Adaptive Reward-Poisoning Attacks against Reinforcement Learning

Xuezhou Zhang [1] **Yuzhe Ma** [1] **Adish Singla** [2] **Xiaojin Zhu** [1]

Abstract

In reward-poisoning attacks against reinforcement learning (RL), an attacker can perturb the environment reward r_t into $r_t + \delta_t$ at each step, with the goal of forcing the RL agent to learn a nefarious policy. We categorize such attacks by the infinity-norm constraint on δ_t: We provide a lower threshold below which reward-poisoning attack is infeasible and RL is certified to be safe; we provide a corresponding upper threshold above which the attack is feasible. Feasible attacks can be further categorized as non-adaptive where δ_t depends only on (s_t, a_t, s_{t+1}), or adaptive where δ_t depends further on the RL agent's learning process at time t. Non-adaptive attacks have been the focus of prior works. However, we show that under mild conditions, adaptive attacks can achieve the nefarious policy in steps polynomial in state-space size $|S|$, whereas non-adaptive attacks require exponential steps. We provide a constructive proof that a Fast Adaptive Attack strategy achieves the polynomial rate. Finally, we show that empirically an attacker can find effective reward-poisoning attacks using state-of-the-art deep RL techniques.

1. Introduction

In many reinforcement learning (RL) applications the agent extracts reward signals from user feedback. For example, in recommendation systems the rewards are often represented by user clicks, purchases or dwell time (Zhao et al., 2018; Chen et al., 2019); in conversational AI, the rewards can be user sentiment or conversation length (Dhingra et al., 2016; Li et al., 2016). In such scenarios, an adversary can manipulate user feedback to influence the RL agent in nefarious ways. Figure 1 describes a hypothetical scenario of how conversational AI can be attacked. One real-world example is that of the chatbot Tay, which was quickly corrupted by a group of Twitter users who deliberately taught it misogynistic and racist remarks shortly after its release (Neff & Nagy, 2016). Such attacks reveal significant security threats in the application of reinforcement learning.

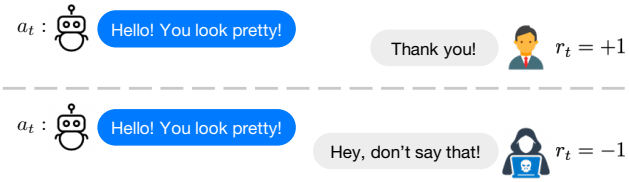


Figure 1. Example: an RL-based conversational AI is learning from real-time conversations with human users. the chatbot says "Hello! You look pretty!" and expects to learn from user feedback (sentiment). A benign user will respond with gratitude, which is decoded as a positive reward signal. An adversarial user, however, may express anger in his reply, which is decoded as a negative reward signal.

In this paper, we formally study the problem of *training-time attack on RL via reward poisoning*. As in standard RL, the RL agent updates its policy π_t by performing action a_t at state s_t in each round t. The environment Markov Decision Process (MDP) generates reward r_t and transits the agent to s_{t+1}. However, the attacker can change the reward r_t to $r_t + \delta_t$, with the goal of driving the RL agent toward a target policy $\pi_t \to \pi^\dagger$.

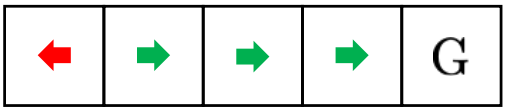


Figure 2. A chain MDP with attacker's target policy $\pi^\dagger$

Figure 2 shows a running example that we use throughout the paper. The episodic MDP is a linear chain with five states, with left or right actions and no movement if it hits the boundary. Each move has a -0.1 negative reward, and G is the absorbing goal state with reward 1. Without attack, the optimal policy π^* would be to always move right. The attacker's goal, however, is to force the agent to learn the nefarious target policy $\pi^\dagger$ represented by the arrows in Figure 2. Specifically, the attacker wants the agent to move left and hit its head against the wall whenever the agent is at the left-most state.

[1]University of Wisconsin-Madison [2]Max Planck Institute for Software Systems (MPI-SWS). Correspondence to: Xuezhou Zhang <xzhang784@wisc.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

Our main contributions are:

1. We characterize conditions under which such attacks are guaranteed to fail (thus RL is safe), and vice versa;
2. In the case where an attack is feasible, we provide upper bounds on the attack cost in the process of achieving $\pi^\dagger$;
3. We show that effective attacks can be found empirically using deep RL techniques.

2. Related Work

Test-time attacks against RL Prior work on adversarial attacks against reinforcement learning focused primarily on *test-time*, where the RL policy π is pre-trained and fixed, and the attacker manipulates the perceived state s_t to $s_t^\dagger$ in order to induce undesired action (Huang et al., 2017; Lin et al., 2017; Kos & Song, 2017; Behzadan & Munir, 2017). For example, in video games the attacker can make small pixel perturbation to a frame (Goodfellow et al., 2014)) to induce an action $\pi(s_t^\dagger) \neq \pi(s_t)$. Although test-time attacks can severely impact the performance of a deployed and fixed policy π, they do not modify π itself. For ever-learning agents, however, the attack surface includes π. This motivates us to study training-time attack on RL policy.

Reward Poisoning: Reward poisoning has been studied in bandits (Jun et al., 2018; Peltola et al., 2019; Altschuler et al., 2019; Liu & Shroff, 2019; Ma et al., 2018), where the authors show that adversarially perturbed reward can mislead standard bandit algorithms to pull a suboptimal arm or suffer large regret.

Reward poisoning has also been studied in *batch RL* (Zhang & Parkes, 2008; Zhang et al., 2009; Ma et al., 2019) where rewards are stored in a pre-collected batch data set by some behavior policy, and the attacker modifies the batch data. Because all data are available to the attacker at once, the batch attack problem is relatively easier. This paper instead focuses on the *online* RL attack setting where reward poisoning must be done on the fly.

