Adversarial Attacks on Multi-Agent Reinforcement Learning

Preliminary Exam

Young Wu

November 17, 2022

Thesis Statement

- There are vulnerabilities in multi-agent systems and attackers can influence the behavior of multi-agent reinforcement learners through data or environment poisoning.

List of Projects

- Completed:

my1 Game Redesign: Training time, reward poisoning, online victims.

my2 Nash Attack: Training time, reward poisoning, two offline victims.

my3 DSE Attack: Training time, reward poisoning, multiple offline victims.

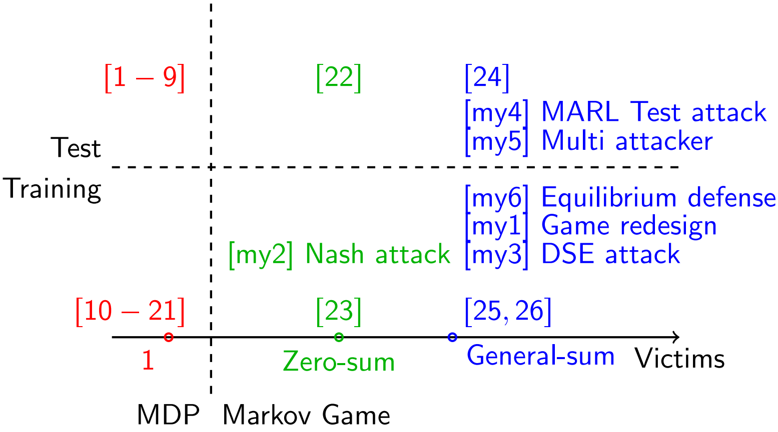
- Future work:

my4 MARL Test Attack: Test time, state or action manipulation, pre-trained victims.

my5 Equilibrium Defense: Training or test time, attack-aware victims.

my6 Multi Attacker: Training or test time, multiple attackers.

Adversarial RL Literature



[my1] Game Redesign

my1 Joint work ($\approx$ 15% contribution) with Yuzhe Ma (main author), and Jerry Zhu.

- Victim setting:

1. The victims are no-regret online learners with $O\left(T^{\alpha}\right)$ regret, e.g. EXP3.P.
2. The victims participate in an n-player general-sum bandit game with original reward $r^{o}\left(a\right) \in \left[-1, 1\right]^{n}$ for action profile $a = \left(a_1, a_2, ..., a_n\right)$.

Attacker Setting

- Attacker setting:

1. The attacker wants the victims to take a target (deterministic) policy $\pi^{\dagger} = \left(\pi_1^{\dagger}, \pi_2^{\dagger}, ..., \pi_n^{\dagger}\right)$ as often as possible, i.e. maximize $\sum_{t=1}^{T} \mathbb{1}_{\left(a_t = \pi^{\dagger}\right)}$.
2. The attacker can modify the rewards that the victims see from $r^o\left(a\right)$ to $r^{\dagger}\left(a\right)$.
3. The attacker wants sublinear design cost $\sum_{t=1}^{T} \left\| r^o\left(a_t\right) - r_t^{\dagger}\left(a_t\right) \right\|_p$.

Interior Design Example

- Suppose $\pi^\dagger = (1, 1)$, the attacker can redesign the game r^o to $r^\dagger$,

$$r^o = \begin{bmatrix} (0,0) & \left(-1, \boxed{1}\right) & \left(\boxed{1}, -1\right) \\ \left(\boxed{1}, -1\right) & (0,0) & \left(-1, \boxed{1}\right) \\ \left(-1, \boxed{1}\right) & \left(\boxed{1}, -1\right) & (0,0) \end{bmatrix},$$

$$r_1^\dagger = r_2^\dagger = \ldots = \begin{bmatrix} \left(\boxed{0}, \boxed{0}\right) & \left(\boxed{0.1}, -0.1\right) & \left(\boxed{0.1}, -0.1\right) \\ \left(-0.1, \boxed{0.1}\right) & (0,0) & (0,0) \\ \left(-0.1, \boxed{0.1}\right) & (0,0) & (0,0) \end{bmatrix}.$$

Interior Design Algorithm

- Given $r^o(a) \in [-1, 1]$, first consider the interior case when $r^o(\pi^\dagger) > -1$.
- Assumption: $r^o(\pi^\dagger) \geqslant -1 + \rho$, for some $\rho > 0$.
- Attack: $r^\dagger_{i,t}(a) = \begin{cases} r^o_i(\pi^\dagger) + \left(1 - \dfrac{d(a_t)}{n}\right)\rho & \text{if } a_{i,t} = \pi^\dagger_i \\ r^o_i(\pi^\dagger) - \dfrac{d(a_t)}{n}\rho & \text{if } a_{i,t} \neq \pi^\dagger_i \end{cases},$

  where $d(a_t) = \displaystyle\sum_{i=1}^{n} \mathbb{1}_{\left\{a_{i,t} = \pi^\dagger_i\right\}}$.

