that the surrogate agent, rolled out through a differentiable environment model, is driven toward failure. We solve this problem with the momentum and Nesterov optimizers of MI-FGSM and NI-FGSM, and we smooth the surrogate policy with a temperature to avoid overfitting to its exact decision boundary.

Our contributions are: (1) a return-minimization formulation of transfer attacks on DRL that explains why per-step relaxations fall short (§3.1); (2) a trajectory-level attack that unifies FGSM, MI-FGSM, and NI-FGSM under a receding-horizon objective (§3.2); and (3) an evaluation of white-box, cross-model, and cross-algorithm transfer over ten DQN/DDQN agents and 100 surrogate-victim pairs (§4).

2 Related Work

Transferable adversarial attacks. Transferability can be improved with iterative attacks [18, 25], momentum [7], Nes-

*Assistant Professor, School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen.

terov look-ahead [23], variance tuning [32], and input diversity [33]. They have also been tailored to face recognition [20] and speech recognition [8]. Beyond accuracy, adversarial inputs can degrade model efficiency [4, 5], manipulate text generation [19], or implant stealthy backdoors [6, 10].

Adversarial attacks on DRL. Huang et al. [11] first showed that FGSM [9] perturbations degrade DRL policies in Atari games [2] and briefly examined transfer across policies and algorithms. Lin et al. [24] proposed the strategically timed attack and the enchanting attack, where the latter uses a learned dynamics model to plan a sequence of target actions. Optimal white-box adversaries can be learned with reinforcement learning [30, 34], and domain randomization improves transfer [28]. In multi-agent settings, PIMbot manipulates the policies and incentives of robot teams [21, 27]. Korkmaz [13] shows that sensitive directions can transfer across states and MDPs, including to adversarially trained policies. Korkmaz [14] further demonstrates that natural observation transformations can undermine nominally robust policies in a black-box setting. Our focus is on optimizing temporally coordinated perturbations through a differentiable dynamics model and evaluating their transfer across seeds and algorithms.

Diagnosing and evaluating robustness. Korkmaz and Brown-Cohen [17] detect adversarial directions using local quadratic approximations of the policy loss. Korkmaz [15] analyzes how unstable directions evolve across states and time, revealing limitations of robust training. Korkmaz [16] shows that enforcing local invariance can impair counterfactual value learning. These findings motivate evaluating defenses beyond the particular perturbations used during training; our experiments examine attack effectiveness rather than establishing a robustness guarantee.

Sequence-level objectives. In reinforcement learning for language-model reasoning, SPEAR [22] uses order-aware alignment of symbolic reasoning trajectories to provide process rewards for distillation. The connection to our work is the use of sequence structure in the objective; SPEAR studies reasoning distillation, whereas we optimize observation perturbations for adversarial control.

3 Method

3.1 Problem Formulation

We model the task as a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$. The victim is a value-based agent with Q-network $Q_v(\cdot, \cdot; \theta_v)$ and greedy policy $\pi_v(s) = \arg \max_a Q_v(s, a; \theta_v)$. At each step t , the attacker observes the true state s_t and injects a perturbation δ_t into the victim’s observation (Figure 1), so that the victim acts on $\tilde{s}_t = s_t + \delta_t$ while the environment evolves according to the true state. The attacker aims to minimize the victim’s expected return under

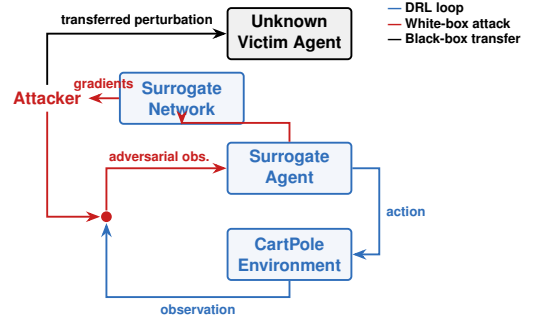


Figure 1: Threat model. In the white-box setting (red), the attacker computes gradients through a known surrogate agent to perturb the observations it receives from the environment. In the transfer-based black-box setting (black), the same perturbations are applied to an unknown victim agent whose parameters are never accessed.

a per-step ℓ_∞ budget ϵ :

$$\begin{aligned} \min_{\{\delta_t\}_{t=0}^H} J(\delta) &= \mathbb{E} \left[\sum_{t=0}^H \gamma^t r(s_t, a_t) \right] \\ \text{s.t. } a_t &= \pi_v(s_t + \delta_t), \quad \|\delta_t\|_\infty \leq \epsilon, \\ s_{t+1} &\sim P(\cdot \mid s_t, a_t), \end{aligned} \quad (1)$$

where H is the episode horizon. In the black-box setting, θ_v is unknown and the attacker cannot query π_v while crafting perturbations. It trains a surrogate Q-network $Q_s(\cdot, \cdot; \theta_s)$ in the same environment, possibly with a different seed or algorithm, and uses it as a stand-in for Q_v . Even with a surrogate, Eq. (1) is hard to optimize: the return is a non-differentiable function of a long sequence of discrete actions and stochastic transitions.