(Huang & Zhu, 2019) studies a restricted version of reward poisoning, in which the perturbation only depend on the current state and action: $\delta_t = \phi(s_t, a_t)$. While such restriction guarantees the convergence of Q-learning under the perturbed reward and makes the analysis easier, we show both theoretically and empirically that such restriction severely harms attack efficiency. Our paper subsumes their results by considering more powerful attacks that can depend on the RL victim's Q-table Q_t. Theoretically, our analysis does not require the RL agent's underlying Q_t to converge while still providing robustness certificates; see section 4.

Reward Shaping: While this paper is phrased from the adversarial angle, the framework and techniques are also applicable to the *teaching* setting, where a *teacher* aims to guide the agent to learn the *optimal policy* as soon as possible, by designing the reward signal. Traditionally, reward shaping and more specifically potential-based reward shaping (Ng et al., 1999) has been shown able to speed up learning while preserving the optimal policy. (Devlin & Kudenko, 2012) extend potential-based reward shaping to be time-varying while remains policy-preserving. More recently, intrinsic motivations(Schmidhuber, 1991; Oudeyer & Kaplan, 2009; Barto, 2013; Bellemare et al., 2016) was introduced as a new form of reward shaping with the goal of encouraging exploration and thus speed up learning. Our work contributes by mathematically defining the teaching via reward shaping task as an optimal control problem, and provide computational tools that solve for problem-dependent high-performing reward shaping strategies.

3. The Threat Model

In the reward-poisoning attack problem, we consider three entities: the environment MDP, the RL agent, and the attacker. Their interaction is formally described by Alg 1.

The environment MDP is $\mathcal{M} = (S, A, R, P, \mu_0)$ where S is the state space, A is the action space, $R : S \times A \times S \to \mathbb{R}$ is the reward function, $P : S \times A \times S \to \mathbb{R}$ is the transition probability, and $\mu_0 : S \to \mathbb{R}$ is the initial state distribution. We assume S, A are finite, and that a uniformly random policy can visit each (s, a) pair infinitely often.

We focus on an RL agent that performs standard Q-learning defined by a tuple $\mathcal{A} = (Q_0, \varepsilon, \gamma, \{\alpha_t\})$, where Q_0 is the initial Q table, ε is the random exploration probability, γ is the discounting factor, $\{\alpha_t\}$ is the learning rate scheduling as a function of t. This assumption can be generalized: in the additional experiments provided in appendix G.2, we show how the same framework can be applied to attack general RL agents, such as DQN. Denote Q^* as the optimal Q table that satisfies the Bellman's equation:

$$Q^*(s,a) = \mathbb{E}_{P(s'|s,a)}\left[R(s,a,s') + \gamma \max_{a' \in A} Q^*(s',a')\right] \tag{1}$$

and denote the corresponding optimal policy as $\pi^*(s) = \arg\max_a Q^*(s,a)$. For notational simplicity, we assume π^* is unique, though it is easy to generalize to multiple optimal policies, since most of our analyses happen in the space of value functions.

The Threat Model The attacker sits between the environment and the RL agent. In this paper we focus on white-box attacks: the attacker has knowledge of the environment MDP and the RL agent's Q-learning algorithm, except for their future randomness. Specifically, at time t the attacker observes the learner Q-table Q_t, state s_t, action a_t, the environment transition s_{t+1} and reward r_t. The attacker

Algorithm 1 Reward Poisoning against Q-learning

PARAMETERS: Agent parameters $\mathcal{A} = (Q_0, \varepsilon, \gamma, \{\alpha_t\})$, MDP parameters $\mathcal{M} = (S, A, R, P, \mu_0)$.

for $t = 0, 1, ...$ **do**
  agent at state s_t, has Q-table Q_t.
  agent acts according to ε-greedy behavior policy

$$a_t \leftarrow \begin{cases} \arg\max_a Q_t(s_t, a), & \text{w.p. } 1-\varepsilon \\ \text{uniform from } A, & \text{w.p. } \varepsilon. \end{cases} \quad (2)$$

  environment transits $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, produces reward $r_t = R(s_t, a_t, s_{t+1})$.
  attacker poisons the reward to $r_t + \delta_t$.
  agent receives $(s_{t+1}, r_t + \delta_t)$, performs Q-learning update:

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha_t) Q_t(s_t, a_t) + \alpha_t \left(r_t + \delta_t + \gamma \max_{a' \in A} Q_t(s_{t+1}, a') \right) \quad (3)$$

  environment resets if episode ends: $s_{t+1} \sim \mu_0$.
end for

can choose to add a perturbation $\delta_t \in \mathbb{R}$ to the current environmental reward r_t. The RL agent receives poisoned reward $r_t + \delta_t$. We assume the attack is inf-norm bounded: $|\delta_t| \le \Delta, \forall t$.

There can be many possible attack goals against an RL agent: forcing the RL agent to perform certain actions; reaching or avoiding certain states; or maximizing its regret. In this paper, we focus on a specific attack goal: **policy manipulation**. Concretely, the goal of policy manipulation is to force a target policy $\pi^\dagger$ on the RL agent for as many rounds as possible.

Definition 1. *Target (partial) policy $\pi^\dagger : S \mapsto 2^A$: For each $s \in S$, $\pi^\dagger(s) \subseteq A$ specifies the set of actions desired by the attacker.*

The partial policy $\pi^\dagger$ allows the attacker to desire multiple target actions on one state. In particular, if $\pi^\dagger(s) = A$ then s is a state that the attacker "does not care." Denote $S^\dagger = \{s \in S : \pi^\dagger(s) \neq A\}$ the set of **target states** on which the attacker does have a preference. In many applications, the attacker only cares about the agent's behavior on a small set of states, namely $|S^\dagger| \ll |S|$.