Interior Design Result

Theorem

Using the interior design, $\pi^\dagger$ is used $T - O\left(nT^\alpha\right)$ times while incurring design cost $O\left(n^{1+1/p}T^\alpha\right)$.

- For example, EXP3.P with L_1 cost can achieve $\pi^\dagger$ being used $T - O\left(n\sqrt{T}\right)$ times with cost $O\left(n^2\sqrt{T}\right)$.

Interior Design Proof Sketch

- Under this attack, we have,

$$r_{i,t}^{\dagger}(a) = \begin{cases} r_i^o\left(\pi^{\dagger}\right) + \left(1 - \frac{d\left(a_t\right)}{n}\right)\rho & \text{if } a_{i,t} = \pi_i^{\dagger} \\ r_i^o\left(\pi^{\dagger}\right) - \frac{d\left(a_t\right)}{n}\rho & \text{if } a_{i,t} \neq \pi_i^{\dagger} \end{cases}.$$

1. $\pi^{\dagger}$ is strictly dominant: $r_{i,t}^{\dagger}\left(\pi_{i,t}^{\dagger}, a_{-i,t}\right) = r_{i,t}^{\dagger}\left(a_{i,t}, a_{-i,t}\right) + \left(1 - \frac{1}{n}\right)\rho, \forall\, a_{i,t} \neq \pi_{i,t}^{\dagger}$.
2. $\pi^{\dagger}$ rewards remain unchanged: $r_{i,t}^{\dagger}\left(\pi^{\dagger}\right) = r_i^o\left(\pi^{\dagger}\right)$.

- No-regret learners will use the optimal action profile $\pi^{\dagger}$ in all but $O\left(T^{\alpha}\right)$ rounds while incurring $O\left(T^{\alpha}\right)$ design cost.

Boundary Design Example

- When $r^o\left(\pi^\dagger\right) = -1$, it is impossible to decrease other entries below -1: another design is needed.
- Suppose again $\pi^\dagger = (1, 1)$, then,

$$r^o = \begin{bmatrix} (-1,-1) & \left(-1,\boxed{1}\right) & \left(\boxed{1},-1\right) \\ \left(\boxed{1},-1\right) & (-1,-1) & \left(-1,\boxed{1}\right) \\ \left(-1,\boxed{1}\right) & \left(\boxed{1},-1\right) & (-1,-1) \end{bmatrix},$$

$$r_1^\dagger \approx \begin{bmatrix} \left(\boxed{-0.8},\boxed{-0.8}\right) & \left(\boxed{-0.7},-0.9\right) & \left(\boxed{-0.7},-0.9\right) \\ \left(-0.9,\boxed{-0.7}\right) & (-1,-1) & (-1,-1) \\ \left(-0.9,\boxed{-0.7}\right) & (-1,-1) & (-1,-1) \end{bmatrix},$$

Boundary Design Example Limit

$$r_1^\dagger \approx \begin{bmatrix} (\boxed{-0.8}, \boxed{-0.8}) & (\boxed{-0.7}, -0.9) & (\boxed{-0.7}, -0.9) \\ (-0.9, \boxed{-0.7}) & (-1, -1) & (-1, -1) \\ (-0.9, \boxed{-0.7}) & (-1, -1) & (-1, -1) \end{bmatrix},$$

$$r_2^\dagger \approx \begin{bmatrix} (\boxed{-0.9}, \boxed{-0.9}) & (\boxed{-0.85}, -0.95) & (\boxed{-0.85}, -0.95) \\ (-0.95, \boxed{-0.85}) & (-1, -1) & (-1, -1) \\ (-0.95, \boxed{-0.85}) & (-1, -1) & (-1, -1) \end{bmatrix},$$

$$\lim_{t \to \infty} r_t^\dagger = \begin{bmatrix} (-1, -1) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) \end{bmatrix}.$$

Boundary Design Algorithm

- Assumption: $r^{o}\left(\pi^{\dagger}\right) = -1$.
- Attack: $r_{i,t}^{\dagger}(a) = w_t r_{i,\text{ interior}}^{\dagger}(a) + (1 - w_t)\, r^{o}\left(\pi^{\dagger}\right)$, where $w_t = t^{\alpha+\varepsilon-1}$, for some $\varepsilon \in (0, 1-\alpha]$.