Per-step relaxation. The standard relaxation, which we use as a baseline, optimizes each perturbation independently. We choose a lure action $a^\dagger$ whose persistent execution drives the system toward failure (for CartPole, always pushing the cart in the same direction), and at each step we solve

$$\delta_t^* = \arg \max_{\|\delta\|_\infty \leq \epsilon} \log \text{softmax} \left(Q_s(s_t + \delta, \cdot) \right)_{a^\dagger} \quad (2)$$

with FGSM [9], MI-FGSM [7], or NI-FGSM [23]. This objective is myopic: one wrong action changes the state only slightly and the victim’s next decision can undo it, so a per-step attack must succeed at almost every step, which is hard under a small budget and harder after transfer.

3.2 Trajectory-Level Transfer Attack

We instead optimize perturbations with respect to their effect on the future trajectory. We assume the attacker has a differentiable model f of the environment dynamics, $\hat{s}_{k+1} = f(\hat{s}_k, u_k)$, where u_k is a continuous relaxation of the action. For CartPole, f is the known physical model; in general, it can be learned from interaction data, as in the enchanting attack [24]. To make the rollout differentiable, we replace the

surrogate’s greedy action with a temperature-smoothed policy

$$\tilde{\pi}_s(a \mid x) = \text{softmax}(Q_s(x, \cdot)/\kappa)_a, \quad (3)$$

and apply the expected action $u_k = \mathbb{E}_{a \sim \tilde{\pi}_s(\cdot \mid \hat{s}_k + \delta_k)}[a]$ (for CartPole, the expected force). At step t , the attacker optimizes a sequence of K perturbations $\Delta_t = (\delta_t, \dots, \delta_{t+K-1})$ to maximize the proximity of the predicted states to the failure region:

$$\max_{\|\delta_k\|_\infty \leq \epsilon} \mathcal{L}_{\text{traj}}(\Delta_t) = \sum_{k=1}^K \phi(\hat{s}_{t+k}), \quad \hat{s}_t = s_t, \quad (4)$$

where $\phi(s) = \max\{(\theta/\theta_{\max})^2, (x/x_{\max})^2\}$ measures how close the pole angle θ and the cart position x are to their termination thresholds. Following model-predictive control, only δ_t is applied, and the remaining perturbations warm-start the optimization at the next step.

We solve Eq. (4) with the same operators as in the per-step case, applied to the whole sequence Δ_t . With step size $\alpha = \epsilon/T$ and momentum decay μ , MI-FGSM accumulates an ℓ_1 -normalized gradient momentum,

$$g_{j+1} = \mu g_j + \frac{\nabla_{\Delta} \mathcal{L}_{\text{traj}}(\hat{\Delta}_j)}{\|\nabla_{\Delta} \mathcal{L}_{\text{traj}}(\hat{\Delta}_j)\|_1}, \quad (5)$$

with $\hat{\Delta}_j = \Delta_j$ for MI-FGSM and the Nesterov look-ahead $\hat{\Delta}_j = \Delta_j + \alpha \mu g_j$ for NI-FGSM, followed by

$$\Delta_{j+1} = \text{Clip}_{\epsilon}(\Delta_j + \alpha \cdot \text{sign}(g_{j+1})). \quad (6)$$

FGSM takes a single step of size ϵ . Algorithm 1 summarizes the full attack.

Why the trajectory objective transfers. First, Eq. (4) depends on the environment dynamics, which the surrogate and the victim share, rather than on the surrogate’s exact action at each step, so perturbations that push the system toward failure over several steps help against any policy that reacts similarly. Second, a larger temperature κ smooths the surrogate’s decision boundary, which differs between agents, and yields gradients that follow the overall trend of the Q-function, similar to the gradient stabilization of momentum [7].

