Decision-Focused Learning of Adversary Behavior in Security Games

Andrew Perrault, Bryan Wilder, Eric Ewing, Aditya Mate, Bistra Dilkina, and Milind Tambe
University of Southern California
Los Angeles, California, United States
{aperraul,bwilder,ericewin,aditya.mate,dilkina,tambe}@usc.edu

ABSTRACT

Stackelberg security games are a critical tool for maximizing the utility of limited defense resources to protect important targets from an intelligent adversary. Motivated by green security, where the defender may only observe an adversary's response to defense on a limited set of targets, we study the problem of defending against the same adversary on a larger set of targets from the same distribution. We give a theoretical justification for why standard two-stage learning approaches, where a model of the adversary is trained for predictive accuracy and then optimized against, may fail to maximize the defender's expected utility in this setting. We develop a decision-focused learning approach, where the adversary behavior model is optimized for *decision* quality, and show empirically that it achieves higher defender expected utility than the two-stage approach when there is limited training data and a large number of target features.

KEYWORDS

Noncooperative games; Single- and multi-agent planning; Machine learning

1 INTRODUCTION

Many real-world settings call for allocating limited defender resources against a strategic adversary, such as protecting public infrastructure [22], transportation networks [20], large public events [28], urban crime [29], and green security [8]. *Stackelberg security games (SSGs)* are a critical framework for computing defender strategies that maximize expected defender utility to protect important targets from an intelligent adversary [22].

In many SSG settings, the adversary's utility function is not known a priori. In domains where there are many interactions with the adversary, the history of interactions can be leveraged to construct an *adversary behavior model*: a mapping from target features to values [14]. An example of such a domain is protecting wildlife from poaching [8]. The adversary's behavior is observable because snares are left behind, which rangers aim to remove (Figure 1). Various features such as animal counts, distance to the edge of the park, weather and time of day may affect how attractive a particular target is to the adversary.

Figure 1: Snares removed by rangers in Srepok National Park, Cambodia.

We focus on the problem of learning adversary models that generalize well: the training data consists of adversary behavior in the context of particular sets of targets, and we wish to achieve a high defender utility in the situation where we are playing against the same adversary and new sets of targets. In problem of poaching prevention, rangers patrol a small portion of the park each day and aim to predict poacher behavior across a large park consisting of targets with novel feature values [10].

The standard approach to this problem [14, 19, 27] breaks the problem into two stages. In the first, the adversary model is fit to the historical data using a standard machine learning loss function, such as mean squared error. In the second, the defender optimizes her allocation of defense resources against the model of adversary behavior learned in the first stage. Extensive research has focused on the first, predictive stage: developing better models of human behavior [1, 6]. We show that models that provide better predictions may not improve the defender's true objective: higher expected utility. This was observed previously by Ford et al. [9] in the context of network security games, motivating our approach.

We propose a decision-focused approach to adversary modeling in SSGs which directly trains the predictive model to maximize defender expected utility on the historical data. Our approach builds on a recently proposed framework (outside of security games) called decision-focused learning, which aims to optimize the quality of the decisions induced by the predictive model, instead of focusing solely on predictive accuracy [23]; Figure 2 illustrates our approach vs. a standard two-stage method. The main idea is to integrate a solver for the defender's equilibrium strategy into the loop of machine learning training and update the model to improve the decisions output by the solver.