Boundary Design Result

Theorem

Using the boundary deisng with $\varepsilon = \dfrac{1-\alpha}{2}, \pi^{\dagger}$ *is used* $T - O\left(nT^{(1+\alpha)/2}\right)$ *times while incurring design cost* $O\left(n^{1/p}\left(1+n\right)T^{(1+\alpha)/2}\right)$.

Boundary Design Proof Sketch

- Under this attack, we have,

$$r^{\dagger}_{i,t}(a) = w_t r^{\dagger}_{i,\text{ interior}}(a) + (1 - w_t)\, r^{o}\left(\pi^{\dagger}\right), \text{ where } w_t = t^{\alpha+\varepsilon-1}.$$

1. $\pi^{\dagger}$ is strictly dominant:
$$r^{\dagger}_{i,t}\left(\pi^{\dagger}_{i,t}, a_{-i,t}\right) = r^{\dagger}_{i,t}\left(a_{i,t}, a_{-i,t}\right) - \left(1 - \frac{1}{n}\right)\rho w_t, \forall\ a_{i,t} \neq \pi^{\dagger}_{i,t}.$$
2. $\pi^{\dagger}$ rewards are almost unchanged:
$$\left\| r^{\dagger}_{i,t}\left(\pi^{\dagger}\right) - r^{o}_{i}\left(\pi^{\dagger}\right) \right\|_{p} \leqslant 2bn^{1/p} w_t.$$

- No-regret learners will use the optimal action profile $\pi^{\dagger}$ in all but $O\left(T^{(1+\alpha)/2}\right)$ rounds while incurring $O\left(T^{(1+\alpha)/2}\right)$ design cost.

[my2] Nash Attack

- In [my1], the attacker modifies the victims' rewards during online learning.
- In [my2], and [my3], the attacker modifies the rewards in an offline data set.

Victim Setting

my2 Joint work ($\approx$ 75% contribution) with Jeremy McMahan, Jerry Zhu, Qiaomin Xie. (Thanks: Yudong Chen)

- Victim setting:

1. The victims are uncertainty-aware offline learners that use additive bonus terms β when estimating the Q function, i.e. $Q = \hat{R} - \beta + \mathbb{E}_{\hat{P}}[V']$.
2. The victims learn a two-player zero-sum Markov game from a training set $\left\{ \left(\left(s_t^{(k)}, a_t^{(k)}, r_t^{(k)} \right)_{t=1}^{T} \right) \right\}_{k=1}^{K}$, with $r_t^{(k)} \in [0, 1]$.

Attacker Setting

- Attacker setting:

1. The attacker wants the victims to learn a target (deterministic) policy $\pi^\dagger$ as the unique Markov perfect (Nash) equilibrium.
2. The attacker can modify the rewards in the training set from r^o to $r^\dagger$.
3. The attacker minimizes the reward modification cost $\left\| r^\dagger - r^o \right\|$, e.g. $\sum_{k=1}^{K} \sum_{t=1}^{T} \left\| r_t^{\dagger,(k)} - r_t^{o,(k)} \right\|_1$.
4. The attacker does not know $\hat{R}$ and $\hat{P}$, but assumes $\left| \hat{R} - R^{(\mathrm{MLE})} \right| < \rho^{(R)}$ and $\left\| \hat{P} - P^{(\mathrm{MLE})} \right\|_1 < \rho^{(P)}$.