4 Experiments

4.1 Experimental Setup

Environment and agents. We use CartPole-v1 from Gymnasium [1, 3]. The observation is a raw 4-D vector (cart position, cart velocity, pole angle, and pole angular velocity), there are two actions (push left or right), and the agent receives a reward of +1 per step, up to a maximum return of 500. We train DQN [26] and Double DQN (DDQN) [31] agents with five seeds each, which gives ten models. All agents use a two-layer MLP (128 units), Adam (learning rate 5×10^{-4}), batch size 128, soft target updates, and 600 training episodes, and we keep the best checkpoint under periodic greedy evaluation. Figure 2 shows the training curves. Without attack, eight

Algorithm 1 Trajectory-Level Transfer Attack

Require: Surrogate Q-network Q_s , black-box victim policy π_v , dynamics model f , budget ϵ , horizon K , iterations T , momentum decay μ , temperature κ , operator $\mathcal{O} \in \{\text{FGSM}, \text{MI}, \text{NI}\}$

Ensure: Perturbation sequence that minimizes the victim’s return J in Eq. (1)

```

1: Observe  $s_0$ ; initialize  $\Delta \leftarrow \mathbf{0}_K$ 
2: for  $t = 0, 1, \dots, H$  do
3:    $\alpha \leftarrow \epsilon/T$ ,  $\Delta_0 \leftarrow \Delta$ ,  $g_0 \leftarrow \mathbf{0}$ 
4:   for  $j = 0, \dots, T-1$  do
5:      $\hat{\Delta}_j \leftarrow \Delta_j + \alpha \mu g_j$  (NI) or  $\Delta_j$  (MI)
6:     Roll out  $f$  from  $s_t$  with  $\tilde{\pi}_s$  (Eq. (3)) under  $\hat{\Delta}_j$ 
7:     Compute  $\mathcal{L}_{\text{traj}}$  via Eq. (4) and backpropagate
8:     Update  $g_{j+1}$  via Eq. (5) and  $\Delta_{j+1}$  via Eq. (6)
9:   end for ▷ FGSM: one step of size  $\epsilon$ 
10:  Victim acts  $a_t \leftarrow \pi_v(s_t + \delta_t)$ ,  $\delta_t = \Delta_T[0]$  ▷ Transfer
11:  Observe  $r_t, s_{t+1}$ ; break if terminal
12:   $\Delta \leftarrow \Delta_T$  shifted by one step ▷ Warm start
13: end for

```

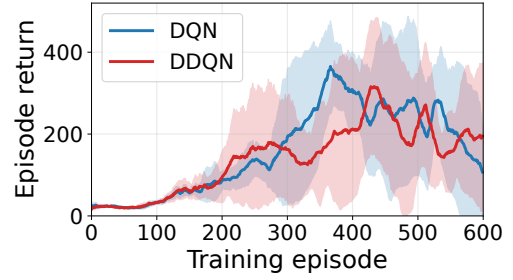


Figure 2: Training curves of the DQN and DDQN agents on CartPole. Each line is the mean episode return over five random seeds, smoothed with a 20-episode moving average, and the shaded band shows one standard deviation across seeds. The best checkpoint of each agent is selected by periodic greedy evaluation and then used as a surrogate or victim.

agents obtain the maximum return of 500, and the other two average 489.3 and 437.8.

Attacks. We compare the per-step FGSM, MI-FGSM, and NI-FGSM attacks (Eq. (2), lure action $a^\dagger = \text{push left}$) with their trajectory-level counterparts Traj-FGSM, Traj-MI-FGSM, and Traj-NI-FGSM (Algorithm 1). All iterative attacks use $T = 10$ iterations and $\mu = 1.0$. The trajectory attacks use horizon $K = 20$ and temperature $\kappa = 2$ and roll out the CartPole equations of motion. We also report two random baselines with the same budget: *Uniform* noise drawn from $[-\epsilon, \epsilon]^4$, and *Sign* noise that places every coordinate at a random corner, $\pm\epsilon$. Budgets are given in raw observation units, and we report $\epsilon \in \{0.05, 0.1\}$.

Protocol. We evaluate three settings: *white-box* (the surrogate is the victim), *cross-model* (same algorithm, different seed; 20 ordered pairs per algorithm), and *cross-algorithm* (DQN↔DDQN; 25 pairs per direction). Each evaluation runs 10 episodes with fixed environment seeds, and we report the victim’s average return (lower is stronger).

Table 1: Attack results on CartPole in the white-box, cross-model, and cross-algorithm settings. Each entry is the average episode return of the victim (lower means a stronger attack), reported as mean and standard deviation over the five victims (white-box), the twenty ordered surrogate–victim pairs of the same algorithm with different seeds (cross-model), or the twenty-five pairs in which the surrogate uses the other algorithm (cross-algorithm). The random baselines use uniform noise and random corner noise with the same budget and are evaluated directly on the victims. The middle attack group optimizes each step independently, and the bottom group optimizes perturbation sequences over a planning horizon. Bold marks the strongest attack in each column.