While decision-focused learning has recently been explored in other domains (see related work), we overcome two main challenges to extend it to SSGs. First, the defender optimization problem is typically nonconvex, whereas previous work has focused on convex problems. Second, decision-focused learning requires counterfactual data—we need to know what our decision outcome quality would have been, had we taken a different action than the one

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 2019, Montreal, Canada

https://doi.org/doi

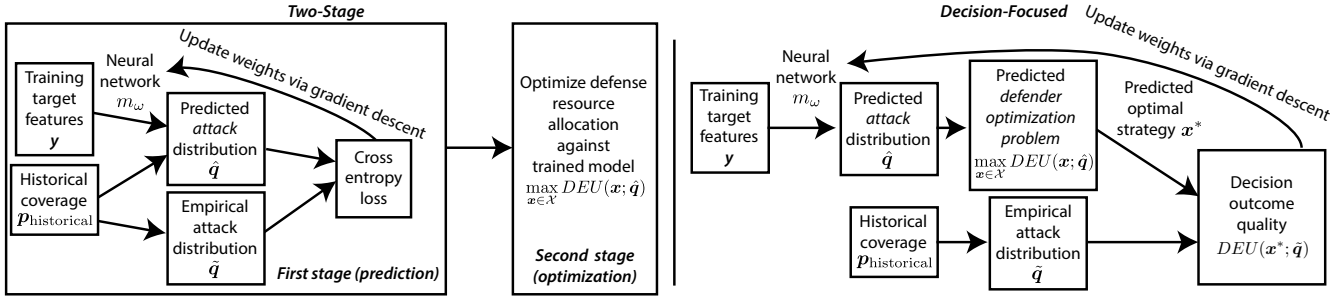

Figure 2: Comparison between a standard two-stage approach to training an adversary model and our decision-focused approach.

observed in training. By contrast, in SSGs we typically only observe the attacker's response to a fixed historical mixed strategy.

In summary, our contributions are:

(1) We provide a theoretical justification for why decision-focused approaches can outperform two-stage approaches in SSGs.
(2) We develop a decision-focused learning approach to adversary modeling in SSGs, showing both how to differentiate through general nonconvex problems as well as estimate counterfactual utilities for subjective utility quantal response [19] and related adversary models.
(3) We test our approach on a combination of synthetic and human subject data and show that decision-focused learning outperforms a two-stage approach in many settings.

The paper is laid out as follows. Section 2 gives background on SSGs, learning in SSGs and introduces decision-focused learning. Section 3 provides a theoretical justification for why decision-focused approaches can outperform two-stage approaches in SSGs. Section 4 describes our technical approach to decision-focused learning in SSGs, shows to perform decision-focused learning under a smooth, nonconvex objective, and presents how we perform counterfactual adversary estimates in SSGs. Section 5 provides experimental results of our approach in simulation and on human subject data.

Related Work. There is a rich literature on SSGs, ranging from information revelation [11, 15] to extensive-form models [5] to patrolling on graphs [3, 4]. Adversary modeling in particular has been a subject of extensive study. Yang et al. [27] show that modeling the adversary with *quantal response (QR)* results in more accurate attack predictions. Nguyen et al. [19] develops *subjective utility quantal response (SUQR)*, which is more accurate than QR. SUQR is the basis of other models such as SHARP [14]. We focus on SUQR in our experiments because it is a relatively simple and widely used approach. Our decision-focused approach extends to other models that decompose the attacker's behavior into the impact of coverage and target value. Sinha et al. [21] and Haghtalab et al. [12] study the sample complexity (i.e., the number of attacks required) of learning an adversary model. Our setting differs from theirs because their defender observes attacks on the same target set that their defense performance is evaluated on. Ling et al. [16, 17] use a differentiable QR equilibrium solver to reconstruct the payoffs of both players from play. This differs from our objective of maximizing the defender's expected utility.

Outside of SSGs, Hartford et al. [13] and Wright and Leyton-Brown [24] study the problem of predicting play in unseen games assuming that all payoffs are fully observable; in our case, the defender seeks to maximize expected utility and does not observe the attacker's payoffs. Hartford et al. [13] is the only other work to apply deep learning to modeling boundedly rational players in games.

Wilder et al. [23] and Donti et al. [7] study decision-focused learning for discrete and convex optimization, respectively. Donti et al. use sequential quadratic programming to solve a convex nonquadratic objective and use the last program to calculate derivatives. Here we propose an approach that works for the broader family of nonconvex functions.