For RL agents utilizing a Q-table, a target policy $\pi^\dagger$ induces a set of Q-tables:

Definition 2. *Target Q-table set*

$$\mathcal{Q}^\dagger := \{Q : \max_{a \in \pi^\dagger(s)} Q(s,a) > \max_{a \notin \pi^\dagger(s)} Q(s,a), \forall s \in S^\dagger\}$$

If the target policy $\pi^\dagger$ always specifies a singleton action or does not care on all states, then $\mathcal{Q}^\dagger$ is a convex set. But in general when $1 < |\pi^\dagger(s)| < |A|$ on any s, $\mathcal{Q}^\dagger$ will be a union of convex sets but itself can be in general non-convex.

4. Theoretical Guarantees

Now, we are ready to formally define the *optimal attack* problem. At time t, the attacker observes an *attack state* (N.B. distinct from MDP state s_t):

$$\xi_t := (s_t, a_t, s_{t+1}, r_t, Q_t) \in \Xi \quad (4)$$

which jointly characterizes the MDP and the RL agent. The attacker's goal is to find an *attack policy* $\phi : \Xi \to [-\Delta, \Delta]$, where for $\xi_t \in \Xi$ the *attack action* is $\delta_t := \phi(\xi_t)$, that minimizes the number of rounds on which the agent's Q_t disagrees with the attack target $\mathcal{Q}^\dagger$:

$$\min_{\phi} \quad \mathbb{E}_\phi \sum_{t=0}^{\infty} \mathbf{1}[Q_t \notin \mathcal{Q}^\dagger], \quad (5)$$

where the expectation accounts for randomness in Alg 1. We denote $J_\infty(\phi) = E_\phi \sum_{t=0}^{\infty} \mathbf{1}[Q_t \notin \mathcal{Q}^\dagger]$ the total attack cost, and $J_T(\phi) = E_\phi \sum_{t=0}^{T} \mathbf{1}[Q_t \notin \mathcal{Q}^\dagger]$ the finite-horizon cost. We say the attack is *feasible* if (5) is finite.

Next, we characterize attack feasibility in terms of poison magnitude constraint Δ, as summarized in Figure 3. Proofs to all the theorems can be found in the appendix.

4.1. Attack Infeasibility

Intuitively, smaller Δ makes it harder for the attacker to achieve the attack goal. We show that there is a threshold Δ_1 such that for any $\Delta < \Delta_1$ the RL agent is eventually safe, in that $\pi_t \to \pi^*$ the correct MDP policy. This implies that (5) is infinite and the attack is infeasible. There is a potentially larger Δ_2 such that for any $\Delta < \Delta_2$ the attack is also infeasible, though π_t may not converge to π^*.

While the above statements are on π_t, our analysis is via the RL agent's underlying Q_t. Note that under attack the rewards $r_t + \delta_t$ are no longer stochastic, and we cannot utilize the usual Q-learning convergence guarantee. Nonetheless, we show that Q_t is bounded in a polytope in the Q-space.

Theorem 1 (Boundedness of Q-learning)**.** *Assume that $\delta_t < \Delta$ for all t, and the stepsize α_t's satisfy that $\alpha_t \le 1$ for all t, $\sum \alpha_t = \infty$ and $\sum \alpha_t^2 < \infty$. Let Q^* be defined as (1). Then, for any attack sequence $\{\delta_t\}$, there exists $N \in \mathbb{N}$ such that, with probability 1, for all $t \ge N$, we have*

$$Q^*(s,a) - \frac{\Delta}{1-\gamma} \le Q_t(s,a) \le Q^*(s,a) + \frac{\Delta}{1-\gamma}. \quad (6)$$

Remark 1: The bounds in Theorem 1 are in fact tight. The lower and upper bound can be achieved by setting $\delta_t = -\Delta$ or $+\Delta$ respectively.

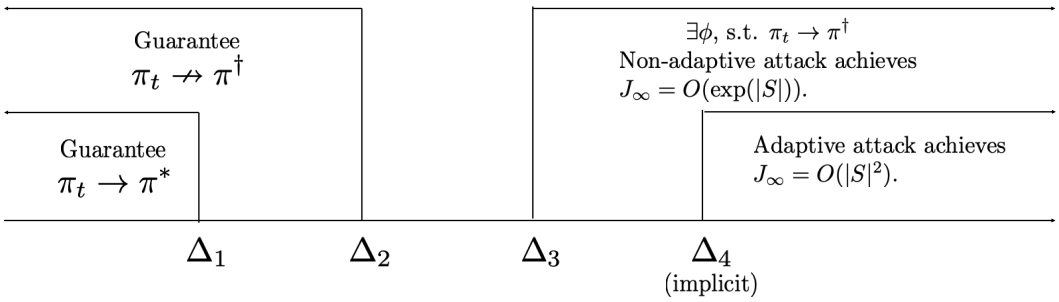


Figure 3. A summary diagram of the theoretical results.

We immediately have the following two infeasibility certificates.

Corollary 2 (Strong Infeasibility Certificate)**.** *Define*

$$\Delta_1 = (1-\gamma)\min_s \left[Q^*(s,\pi^*(s)) - \max_{a\neq\pi^*(s)} Q^*(s,a) \right]/2.$$

If $\Delta < \Delta_1$, there exist $N \in \mathbb{N}$ such that, with probability 1, for all $t > N$, $\pi_t = \pi^*$. In other words, eventually the RL agent learns the optimal MDP policy π^* despite the attacks.

Corollary 3 (Weak Infeasibility Certificate)**.** *Given attack target policy $\pi^\dagger$, define*

$$\Delta_2 = (1-\gamma)\max_s \left[Q^*(s,\pi^*(s)) - \max_{a\in\pi^\dagger(s)} Q^*(s,a) \right]/2.$$

If $\Delta < \Delta_2$, there exist $N \in \mathbb{N}$ such that, with probability 1, for all $t > N$, $\pi_t(s) \notin \pi^\dagger(s)$ for some $s \in S^\dagger$. In other words, eventually the attacker is unable to enforce $\pi^\dagger$ (though π_t may not settle on π^* either).