Budget	White-box				Cross-model transfer				Cross-algorithm transfer			
	0.05		0.1		0.05		0.1		0.05		0.1	
	DQN	DDQN	DQN	DDQN	DQN	DDQN	DQN	DDQN	DDQN	DQN	DDQN	DQN
No attack	500.0±0	485.4±24	500.0±0	485.4±24	500.0±0	485.4±24	500.0±0	485.4±24	485.4±24	500.0±0	485.4±24	500.0±0
Uniform noise	375.0±130	304.7±193	172.7±53	247.0±193	375.0±130	304.7±193	172.7±53	247.0±193	304.7±193	375.0±130	247.0±193	172.7±53
Sign noise	240.5±92	284.9±204	74.3±38	121.2±133	240.5±92	284.9±204	74.3±38	121.2±133	284.9±204	240.5±92	121.2±133	74.3±38
FGSM	322.0±140	336.0±179	262.5±177	214.7±163	330.9±138	314.7±204	173.4±128	274.6±185	302.1±212	411.2±124	163.4±138	364.5±140
MI-FGSM	315.8±157	277.0±200	159.5±174	125.4±188	338.9±140	317.0±209	147.7±120	211.2±190	317.7±214	400.9±127	148.6±160	324.0±166
NI-FGSM	304.3±163	286.3±201	153.5±176	125.4±189	360.5±138	306.7±206	158.2±121	215.6±189	319.2±204	400.2±130	169.7±184	322.3±164
Traj-FGSM	113.0±108	243.9±211	56.5±52	30.6±23	190.0 ±147	240.0 ±205	77.3±54	43.3±30	259.1 ±214	181.8 ±126	41.6 ±27	82.3±74
Traj-MI-FGSM	73.5 ±49	157.3 ±178	19.5±13	15.7±6	203.4±154	268.1±217	60.6±95	41.1 ±40	266.9±216	186.3±162	47.5±47	53.1 ±57
Traj-NI-FGSM	74.1±44	173.1±177	18.5 ±12	13.9 ±4	202.9±149	271.6±219	59.6 ±78	45.5±40	268.9±214	189.3±154	55.1±50	58.7±62

4.2 Main Results

Per-step attacks transfer but are weak. Iterative per-step attacks are clearly stronger than FGSM: in the white-box setting at a budget of 0.1, MI-FGSM and NI-FGSM reduce the DQN return from 500 to 159.5 and 153.5, whereas FGSM only reaches 262.5 (Table 1, white-box). They transfer across seeds almost without loss (147.7 from one DQN to another) but are consistently weaker than Sign noise (74.3). Transfer across algorithms is also asymmetric: per-step attacks crafted on DDQN barely affect DQN victims, whose return stays above 320 (Table 1, cross-algorithm).

Trajectory-level attacks are much stronger. In the white-box setting at a budget of 0.1, Traj-NI-FGSM reduces the return to 18.5 on DQN and 13.9 on DDQN, 4–9× lower than Sign noise. At the smaller budget of 0.05, where all per-step attacks leave the victims above a return of 277, Traj-MI-FGSM reaches 73.5 on DQN and 157.3 on DDQN.

Transfer across seeds and algorithms. For cross-model transfer at a budget of 0.1, the trajectory attacks reduce DQN and DDQN victims to 59.6 and 41.1, compared with 147.7 and 211.2 for the best per-step attack and 74.3 and 121.2 for Sign noise. They also remove the asymmetry of cross-algorithm transfer: perturbations crafted on DDQN reduce DQN victims to 53.1, compared with more than 320 for per-step attacks. This supports our hypothesis that an objective defined through the shared dynamics transfers better than one defined through individual actions. At a budget of 0.05, they remain the strongest in every column, with a smaller margin (e.g., 190.0 compared with 240.5 for Sign noise).

Momentum and look-ahead. Momentum and look-ahead help most in the white-box setting (56.5 for Traj-FGSM compared with about 19 on DQN at a budget of 0.1). Under transfer, the three trajectory operators perform similarly, so the trajectory objective itself is the main source of transferability.

5 Conclusion

Per-step transfer attacks from image classification transfer across DRL agents but are no stronger than random corner noise. Optimizing perturbation sequences over a receding horizon through a differentiable environment model makes the same optimizers far more effective in white-box, cross-model, and cross-algorithm settings, which calls for defenses against temporally coordinated perturbations. Extending the attack to high-dimensional observations and learned dynamics models is left for future work.