2 SETTING

We begin by providing a brief background on SSGs.

2.1 Stackelberg Security Games (SSGs)

Our focus is on optimizing defender strategies for SSGs, which describe the problem of protecting a set of targets given limited defense resources and constraints on how the resources may be deployed [22]. Formally, an SSG is a tuple $\{\mathcal{T}, \boldsymbol{u}_d, \boldsymbol{u}_a, C_d\}$, where $\mathcal{T}$ is a set of targets, $\boldsymbol{u}_d \leq 0$ is the defender's payoff if each target is successfully attacked, $\boldsymbol{u}_a \geq 0$ is the attacker's, and C_d is the set of constraints the defender's strategy must satisfy. Both players receive a payoff of zero when the attacker attacks a target that is defended.

The game proceeds in two stages: the defender computes a mixed strategy that satisfies the constraints C_d, which induces a *marginal coverage probability (or coverage)* $\boldsymbol{p} = \{\boldsymbol{p}_i : i \in \mathcal{T}\}$. The attacker's *attack function* $\boldsymbol{q}$ determines which target is attacked, inducing an *attack probability* for each target. The defender seeks to maximize her expected utility:

$$\max_{\boldsymbol{p} \text{ satisfying } C_d} DEU(\boldsymbol{p}; \boldsymbol{q}) = \max_{\boldsymbol{p} \text{ satisfying } C_d} \sum_{i \in \mathcal{T}} (1 - \boldsymbol{p}_i) \boldsymbol{q}_i(\boldsymbol{u}_a, \boldsymbol{p}) \boldsymbol{u}_d(i). \quad (1)$$

The attacker's q function can represent a rational attacker, e.g., $\boldsymbol{q}_i(\boldsymbol{p}, \boldsymbol{u}_a) = 1$ if $i = \text{argmax}_{j \in \mathcal{T}} (1 - \boldsymbol{p}_j) \boldsymbol{u}_a(j)$ else 0, or a boundedly

rational attacker. A QR attacker [18] attacks each target with probability proportional to the exponential of its payoff scaled by a constant λ, i.e., $\boldsymbol{q}_i(\boldsymbol{p}) \propto \exp(\lambda(1-\boldsymbol{p}_i)\boldsymbol{u}_a)$. An SUQR [19] attacker attacks each target with probability proportional to the exponential of an *attractiveness function*:

$$\boldsymbol{q}_i(\boldsymbol{p}, \boldsymbol{y}) \propto \exp(w\boldsymbol{p}_i + \phi(\boldsymbol{y}_i)), \tag{2}$$

where $\boldsymbol{y}_i$ is a vector of features of target i and $w < 0$ is a constant. We call ϕ the *target value function*.

2.2 Learning in SSGs

We consider the problem of learning to play against an attacker with an unknown attack function $\boldsymbol{q}$. We observe attacks made by the adversary against sets of targets with differing features, and our goal is to generalize to new sets of targets with unseen feature values.

Formally, let $\langle \boldsymbol{q}, C_d, D_{\text{train}}, D_{\text{test}} \rangle$ be an instance of a *Stackelberg security game with latent attack function (SSG-LA)*. $\boldsymbol{q}$, which is not observed by the defender, is the true mapping from the features and coverage of each target to the probability that the attacker will attack that target. C_d is the set of constraints that a mixed strategy defense must satisfy for the defender. D_{train} are *training games* of the form $\langle \mathcal{T}, \boldsymbol{y}, \mathcal{A}, \boldsymbol{u}_d, \boldsymbol{p}_{\text{historical}} \rangle$, where $\mathcal{T}$ is the set of targets, and $\boldsymbol{y}$, $\mathcal{A}$, $\boldsymbol{u}_d$ and $\boldsymbol{p}_{\text{historical}}$ are the features, observed attacks, defender's utility function, and historical coverage probabilities, respectively, for each target $i \in \mathcal{T}$. D_{test} are *test games* $\langle \mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d \rangle$, each containing a set of targets and the associated features and defender values for each target. We assume that all games are drawn i.i.d. from the same distribution. In a green security setting, the training games represent the results of patrols on limited areas of the park and the test games represent the entire park.