iNash Formulation

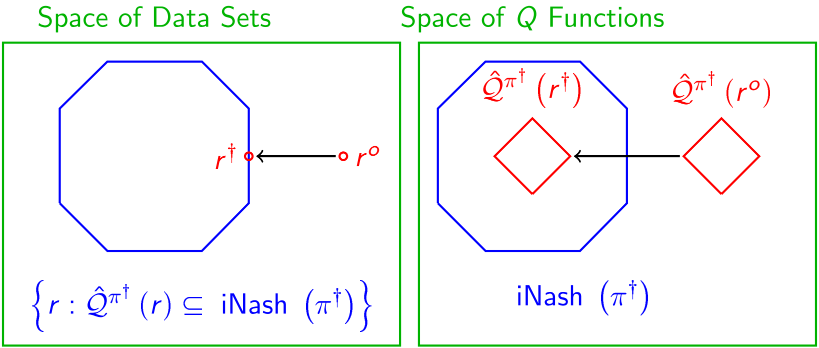
- The attack can be formulated as

$$\min_{r^\dagger} \left\| r^\dagger - r^o \right\|$$
$$\text{s.t.} \hat{\mathcal{Q}}^{\pi^\dagger}\left(r^\dagger; \rho^{(R)}, \rho^{(P)}\right) \subseteq \text{ iNash } \left(\pi^\dagger\right),$$

where,

1. $\hat{\mathcal{Q}}^\pi\left(r\right)$ is the set of plausible Q functions computed based on r evaluated on π,
2. iNash $\left(\pi\right)$ is the inverse Nash polytope of Q functions such that π is the strict Markov perfect (Nash) equilibrium.

iNash Diagram



Feasibility

Theorem

The attack is feasible if $\rho_t^{(R)}(s,a) + |\beta_t(s,a)| < \dfrac{1}{4T}, \forall\ t, s,$ *and actions* a *such that* $a_1 = \pi^{\dagger}_{1,t}(s)$ *or* $a_2 = \pi^{\dagger}_{2,t}(s)$.

- For example, if $\rho^{(R)} = 0$ and $\beta = \dfrac{c}{\sqrt{N_t(s,a)}}$, then the condition is a data coverage condition, $N_t(s,a) > 16cT^2$ for actions profiles in the same row or column as $\pi^{\dagger}$ in the stage game matrices.

Feasible Example

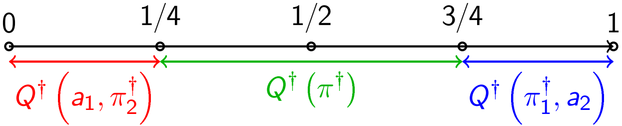
- Suppose $\pi^\dagger = (1, 1)$ in a stage game, then the following attack is feasible under the previous feasibility condition,

$a_1 \backslash a_2$	1	2	3	4
1	0.5	1	1	1
2	0	-	-	-
3	0	-	-	-
4	0	-	-	-

- Unspecified cells' corresponding rewards do not need to be poisoned.

Feasibility Proof Sketch

- The condition $\rho_t^{(R)}(s,a) + |\beta_t(s,a)| < 1/(4T)$ implies that the cumulated confidence interval width for R and P in the future periods is bounded by $1/4$.
- In period t, state s, for every $a_1 \neq \pi_1^\dagger$ and $a_2 \neq \pi_2^\dagger$, the Q values have the following relationship.



- Therefore, $\pi_t^\dagger(s)$ is the strict, thus unique, Nash equilibrium in every stage game (t,s).

Linear Program Formulation

- The attacker's problem is given by,

$$\min_{r^\dagger} \sum_{k=1}^{K} \sum_{t=1}^{T} \left\| r_t^{\dagger,(k)} - r_t^{o,(k)} \right\|_1$$

s.t. for every t, s, and $Q_t^\dagger \in \hat{\mathcal{Q}}^{\pi^\dagger}\left(r^\dagger\right)$,

$$Q_t^\dagger\left(s, \pi_t^\dagger(s)\right) > Q_t^\dagger\left(s, \left(a_1, \pi_{t,2}^\dagger(s)\right)\right), \forall\, a_1 \neq \pi_{t,1}^\dagger(s),$$
$$Q_t^\dagger\left(s, \pi_t^\dagger(s)\right) < Q_t^\dagger\left(s, \left(\pi_{t,1}^\dagger(s), a_2\right)\right), \forall\, a_2 \neq \pi_{t,2}^\dagger(s).$$

- Since $\hat{\mathcal{Q}}^{\pi}(r)$ are polytopes, this problem can be formulated as a linear program and solved efficiently.

[my3] DSE Attack

my3 Joint work ($\approx 50\%$ contribution) with Jeremy McMahan, Jerry Zhu, Qiaomin Xie. (Thanks: Yudong Chen)

- The settings are similar to the Nash Attack [my2], except there are n victims learning general-sum Markov games.
- iDSE (Markov perfect dominant strategy equilibrium) is used in place of iNash: the feasibility conditions are similar, and the attack can also be converted into a linear program.