References

- [1] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077.
- [2] 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.
- [3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [4] Yiming Chen, Simin Chen, Zexin Li, Wei Yang, Cong Liu, Robby T. Tan, and Haizhou Li. Dynamic transformers provide a false sense of efficiency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7164–7180, 2023.
- [5] Yiming Chen, Zexin Li, Xianghu Yue, Robby T. Tan, and Haizhou Li. NaturalSloth: Revisiting denial-of-service attacks on large language models. In *Proceedings of the Annual*

- [6] Yezheng Cheng, Zexin Li, Jiaqi Wu, Zhihong Zhang, and Simin Chen. VLMSysTrojan: Stealthy system-aware backdoor attacks against vision-language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.
- [7] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9185–9193, 2018.
- [8] Xiaoxue Gao, Zexin Li, Yiming Chen, Cong Liu, and Haizhou Li. Transferable adversarial attacks against ASR. *IEEE Signal Processing Letters*, 2024. doi: 10.1109/LSP.2024.3443711.
- [9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. URL <http://arxiv.org/abs/1412.6572>.
- [10] Junfeng Guo, Ang Li, and Cong Liu. Backdoor detection and mitigation in competitive reinforcement learning. *arXiv preprint arXiv:2202.03609*, 2022.
- [11] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.
- [12] Gregory Kahn, Adam Villafior, Bosen Ding, Pieter Abbeel, and Sergey Levine. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, 2018.
- [13] Ezgi Korkmaz. Deep reinforcement learning policies learn shared adversarial features across MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7229–7238, 2022. doi: 10.1609/aaai.v36i7.20684.
- [14] Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8369–8377, 2023. doi: 10.1609/aaai.v37i7.26009.
- [15] Ezgi Korkmaz. Understanding and diagnosing deep reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 25265–25275. PMLR, 2024. URL <https://proceedings.mlr.press/v235/korkmaz24a.html>.
- [16] Ezgi Korkmaz. How to lose inherent counterfactuality in reinforcement learning. In *International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=2kutK2Y8Sv>.
- [17] Ezgi Korkmaz and Jonah Brown-Cohen. Detecting adversarial directions in deep reinforcement learning to make robust decisions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 17534–17543. PMLR, 2023. URL <https://proceedings.mlr.press/v202/korkmaz23a.html>.
- [18] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.
- [19] Yufei Li, Zexin Li, Yingfan Gao, and Cong Liu. White-box multi-objective adversarial attack on dialogue generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1778–1792, 2023.
- [20] Zexin Li, Bangjie Yin, Taiping Yao, Junfeng Guo, Shouhong Ding, Simin Chen, and Cong Liu. Sibling-attack: Rethinking transferable adversarial attacks against face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24626–24637, 2023.
- [21] Zexin Li, Ziliang Zhang, Hyoseung Kim, and Cong Liu. PIMbot: A self-adaptive attack framework for adversarial manipulation of multi-robot reinforcement learning. *arXiv preprint arXiv:2605.23027*, 2026.
- [22] Zhuochun Li, Yuelyu Ji, Yiming Zeng, and Daqing He. SPEAR: Distilling domain-adaptive reasoning skeletons via sequential symbolic alignment in reinforcement learning. *arXiv preprint arXiv:2608.26550*, 2026. doi: 10.48550/arXiv.2608.26550.
- [23] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=SJlHwkBYDH>.
- [24] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3756–3762, 2017.
- [25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations (ICLR)*, 2018.
- [26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fiedjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [27] Shahab Nikkhoo, Zexin Li, Aritra Samanta, Yufei Li, and Cong Liu. PIMbot: Policy and incentive manipulation for multi-robot reinforcement learning in social dilemmas. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5630–5636, 2023.
- [28] Xinlei Pan, Yulong Cao, Xindi Wu, Eric Zelikman, Chaowei Xiao, Yanan Sui, Rudrasis Chakraborty, and Ronald S Fearing. Transferable adversarial attacks on deep reinforcement learning with domain randomization. In *CVPR Workshops*, 2020. URL https://adv-workshop-2020.github.io/short_papers/79.pdf.
- [29] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):143, 2018.

- [30] Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep RL. In *International Conference on Learning Representations (ICLR)*, 2022.
- [31] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.
- [32] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1933, 2021.
- [33] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2730–2739, 2019.
- [34] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 21024–21037, 2020.