The defender's goal is to select a coverage function $\boldsymbol{x}$ that takes the parameters of each test game as input and maximizes her expected utility across the test games against the attacker's true $\boldsymbol{q}$:

$$\max_{\boldsymbol{x} \text{ satisfying } C_d} \mathop{\mathbb{E}}_{\langle \mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d \rangle \sim D_{\text{test}}} \left[DEU(\boldsymbol{x}(\mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d); \boldsymbol{q}) \right]. \tag{3}$$

To achieve this, she can observe the attacker's behavior in the training data and learn how he values different combinations of features. We now explore two approaches to the learning problem: the standard two-stage approach taken by previous work and our proposed decision-focused approach.

2.3 Two-Stage Approach

A standard two-stage approach to the defender's problem is to estimate the attacker's $\boldsymbol{q}$ function from the training data and optimize against the estimate during testing. This process resembles multiclass classification where the targets are the classes: the inputs are the target features and historical coverages, and the output is a distribution over the predicted attack. Specifically, the defender fits a function $\hat{\boldsymbol{q}}$ to the training data that minimizes a loss function. Using the cross entropy, the loss for a particular training example is

$$\mathcal{L}(\hat{q}(\boldsymbol{y}, \boldsymbol{p}_{\text{historical}}), \mathcal{A}) = -\sum_{i \in T} \tilde{\boldsymbol{q}} \log(\hat{\boldsymbol{q}}_i(\boldsymbol{y}, \boldsymbol{p}_{\text{historical}})), \tag{4}$$

where $\tilde{\boldsymbol{q}} = \frac{\mathcal{A}_i}{|\mathcal{A}|}$ is the *empirical attack distribution* and $\mathcal{A}_i$ is the number of historical attacks that were observed on target i. Note that we use hats to indicate model outputs and tildes to indicate the ground truth. For each test game $\langle \mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d \rangle$, coverage is selected by maximizing the defender's expected utility assuming the attack function is $\hat{\boldsymbol{q}}$:

$$\max_{\boldsymbol{x} \text{ satisfying } C_d} DEU(\boldsymbol{x}(\mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d); \hat{\boldsymbol{q}}). \tag{5}$$

2.4 Decision-Focused Learning

The standard approach may fall short when the loss function (e.g., cross entropy) does not align with the true goal of maximizing expected utility. Ultimately, the defender just wants $\hat{\boldsymbol{q}}$ to induce the correct mixed strategy, regardless of how accurate it is in a general sense. The idea behind our decision-focused learning approach is to directly train $\hat{\boldsymbol{q}}$ to maximize defender utility. Define

$$\boldsymbol{x}^*(\hat{\boldsymbol{q}}) = \mathop{\arg\max}_{\boldsymbol{x} \text{ satisfying } C_d} DEU(\boldsymbol{x}; \hat{\boldsymbol{q}}) \tag{6}$$

to be the optimal defender coverage function given attack function $\hat{\boldsymbol{q}}$. Ideally, we would find a $\hat{\boldsymbol{q}}$ which maximizes

$$DEU(\hat{\boldsymbol{q}}) = \mathop{\mathbb{E}}_{\langle \mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d \rangle \sim D_{\text{test}}} \left[DEU(\boldsymbol{x}^*(\hat{\boldsymbol{q}}); \boldsymbol{q}) \right]. \tag{7}$$