Intuitively, an MDP is difficult to attack if its margin $\min_s \left[Q^*(s,\pi^*(s)) - \max_{a\neq\pi^*(s)} Q^*(s,a)\right]$ is large. This suggests a defense: for RL to be robust against poisoning, the environmental reward signal should be designed such that the optimal actions and suboptimal actions have large performance gaps.

4.2. Attack Feasibility

We now show there is a threshold Δ_3 such that for all $\Delta > \Delta_3$ the attacker can enforce $\pi^\dagger$ for all but finite number of rounds.

Theorem 4. *Given a target policy $\pi^\dagger$, define*

$$\Delta_3 = \frac{1+\gamma}{2}\max_{s\in S^\dagger}[\max_{a\notin\pi^\dagger(s)} Q^*(s,a) - \max_{a\in\pi^\dagger(s)} Q^*(s,a)]_+ \tag{7}$$

where $[x]_+ \coloneqq \max(x,0)$. Assume the same conditions on α_t as in Theorem 1. If $\Delta > \Delta_3$, there is a feasible attack policy $\phi^{sas}_{\Delta_3}$. Furthermore, $J_\infty(\phi^{sas}_{\Delta_3}) \leq O(L^5)$, where L is the covering number.

Algorithm 2 The Non-Adaptive Attack $\phi^{sas}_{\Delta_3}$

PARAMETERS: target policy $\pi^\dagger$, agent parameters $\mathcal{A} = (Q_0, \varepsilon, \gamma, \{\alpha_t\})$, MDP parameters $\mathcal{M} = (S, A, R, P, \mu_0)$, maximum magnitude of poisoning Δ.

def Init($\pi^\dagger, \mathcal{A}, \mathcal{M}$)**:**

1: Construct a Q-table Q', where $Q'(s,a)$ is defined as

$$\begin{cases} Q^*(s,a) + \dfrac{\Delta}{(1+\gamma)}, & \text{if } s \in S^\dagger, a \in \pi^\dagger(s) \\ Q^*(s,a) - \dfrac{\Delta}{(1+\gamma)}, & \text{if } s \in S^\dagger, a \notin \pi^\dagger(s) \\ Q^*(s,a), & \text{if } s \notin S^\dagger \end{cases}$$

2: Calculate a new reward function

$$R'(s,a) = Q'(s,a) - \gamma \mathbb{E}_{P(s'|s,a)}\left[\max_{a'} Q'(s',a')\right].$$

3: Define the attack policy $\phi^{sas}_{\Delta_3}$ as:

$$\phi^{sas}_{\Delta_3}(s,a) = R'(s,a) - \mathbb{E}_{P(s'|s,a)}\left[R(s,a,s)\right], \forall s,a.$$

def Attack(ξ_t)**:**

1: Return $\phi^{sas}_{\Delta_3}(s_t, a_t)$

Theorem 4 is proved by constructing an attack policy $\phi^{sas}_{\Delta_3}(s_t, a_t)$, detailed in Alg. 2. Note that this attack policy does not depend on Q_t. We call this type of attack *non-adaptive attack*. Under such construction, one can show that Q-learning converges to the target policy $\pi^\dagger$. Recall the covering number L is the upper bound on the minimum sequence length starting from any (s,a) pair and follow the MDP until all (state, action) pairs appear in the sequence (Even-Dar & Mansour, 2003). It is well-known that ε-greedy exploration has a covering time $L \leq O(e^{|S|})$ (Kearns & Singh, 2002). Prior work has constructed examples on which this bound is tight (Jin et al., 2018). We show in appendix C that on our toy example ε-greedy indeed has a covering time $O(e^{|S|})$. Therefore,

the objective value of (5) for non-adaptive attack is upper-bounded by $O(e^{|S|})$. In other words, the non-adaptive attack is slow.

4.3. Fast Adaptive Attack (FAA)

We now show that there is a fast adaptive attack ϕ^{ξ}_{FAA} which depends on Q_t and achieves J_∞ polynomial in $|S|$. The price to pay is a larger attack constraint Δ_4, and the requirement that the attack target states are sparse: $k = |S^\dagger| \leq O(\log|S|)$. The FAA attack policy ϕ^{ξ}_{FAA} is defined in Alg. 3.

Conceptually, the FAA algorithm ranks the target states in descending order by their distance to the starting states, and focusing on attacking one target state at a time. Of central importance is the temporary target policy ν_i, which is designed to navigate the agent to the currently focused target state $s^\dagger_{(i)}$, while not altering the already achieved target actions on target states of earlier rank. This allows FAA to achieve a form of program invariance: after FAA achieves the target policy in a target state $s^\dagger_{(i)}$, the target policy on target state (i) will be preserved indefinitely. We provide a more detailed walk-through of Alg. 3 with examples in appendix E.

Definition 3. *Define the shortest ε-**distance** from s to s' as*

$$d_\varepsilon(s,s') = \min_{\pi\in\Pi} \mathbb{E}_{\pi_\varepsilon}[T] \quad (9)$$
$$s.t.\ s_0 = s, s_T = s', s_t \neq s', \forall t < T$$

*where π_ε denotes the epsilon-greedy policy based on π. Since we are in an MDP, there exists a common (partial) policy $\pi_{s'}$ that achieves $d_\varepsilon(s,s')$ for all source state $s \in S$. Denote $\pi_{s'}$ as the **navigation policy** to s'.*

Definition 4. *The ε-**diameter** of an MDP is defined as the longest shortest ε-distance between pairs of states in S:*

$$D_\varepsilon = \max_{s,s'\in S} d_\varepsilon(s,s') \quad (10)$$

Theorem 5. *Assume that the learner is running ε-greedy Q-learning algorithm on an episodic MDP with ε-diameter D_ε and maximum episode length H, and the attacker aims at k distinct target states, i.e. $|S^\dagger| = k$. If Δ is large enough that the $Clip_\Delta()$ function in Alg. 3 never takes effect, then ϕ^{ξ}_{FAA} is feasible, and we have*

$$J_\infty(\phi^{\xi}_{FAA}) \leq k\frac{|S||A|H}{1-\varepsilon} + \frac{|A|}{1-\varepsilon}\left[\frac{|A|}{\varepsilon}\right]^k D_\varepsilon, \quad (11)$$