Future Work

- [my4], [my5], [my6] are incomplete future work, and focus mostly on modes of attack other than training time reward poisoning.

[my4] MARL Test Attack

- The setting:

1. The attacker knows the victims' pre-trained policy $\pi = (\pi_1, \pi_2, ..., \pi_n)$.
2. The attacker wants to minimize some function of the victims' rewards $g((r_1, r_2, ..., r_n))$.
3. The attacker may poison the environment at test time, for example, modify the perceived states from s_t to $s_t^\dagger$.

Motivation

- Single-agent test time attacks have been studied, but they can be extended to the multi-agent reinforcement learning setting.

Attacker Goal

- The attacker wants to minimize some social welfare $g\left(\left(r_1, r_2, ..., r_n\right)\right)$ of the victims, for example,

1. Utilitarian: $g\left(r\right) = \displaystyle\sum_{i=1}^{n} r_i\left(\pi\right)$.
2. Rawlsian: $g\left(r\right) = \min_i r_i\left(\pi\right)$.
3. Other functions of rewards such as $g\left(r\right) = \max_i r_i\left(\pi\right) - \min_i r_i\left(\pi\right)$.

Attacker Action

- The attacker may modify one of the following during test time,

1. Perceived state, common to all victims, i.e. change $s_t \to s_t^\dagger$, but $P\left(s_{t+1} | a_t, s_t\right)$ stays the same.
2. Perceived state, different to different victims, i.e. change $s_t \to \left(s_{t,1}^\dagger, s_{t,2}^\dagger, ..., s_{t,n}^\dagger\right)$, but $P\left(s_{t+1} | a_t, s_t\right)$ stays the same.
3. True state, i.e. change $s_t \to s_t^\dagger$, and $P\left(s_{t+1} | a_t, s_t\right) \to P\left(s_{t+1} | a_t, s_t^\dagger\right)$.
4. Victim action, i.e. change $a_t \to a_t^\dagger$, and $P\left(s_{t+1} | a_t, s_t\right) \to P\left(s_{t+1} | a_t^\dagger, s_t\right)$.

Roadmap to Solve [my4]

- The original Markov game, given by its states, actions, transitions, and rewards, say $\mathcal{G} = (\mathcal{S}, \mathcal{A}, P, R)$.
- In the perceived common state attack, the attacker's problem can be formulated as a meta Markov decision process $\mathcal{M} = (\mathcal{S}', \mathcal{A}', P', R')$, where,

1. The meta states $\mathcal{S}' = \mathcal{S}$.
2. The meta actions $\mathcal{A}' \subseteq \mathcal{S}$.
3. The meta transitions $P'_t\left(\left(s_{t+1}, a_{t+1}\right) \mid \left(s_t, a_t\right), s^{\dagger}_t\right) = P_t\left(s_{t+1} | s^{\dagger}_t, a_t\right) \pi_{t+1}\left(a_{t+1} | s_{t+1}\right)$.
4. The meta rewards $R'_t\left(\left(s_t, a_t\right), s^{\dagger}_t\right) = -g\left(r_t\left(s^{\dagger}_t, a_t\right)\right)$.

Roadmap, Continued

- The meta MDP $\mathcal{M}$ can be solved using any reinforcement learning or planning algorithms.
- There might be special algorithms to solve $\mathcal{M}$ more efficiently since the meta action space might be large.
- The setting where the attacker does not know $\mathcal{G}$ or π could be studied.
- Experiments could be implemented.

[my5] Equilibrium Defense

- The setting:

1. The attacker wants to minimize some social welfare $g\left(\left(r_1, r_2, ..., r_n\right)\right)$.
2. The victims want to maximize expected discounted individual rewards r_i.
3. The attacker and victims simultaneously select and commit to a perceived state attack $\nu : \mathcal{S} \to \mathcal{S}$ and a policy $\pi = \left(\pi_1, \pi_2, ..., \pi_n\right)$, with $\pi_i : \mathcal{S} \to \mathcal{A}_i$.

Motivation

- In MARL Test Attack [my4] and most of the attack-defense literature, either the victim has a fixed policy, or the attacker has a fixed attack algorithm, and the other agent best responds to the fixed action. Both models are not realistic and equilibrium attack-defense should be studied instead.