This is just the defender's expected utility on the test games when she plans her mixed strategy defense based on attack function $\hat{\boldsymbol{q}}$ but the true function is $\boldsymbol{q}$. While we do not have access to D_{test}, we can estimate Eq. 7 using samples from D_{train} (taking the usual precaution of controlling model complexity to avoid overfitting). The idea behind decision-focused learning is to directly optimize Eq. 7 on the training data instead of using an intermediate loss function such as cross entropy. Minimizing Eq. 7 on the training set via gradient descent requires the gradient, which we can derive using the chain rule:

$$\frac{\partial DEU(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}} = \mathop{\mathbb{E}}_{\langle \mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d \rangle \sim D_{\text{train}}} \left[\frac{\partial DEU(\boldsymbol{x}^*(\hat{\boldsymbol{q}}); \boldsymbol{q})}{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})} \frac{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}} \right].$$

Here, $\frac{\partial DEU(\boldsymbol{x}^*(\hat{\boldsymbol{q}}); \boldsymbol{q})}{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}$ describes how the defender's true utility with respect to $\boldsymbol{q}$ changes as a function of her strategy $\boldsymbol{x}^*$. $\frac{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}}$ describes how $\boldsymbol{x}^*$ depends on the estimated attack function $\hat{\boldsymbol{q}}$, which requires differentiating through the optimization problem in Eq. 6. Suppose that we have a means to calculate both terms. Then we can estimate $\frac{\partial DEU(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}}$ by sampling example games from D_{train} and computing gradients on the samples. If $\hat{\boldsymbol{q}}$ is itself implemented in a differentiable manner (e.g., a neural network), this allows us to train the entire system end-to-end via gradient descent. Previous work has explored decision-focused learning in other contexts [7, 23], but SSGs pose unique challenges that complicate the process of computing both of the required terms above. In Section 4, we explore these challenges and propose solutions.

3 IMPACT OF TWO-STAGE LEARNING ON DEU

We demonstrate that, for natural two-stage training loss functions, decreasing the loss may not lead to increasing the *DEU*. This indicates that we may be able to improve decision quality by making

use of decision-focused learning because a decision-focused approach uses the decision objective as the loss. Thus, reducing the loss function increases the *DEU* in decision-focused learning.

We begin with a simple case: two-target games with a rational attacker and zero-sum utilities.

Theorem 3.1. *Consider a two-target SSG with a rational attacker, zero-sum utilities, and a single defense resource to allocate, which is not subject to scheduling constraints (i.e., any nonnegative marginal coverage that sums to one is feasible). Let $z_0 \geq z_1$ be the attacker's values for the targets, which are observed by the attacker, but not the defender, and we assume w.l.o.g. are non-negative and sum to 1.*

The defender has an estimate of the attacker's values $(\hat{z}_0, \hat{z}_1)$ with mean squared error (MSE) ϵ^2. *Suppose the defender optimizes coverage against this estimate. If $\epsilon^2 \leq (1-z_0)^2$, the ratio between the highest DEU under the estimate of $(\hat{z}_0, \hat{z}_1)$ with MSE ϵ^2 and the lowest DEU is:*

$$\frac{(1-(z_0-\epsilon))z_0}{(1-(z_1-\epsilon))z_1}. \tag{8}$$

Proof. Given the condition that $\epsilon^2 \leq (1-z_0)^2$, there are two configurations of $\hat{z}$ that have mean squared error ϵ^2: $\hat{z}_0 = z_0 \pm \epsilon, \hat{z}_1 = z_1 \mp \epsilon$. Using Lemma 3.2, the configurations have defender utility $-(1-(z_1-\epsilon))z_1$ and $(1-(z_0-\epsilon))z_0$, respectively, because the attacker always attacks the target with underestimated value. The condition on ϵ^2 is required to make both estimates feasible. Because $z_0 \geq z_1$, $-(1-(z_0-\epsilon))z_0 \leq -(1-(z_1-\epsilon))z_1$. □