How large is D_ε? For MDPs with underlying structure as undirected graphs, such as the grid worlds, it is shown that the expected hitting time of a uniform random walk is bounded by $O(|S|^2)$(Lawler, 1986). Note that the random hitting time tightly upper bounds the optimal hitting time, a.k.a. the ε-diameter D_ε, and they match when $\varepsilon = 1$. This immediately gives us the following result:

Corollary 6. *If in addition to the assumptions of Theorem 5, the maximal episode length $H = O(|S|)$, then $J_\infty(\phi^{\xi}_{FAA}) \leq O(e^k|S|^2|A|)$ in Grid World environments. When the number of target states is small, i.e. $k \leq O(\log|S|)$, $J_\infty(\phi^{\xi}_{FAA}) \leq O(\text{poly}(|S|))$.*

Remark 2: Theorem 5 and Corollary 6 can be thought of as defining an implicit Δ_4, such that for any $\Delta > \Delta_4$, the clip function in Alg. 3 never take effect, and ϕ^{ξ}_{FAA} achieves polynomial cost.

Algorithm 3 The Fast Adaptive Attack (FAA)

PARAMETERS: target policy $\pi^\dagger$, margin η, agent parameters $\mathcal{A} = (Q_0, \varepsilon, \gamma, \{\alpha_t\})$, MDP parameters $\mathcal{M} = (S, A, R, P, \mu_0)$.
def Init($\pi^\dagger, \mathcal{A}, \mathcal{M}, \eta$):

1: Given (s_t, a_t, Q_t), define the hypothetical Q-update function without attack as $Q'_{t+1}(s_t,a_t) = (1-\alpha_t)Q_t(s_t,a_t) + \alpha_t\left(r_t + \gamma(1-EOE)\max_{a'\in A} Q_t(s_{t+1},a')\right)$.
2: Given (s_t, a_t, Q_t), denote the greedy attack function at s_t w.r.t. a target action set A_{s_t} as $g(A_{s_t})$, defined as

$$\begin{cases} \frac{1}{\alpha_t}[\max_{a\notin A_{s_t}} Q_t(s_t,a) - & \\ \quad Q'_{t+1}(s_t,a_t) + \eta]_+ & \text{if } a_t \in A_{s_t} \\ \frac{1}{\alpha_t}[\max_{a\in A_{s_t}} Q_t(s_t,a) - & \\ \quad Q'_{t+1}(s_t,a_t) + \eta]_- & \text{if } a_t \notin A_{s_t}. \end{cases} \quad (8)$$

3: Define $\text{Clip}_\Delta(\delta) = \min(\max(\delta, -\Delta), \Delta)$.
4: Rank the target states in descending order as $\{s^\dagger_{(1)}, ..., s^\dagger_{(k)}\}$, according to their shortest ε-distance to the initial state $\mathbb{E}_{s\sim\mu_0}\left[d^\varepsilon(s, s_{(i)})\right]$.
5: **for** $i = 1, ..., k$ **do**
6: Define the temporary target policy ν_i as

$$\nu_i(s) = \begin{cases} \pi_{s^\dagger_{(i)}}(s) & \text{if } s \notin \{s^\dagger_{(j)} : j \leq i\} \\ \pi^\dagger(s) & \text{if } s \in \{s^\dagger_{(j)} : j \leq i\}. \end{cases}$$

7: **end for**

def Attack(ξ_t):

1: **for** $i = 1, ..., k$ **do**
2: **if** $\arg\max_a Q_t(s^\dagger_{(i)}, a) \notin \pi^\dagger(s^\dagger_{(i)})$ **then**
3: Return $\delta_t \leftarrow \text{Clip}_\Delta(g(\{\nu_i(s_t)\}))$.
4: **end if**
5: **end for**
6: Return $\delta_t \leftarrow \text{Clip}_\Delta(g(\{\pi^\dagger(s_t)\}))$.

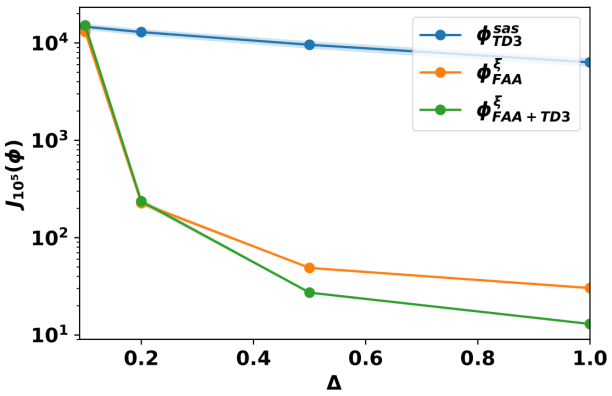


Figure 4. Attack cost $J_{10^5}(\phi)$ on different Δ's. Each curve shows mean ± 1 standard error over 1000 independent test runs.

4.4. Illustrating Attack (In)feasibility Δ Thresholds

The theoretical results developed so far can be summarized as a diagram in Figure 3. We use the chain MDP in Figure 2 to illustrate the four thresholds $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ developed in this section. On this MDP and with this attack target policy $\pi^\dagger$, we found that $\Delta_1 = \Delta_2 = 0.0069$. The two matches because this $\pi^\dagger$ is the easiest to achieve in terms of having the smallest upperbound Δ_2. Attackers whose poison magnitude $|\delta_t| < \Delta_2$ will not be able to enforce the target policy $\pi^\dagger$ in the long run.