Roadmap to Solve [my5]

- The problem can be formulated as a static game $\mathcal{G} = (\mathcal{A}', R')$, where,

1. The meta actions
   $\mathcal{A}' = (\mathcal{S} \to \mathcal{S}, \mathcal{S} \to \mathcal{A}_1, \mathcal{S} \to \mathcal{A}_2, ..., \mathcal{S} \to \mathcal{A}_n)$.
2. The meta rewards
   $R' (\nu, \pi_1, \pi_2, ..., \pi_n) = (-g (V (\pi (\nu))), V (\pi (\nu)))$, where
   $$V (\pi (\nu)) = \sum_{t=1}^{\infty} \gamma^t \mathbb{E}_P [R (s_t, \pi_1 (\nu (s_t)), ..., \pi_n (\nu (s_t)))].$$

Roadmap, Continued

- The meta game $\mathcal{G}$ can be solved using a Nash solver, e.g. a linear program when the game is zero-sum.
- There might be special classes of equilibria that are easier to solve since the meta action space might be large.
- The equilibrium policy π might correspond to some robust policy in the RL literature.
- Training time equilibrium attack defense could also be studied.

[my6] Multi Attacker

- The setting:

1. Multiple attackers $j \in [m]$, each attacks a subset of the victims.
2. Each attacker wants to minimize a different social welfare $g_j\left(\left(r_1, r_2, ..., r_n\right)\right)$.
3. Each attacker may modify the perceived state of the set of victims it attacks.

Motivation

- In MARL Test Attack [my4] and most of the training or test time attack literature, there is only one attacker. The problem with multiple attackers with different objectives is an interesting problem with many real-world applications.

Roadmap to Solve [my6]

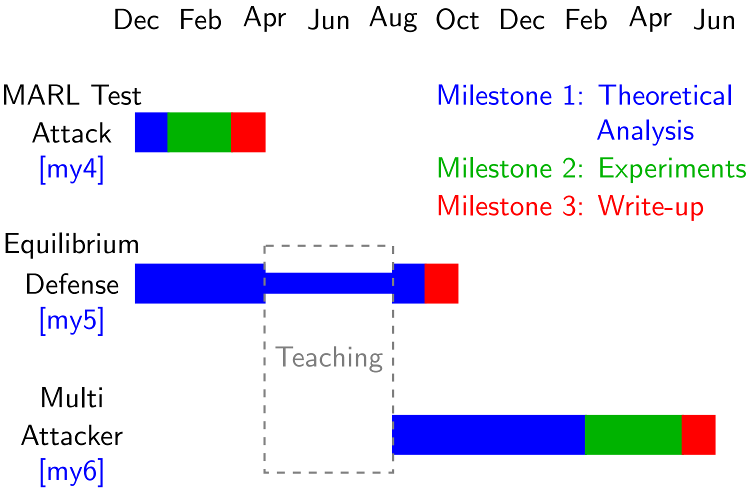
- The original Markov game, given by its states, actions, transitions, and rewards, say $\mathcal{G} = (\mathcal{S}, \mathcal{A}, P, R)$.
- In the perceived state attack where each attacker attacks a single victim, the attackers' problem can be formulated as a meta Markov game $\mathcal{M} = (\mathcal{S}', \mathcal{A}', P', R')$, where,

1. The meta states $\mathcal{S}' = \mathcal{S}$.
2. The meta actions $\mathcal{A}' \subseteq \mathcal{S}^{n.}$
3. The meta transitions $P'_t\left(\left(s_{t+1}, a_{t+1}\right) \middle| \left(s_t, a_t\right), s^{\dagger}_t\right) = P_t\left(s_{t+1} \middle| s^{\dagger}_t, a_t\right) \pi_{t+1}\left(a_{t+1} \middle| s_{t+1}\right)$.
4. The meta rewards $R'_t\left(\left(s_t, a_t\right), s^{\dagger}_t\right) = -g\left(r_t\left(s^{\dagger}_t, a_t\right)\right)$.

Roadmap, Continued

- The meta Markov Game $\mathcal{G}$ can be solved using any multi-agent reinforcement learning or planning algorithms.
- There might be special algorithms to solve $\mathcal{G}$ more efficiently since the meta action space might be large.
- The setting where the attackers do not know $\mathcal{G}$ or π could be studied.
- Experiments could be implemented.
- Training time attacks with multiple attackers could also be studied.

Thesis Expected Timeline



Thank you!