Lemma 3.2. *Consider a two-target, zero-sum SSG with a rational attacker, and a single defense resource, which is not subject to scheduling constraints. The optimal defender coverage is $x_0 = z_0$ and $x_1 = z_1$, and the defender's payoff under this coverage is $-(1-z_0)z_0 = -(1-z_1)z_1$.*

Proof. The defender's maximum payoff is achieved when the expected value for attacking each target is equal, and we require that $x_0 + x_1 \leq 1$ for feasibility. With $x_0 = z_0$ and $x_1 = z_1$, the attacker's payoff is $(1-z_0)z_0$ if he attacks target 0 and $(1-z_1)z_1 = (1-(1-z_0))(1-z_0) = z_0(1-z_0)$ if he attacks target 1. □

The reason for the gap in defender expected utilities is that the attacker attacks the target with value that is underestimated by $(\hat{z}_0, \hat{z}_1)$. This target has less coverage than it would have if the defender knew the attacker's utilities precisely, allowing the attacker to benefit. When the defender reduces the coverage on the larger value target, the attacker benefits more, causing the gap in expected defender utilities.

Note that because (8) is at least one (since *DEU* are negative), decreasing the MSE does not necessarily lead to higher *DEU*. For $\epsilon > \epsilon'$, the learned model at MSE=ϵ^2 will have higher DEU than the model at MSE=$(\epsilon')^2$ if the former underestimates the value of z_1, the latter underestimates the value of z_0 and ϵ, and ϵ' are sufficiently close. In decision-focused learning, the *DEU* is used as the loss directly—thus, a model with lower loss must have higher *DEU*.

Figure 3 shows the *DEU* of the highest and lowest *DEU* estimates of z as z_1 is varied at two different loss levels: $\epsilon^2 = 0.01$ and $\epsilon^2 = 0.02$. A larger gap in target values results in a larger impact on decision quality, and the largest gap occurs when $z_1 \to 0$.

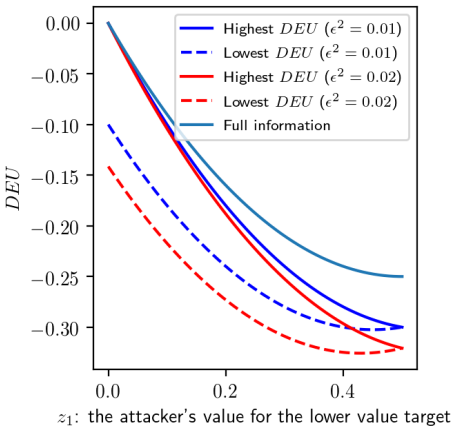

Figure 3: *DEU* as z_1 and ϵ^2 vary. A larger gap in target values increases the difference in *DEU* between the highest and lowest *DEU* at a particular loss level ϵ^2.

In the case of Theorem 3.1, the defender can lose value $z_0\epsilon$, or ϵ as $z_0 \to 1$, compared to the optimum because of an unfavorable distribution of estimation error. We show that this carries over to a boundedly rational QR attacker, with the degree of loss converging towards the rational case as λ increases.

Theorem 3.3. *Consider the setting of Theorem. 3.1, but in the case of a QR attacker. For any $0 \leq \alpha \leq 1$, if $\lambda \geq \frac{2}{(1-\alpha)\epsilon} \log \frac{1}{(1-\alpha)\epsilon}$, the defender's loss compared to the optimum may be as much as $\alpha(1-\epsilon)\epsilon$ under a target value estimate with MSE ϵ^2.*

Proof. Let $f(p)$ denote the defender's utility with coverage probability p against a perfectly rational attacker and $g(p)$ denote their utility against a QR attacker. Suppose that we have a bound

$$g(p) - f(p) \leq \delta$$

for some value δ. Let p^* be the optimal coverage probability under perfect rationality. Note that for an alternate probability $p' > p^*$

$$\begin{aligned} g(p') &\leq f(p') + \delta \\ &= f(p^*) - (p' - p^*)\epsilon + \delta \\ &\leq g(p^*) - (p' - p^*)\epsilon + \delta \quad \text{(since } f(p) \leq g(p) \text{ holds for all } p\text{)} \end{aligned}$$

and so any $p' > p^* + \frac{\delta}{\epsilon}$ is guaranteed to have $g(p') < g(p^*)$, implying that the defender must have $p' \leq p^* + \frac{\delta}{\epsilon}$ in the optimal QR solution.