We found that $\Delta_3 = 0.132$. We know that $\phi^{sas}_{\Delta_3}$ should be feasible if $\Delta > \Delta_3$. To illustrate this, we ran $\phi^{sas}_{\Delta_3}$ with $\Delta = 0.2 > \Delta_3$ for 1000 trials and obtained estimated $J_{10^5}(\phi^{sas}_{\Delta_3}) = 9430$. The fact that $J_{10^5}(\phi^{sas}_{\Delta_3}) \ll T = 10^5$ is empirical evidence that $\phi^{sas}_{\Delta_3}$ is feasible. We found that $\Delta_4 = 1$ by simulation. The adaptive attack ϕ^{ξ}_{FAA} constructed in Theorem 5 should be feasible with $\Delta = \Delta_4 = 1$. We run ϕ^{ξ}_{FAA} for 1000 trials and observed $J_{10^5}(\phi^{\xi}_{FAA}) = 30.4 \ll T$, again verifying the theorem. Also observe that $J_{10^5}(\phi^{\xi}_{FAA})$ is much smaller than $J_{10^5}(\phi^{sas}_{\Delta_3})$, verifying the foundamental difference in attack efficiency between the two attack policies as shown in Theorem 4 and Corollary 6.

While FAA is able to force the target policy in polynomial time, it's not necessarily the optimal attack strategy. Next, we demonstrate how to solve for the optimal attack problem in practice, and empirically show that with the techniques from Deep Reinforcement Learning (DRL), we can find efficient attack policies in a variety of environments.

5. Attack RL with RL

The attack policies $\phi^{sas}_{\Delta_3}$ and ϕ^{ξ}_{FAA} were manually constructed for theoretical analysis. Empirically, though, they do not have to be the most effective attacks under the relevant Δ constraint.

In this section, we present our key computational insight: the attacker can find an effective attack policy by relaxing the attack problem (5) so that the relaxed problem can be effectively solved with RL. Concretely, consider the higher-level attack MDP $\mathcal{N} = (\Xi, \Delta, \rho, \tau)$ and the associated optimal control problem:

- The attacker observes the attack state $\xi_t \in \Xi$.
- The attack action space is $\{\delta_t \in \mathbb{R} : |\delta_t| \leq \Delta\}$.
- The original attack loss function $\mathbf{1}[Q_t \notin \mathcal{Q}^\dagger]$ is a 0-1 loss that is hard to optimize. We replace it with a continuous surrogate loss function ρ that measures how close the current agent Q-table Q_t is to the target Q-table set:

$$\rho(\xi_t) = \sum_{s \in S^\dagger} \left[\max_{a \notin \pi^\dagger(s)} Q_t(s,a) - \max_{a \in \pi^\dagger(s)} Q_t(s,a) + \eta \right]_+ \tag{12}$$

  where $\eta > 0$ is a margin parameter to encourage that $\pi^\dagger(s)$ is strictly preferred over $A \backslash \pi^\dagger(s)$ with no ties.
- The attack state transition probability is defined by $\tau(\xi_{t+1} \mid \xi_t, \delta_t)$. Specifically, the new attack state $\xi_{t+1} = (s_{t+1}, a_{t+1}, s_{t+2}, r_{t+1}, Q_{t+1})$ is generated as follows:
  - s_{t+1} is copied from ξ_t if not the end of episode, else $s_{t+1} \sim \mu_0$.
  - a_{t+1} is the RL agent's exploration action drawn according to (2), note it involves Q_{t+1}.
  - s_{t+2} is the RL agent's new state drawn according to the MDP transition probability $P(\cdot \mid s_{t+1}, a_{t+1})$.
  - r_{t+1} is the new (not yet poison) reward according to MDP $R(s_{t+1}, a_{t+1}, s_{t+2})$.
  - The attack δ_t happens. The RL agent updates Q_{t+1} according to (3).

With the higher-level attack MDP $\mathcal{N}$, we relax the optimal attack problem (5) into

$$\phi^* = \arg\min_{\phi} \mathbb{E}_\phi \sum_{t=0}^{\infty} \rho(\xi_t) \tag{13}$$

One can now solve (13) using Deep RL algorithms. In this paper, we choose Twin Delayed DDPG (TD3) (Fujimoto et al., 2018), a state-of-the-art algorithm for continuous action space. We use the same set of hyperparameters for TD3 across all experiments, described in appendix F.

6. Experiments

In this section, We make empirical comparisons between a number of attack policies ϕ: We use the naming convention where the superscript denotes non-adaptive or adaptive

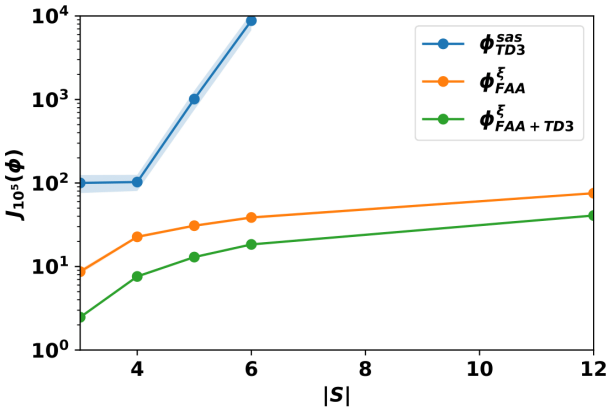


Figure 5. Attack performances on the chain MDPs of different lengths. Each curve shows mean ± 1 standard error over 1000 independent test runs.