References I

[1] Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu.
Stealthy and efficient adversarial attacks against deep reinforcement learning.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5883–5891, 2020.

[2] 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.
Advances in Neural Information Processing Systems, 33:21024–21037, 2020.

References II

[3] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel.
Adversarial attacks on neural network policies.
arXiv preprint arXiv:1702.02284, 2017.

[4] Jernej Kos and Dawn Song.
Delving into adversarial attacks on deep policies.
arXiv preprint arXiv:1705.06452, 2017.

[5] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary.
Robust deep reinforcement learning with adversarial attacks.
arXiv preprint arXiv:1712.03632, 2017.

References III

[6] Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh.
Robust reinforcement learning on state observations with learned optimal adversary.
arXiv preprint arXiv:2101.08452, 2021.

[7] Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang.
Who is the strongest enemy? towards optimal and efficient evasion attacks in deep rl.
arXiv preprint arXiv:2106.05087, 2021.

References IV

[8] Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li.
Crop: Certifying robust policies for reinforcement learning through functional smoothing.
arXiv preprint arXiv:2106.09292, 2021.

[9] Edgar Tretschk, Seong Joon Oh, and Mario Fritz.
Sequential attacks on agents for long-term adversarial goals.
arXiv preprint arXiv:1805.12487, 2018.

[10] Haoqi Zhang and David C Parkes.
Value-based policy teaching with active indirect elicitation.
In *AAAI*, volume 8, pages 208–214, 2008.

References V

[11] Haoqi Zhang, David C Parkes, and Yiling Chen.
Policy teaching through reward function learning.
In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 295–304, 2009.

[12] Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu.
Policy poisoning in batch reinforcement learning and control.
Advances in Neural Information Processing Systems, 32, 2019.

[13] Yunhan Huang and Quanyan Zhu.
Deceptive reinforcement learning under adversarial manipulations on cost signals.
In *International Conference on Decision and Game Theory for Security*, pages 217–237. Springer, 2019.

References VI

[14] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu.
Adaptive reward-poisoning attacks against reinforcement learning.
In *International Conference on Machine Learning*, pages 11225–11234. PMLR, 2020.

[15] Anshuka Rangi, Haifeng Xu, Long Tran-Thanh, and Massimo Franceschetti.
Understanding the limits of poisoning attacks in episodic reinforcement learning.
arXiv preprint arXiv:2208.13663, 2022.

[16] Huazheng Wang, Haifeng Xu, and Hongning Wang.
When are linear stochastic bandits attackable?
In *International Conference on Machine Learning*, pages 23254–23273. PMLR, 2022.

References VII

[17] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla.
Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning.
In *International Conference on Machine Learning*, pages 7974–7984. PMLR, 2020.

[18] Amin Rakhsha, Xuezhou Zhang, Xiaojin Zhu, and Adish Singla.
Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments.
arXiv preprint arXiv:2102.08492, 2021.

References VIII

[19] Yanchao Sun, Da Huo, and Furong Huang.
Vulnerability-aware poisoning mechanism for online rl with unknown dynamics.
arXiv preprint arXiv:2009.00774, 2020.

[20] Guanlin Liu and Lifeng Lai.
Provably efficient black-box action poisoning attacks against reinforcement learning.
Advances in Neural Information Processing Systems, 34:12400–12410, 2021.

[21] Kiarash Banihashem, Adish Singla, Jiarui Gan, and Goran Radanovic.
Admissible policy teaching through reward design.
arXiv preprint arXiv:2201.02185, 2022.

References IX

[22] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell.
Adversarial policies: Attacking deep reinforcement learning.
arXiv preprint arXiv:1905.10615, 2019.

[23] Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing.
Adversarial policy learning in two-player competitive games.
In *International Conference on Machine Learning*, pages 3910–3919. PMLR, 2021.

[24] Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang.
Certifiably robust policy learning against adversarial communication in multi-agent systems.
arXiv preprint arXiv:2206.10158, 2022.

References X

[25] Junyan Liu, Shuai Li, and Dapeng Li.
Cooperative stochastic multi-agent multi-armed bandits robust to adversarial corruptions.
arXiv preprint arXiv:2106.04207, 2021.

[26] Martin Figura, Krishna Chaitanya Kosaraju, and Vijay Gupta.
Adversarial attacks in consensus-based multi-agent reinforcement learning.
In *2021 American Control Conference (ACC)*, pages 3050–3055. IEEE, 2021.