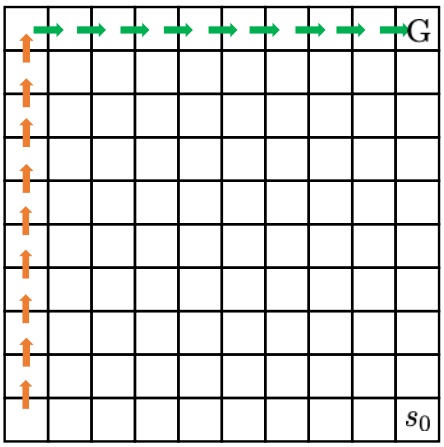


Figure 6. The 10×10 Grid World. s_0 is the starting state and G the terminal goal. Each move has a -0.1 negative reward, and a $+1$ reward for arriving at the goal. We consider two partial target policies: $\pi_1^\dagger$ marked by the green arrows, and $\pi_2^\dagger$ by *both* the green and the orange arrows.

policy: ϕ^{sas} depends on (s_t, a_t, s_{t+1}) but not Q_t. Such policies have been extensively used in the reward shaping literature and prior work (Ma et al., 2019; Huang & Zhu, 2019) on reward poisoning; ϕ^ξ depends on the whole attack state ξ_t. We use the subscript to denote how the policy is constructed. Therefore, ϕ^ξ_{TD3} is the attack policy found by solving (13) with TD3; $\phi^\xi_{FAA+TD3}$ is the attack policy found by TD3 initialized from FAA (Algorithm 3), where TD3 learns to provide an additional δ'_t on top of the δ_t generated by ϕ^ξ_{FAA}, and the agent receives $r_t + \delta_t + \delta'_t$ as reward; ϕ^{sas}_{TD3} is the attack policy found using TD3 with the restriction that the attack policy only takes (s_t, a_t, s_{t+1}) as input.

In all of our experiments, we assume a standard Q-learning RL agent with parameters: $Q_0 = 0^{S\times A}$, $\varepsilon = 0.1, \gamma = 0.9, \alpha_t = 0.9, \forall t$. The plots show ± 1 standard error around each curve (some are difficult to see). We will often evaluate an attack policy ϕ using a Monte Carlo estimate of the 0-1 attack cost $J_T(\phi)$ for $T = 10^5$, which approximates the objective $J_\infty(\phi)$ in (5).

6.1. Efficiency of Attacks across different Δ's

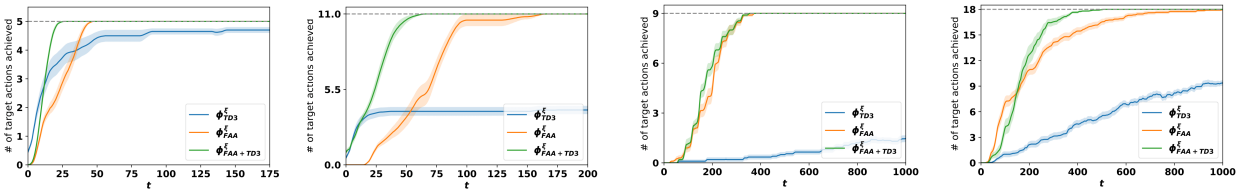
Recall that $\Delta > \Delta_3$, $\Delta > \Delta_4$ are sufficient conditions for manually-designed attack policies $\phi^{sas}_{\Delta_3}$ and ϕ^ξ_{FAA} to be feasible, but they are not necessary conditions. In this experiment, we empirically investigate the feasibilities and efficiency of non-adaptive and adaptive attacks across different Δ values.

We perform the experiments on the chain MDP in Figure 2. Recall that on this example, $\Delta_3 = 0.132$ and $\Delta_4 = 1$ (implicit). We evaluate across 4 different Δ values, $[0.1, 0.2, 0.5, 1]$, covering the range from Δ_3 to Δ_4. The result is shown in Figure 4.

We are able to make several interesting observations:
(1) All attacks are feasible (y-axis $\ll T$), even when Δ falls under the thresholds Δ_3 and Δ_4 for corresponding methods. This suggests that the feasibility thresholds are not tight.
(2) For non-adaptive attacks, as Δ increases the best-found attack policies ϕ^{sas}_{TD3} achieve small improvement, but generally incur a large attack cost.
(3) Adaptive attacks are very efficient when Δ is large. At $\Delta = 1$, the best adaptive attack $\phi^\xi_{FAA+TD3}$ achieves a cost of merely 13 (takes 13 steps to always force $\pi^\dagger$ on the RL agent). However, as Δ decreases the performance quickly degrades. At $\Delta = 0.1$ adaptive attacks are only as good as non-adaptive attacks. This shows an interesting transition region in Δ that our theoretical analysis does not cover.

6.2. Adaptive Attacks are Faster

In this experiment, we empirically verify that, while both are feasible, adaptive attacks indeed have an attack cost $O(\text{Poly}|S|)$ while non-adaptive attacks have $O(e^{|S|})$. The 0-1 costs $1[\pi_t \neq \pi^\dagger]$ are in general incurred at the beginning of each $t = 0 \dots T$ run. In other words, adaptive attacks achieve $\pi^\dagger$ faster than non-adaptive attacks. We use several chain MDPs similar to Figure 2 but with increasing number of states $|S| = 3, 4, 5, 6, 12$. We provide a large enough $\Delta = 2 \gg \Delta_4$ to ensure the feasibility of all attack policies. The result is shown in Figure 5. The best-found non-adaptive attack ϕ^{sas}_{TD3} is approximately straight on the log-scale plot, suggesting attack cost J growing exponentially with MDP size $|S|$. In contrast, the two adaptive attack polices ϕ^ξ_{FAA} and $\phi^\xi_{FAA+TD3}$ actually achieves attack cost linear in $|S|$. This is not easy to see from this log-scaled plot; We reproduce Figure 5 without the log scale in the



(a) 6-state chain MDP (b) 12-state chain MDP (c) 10×10 MDP with $\pi_1^\dagger$. (d) 10×10 MDP with $\pi_2^\dagger$.

Figure 7. Experiment results for the ablation study. Each curve shows mean ± 1 standard error over 20 independent test runs. The gray dashed lines indicate the total number of target actions.

appendix G.1, where the linear rate can be clearly verified. This suggests that the upperbound developed in Theorem 5 and Corollary 6 can be potentially improved.

6.3. Ablation Study

In this experiment, we compare three adaptive attack policies: ϕ^{ξ}_{TD3} the policy found by out-of-the-box TD3, ϕ^{ξ}_{FAA} the manually designed FAA policy, and $\phi^{\xi}_{FAA+TD3}$ the policy found by using FAA as initialization for TD3.

We use three MDPs: a 6-state chain MDP, a 12-state chain MDP, and a 10×10 grid world MDP.. The 10×10 MDP has two separate target policies $\pi_1^\dagger$ and $\pi_2^\dagger$, see Figure 6.

For evaluation, we compute the number of target actions achieved $|\{s \in S^\dagger : \pi_t(s) \in \pi^\dagger(s)\}|$ as a function of t. This allows us to look more closely into the progress made by an attack. The results are shown in Figure 7.

First, observe that across all 4 experiments, attack policy ϕ^{ξ}_{TD3} found by out-of-the-box TD3 never succeeded in achieving all target actions. This indicates that TD3 alone cannot produce an effective attack. We hypothesize that this is due to a lack of effective exploration scheme: when the target states are sparse ($|S^\dagger| \ll |S|$) it can be hard for TD3 equiped with Gaussian exploration noise to locate all target states. As a result, the attack policy found by vanilla TD3 is only able to achieve the target actions on a subset of frequently visited target states.

Hand-crafted ϕ^{ξ}_{FAA} is effective in achieving the target policies, as is guaranteed by our theory. Nevertheless, we found that $\phi^{\xi}_{FAA+TD3}$ always improves upon ϕ^{ξ}_{TD3}. Recall that we use FAA as the initialization and then run TD3. This indicates that TD3 can be highly effective with a good initialization, which effectively serves as the initial exploration policy that allows TD3 to locate all the target states.

Of special interest are the two experiments on the 10×10 Grid World with different target policies. Conceptually, the advantage of the adaptive attack is that the attacker can perform explicit navigation to lure the agent into the target states. An efficient navigation policy that leads the agent to all target states will make the attack very efficient. Observe that in Figure 6, both target polices form a chain, so that if the agent starts at *the beginning of the chain*, the target actions naturally lead the agent to the subsequent target states, achieving efficient navigation.

Recall that the FAA algorithm prioritizes the target states farthest to the starting state. In the 10×10 Grid World, the farthest state is the top-left grid. For target states $S_1^\dagger$, the top-left grid turns out to be the beginning of the *target chain*. As a result, ϕ^{ξ}_{FAA} is already very efficient, and $\phi^{\xi}_{FAA+TD3}$ couldn't achieve much improvement, as shown in 7c. On the other hand, for target states $S_2^\dagger$, the top-left grid is in the middle of the target chain, which makes ϕ^{ξ}_{FAA} not as efficient. In this case, $\phi^{\xi}_{FAA+TD3}$ makes a significant improvement, successfully forcing the target policy in about 500 steps, whereas it takes ϕ^{ξ}_{FAA} as many as 1000 steps, about twice as long as $\phi^{\xi}_{FAA+TD3}$.

7. Conclusion

In this paper, we studied the problem of reward-poisoning attacks against reinforcement-learning agents. Theoretically, we provide robustness certificates that guarantee the truthfulness of the learned policy when the attacker's constraint is stringent. When the constraint is loose, we show that by being adaptive to the agent's internal state, the attacker can force the target policy in polynomial time, whereas a naive non-adaptive attack takes exponential time. Empirically, we formulate that the reward poisoning problem as an optimal control problem on a higher-level attack MDP, and developed computational tools based on DRL that is able to find efficient attack policies across a variety of environments.

Acknowledgments

This work is supported in part by NSF 1545481, 1623605, 1704117, 1836978 and the MADLab AF Center of Excellence FA9550-18-1-0166.

References

Altschuler, J., Brunel, V.-E., and Malek, A. Best arm identification for contaminated bandits. *Journal of Machine Learning Research*, 20(91):1–39, 2019.

Barto, A. G. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pp. 17–47. Springer, 2013.

Behzadan, V. and Munir, A. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp. 262–275. Springer, 2017.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 456–464. ACM, 2019.

Devlin, S. M. and Kudenko, D. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pp. 433–440. IFAAMAS, 2012.

Dhingra, B., Li, L., Li, X., Gao, J., Chen, Y.-N., Ahmed, F., and Deng, L. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*, 2016.

Even-Dar, E. and Mansour, Y. Learning rates for q-learning. *Journal of machine learning Research*, 5(Dec):1–25, 2003.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Huang, Y. and Zhu, Q. Deceptive reinforcement learning under adversarial manipulations on cost signals. *arXiv preprint arXiv:1906.10571*, 2019.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

Jun, K.-S., Li, L., Ma, Y., and Zhu, J. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 3640–3649, 2018.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

Kos, J. and Song, D. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*, 2017.

Lawler, G. F. Expected hitting times for a random walk on a connected graph. *Discrete mathematics*, 61(1):85–92, 1986.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017.

Liu, F. and Shroff, N. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pp. 4042–4050, 2019.

Ma, Y., Jun, K.-S., Li, L., and Zhu, X. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pp. 186–204. Springer, 2018.

Ma, Y., Zhang, X., Sun, W., and Zhu, J. Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, pp. 14543–14553, 2019.

Melo, F. S. Convergence of q-learning: A simple proof.

Neff, G. and Nagy, P. Talking to bots: Symbiotic agency and the case of tay. *International Journal of Communication*, 10:17, 2016.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287, 1999.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Peltola, T., Çelikok, M. M., Daee, P., and Kaski, S. Machine teaching of active sequential learners. In *Advances in Neural Information Processing Systems*, pp. 11202–11213, 2019.

Schmidhuber, J. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

Zhang, H. and Parkes, D. C. Value-based policy teaching with active indirect elicitation. 2008.

Zhang, H., Parkes, D. C., and Chen, Y. Policy teaching through reward function learning. In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 295–304, 2009.

Zhao, X., Xia, L., Zhang, L., Ding, Z., Yin, D., and Tang, J. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 95–103. ACM, 2018.