We now turn to estimating how large λ must be in order to get a sufficiently small δ. Let q be the probability that the attacker chooses the first target under QR. Note that we have $f(p) = \epsilon p$ and $g(p) = (1-p)(1-\epsilon)q + p\epsilon(1-q)$. We have

$$\begin{aligned} g(p) - f(p) &= (1-p)(1-\epsilon)q + p\epsilon(1-q) - \epsilon p \\ &= [(1-p)(1-\epsilon) - p\epsilon]q \\ &\leq q \end{aligned}$$

For two targets with value 1 and ϵ, q is given by

$$\frac{e^{\lambda(1-\epsilon)(1-p)}}{e^{\lambda\epsilon p} + e^{\lambda(1-\epsilon)(1-p)}} = \frac{1}{1 + e^{\lambda[\epsilon p - (1-\epsilon)(1-p)]}}$$

Provided that $\lambda \geq \frac{1}{\epsilon p-(1-\epsilon)(1-p)} \log \frac{1}{\delta} = \frac{1}{p-(1-\epsilon)} \log \frac{1}{\delta}$, we will have $g(p) - f(p) \leq \delta$. Suppose that we would like this bound to hold over all $p \geq 1 - \alpha\epsilon$ for some $0 < \alpha < 1$. Then, $p - (1 - \epsilon) \geq (1 - \alpha)\epsilon$ and so $\lambda \geq \frac{1}{(1-\alpha)\epsilon} \log \frac{1}{\delta}$ suffices. Now if we take $\delta \leq (1 - \alpha)\epsilon^2$, we have that for $\lambda \geq \frac{2}{(1-\alpha)\epsilon} \log \frac{1}{(1-\alpha)\epsilon}$, the QR optimal strategy p' must satisfy $p' \leq 1 - \alpha\epsilon$, implying that the defender allocates at least $\alpha\epsilon$ coverage to the target with true value 0. Suppose the attacker chooses the target with value 1 with probability q^*. Then, the defender's loss compared to the optimum is $q^*\alpha\epsilon$. By a similar argument as above, it is easy to verify that under our stated conditions on λ, and assuming $\alpha \geq \frac{1}{2}$, we have $q^* \geq (1 - \epsilon)$, for total defender loss $(1 - \epsilon)\alpha\epsilon$. □

4 DECISION-FOCUSED LEARNING IN SSGS WITH AN SUQR ADVERSARY

We now present our technical approach to decision-focused learning in SSGs. As discussed above, we use $DEU(\hat{\boldsymbol{q}})$, the expected utility induced by an estimate $\hat{\boldsymbol{q}}$, as the objective for training. The key idea is to embed the defender optimization problem into training and compute gradients of DEU with respect to the model's predictions. In order to do so, we need two quantities, each of which poses a unique challenge in the context of SSGs.

First, we need $\frac{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}}$, which describes how the defender's strategy $\boldsymbol{x}^*$ depends on $\hat{\boldsymbol{q}}$. Computing this requires differentiating through the defender's optimization problem. Previous work on differentiable optimization considers convex problems [2]. However, typical bounded rationality models for $\hat{\boldsymbol{q}}$ (e.g., QR, SUQR, and SHARP) all induce *nonconvex* defender problems. We resolve this challenge by showing how to differentiate through the local optimum output by a black-box nonconvex solver.

Second, we need $\frac{\partial DEU(\boldsymbol{x}^*(\hat{\boldsymbol{q}}):\boldsymbol{q})}{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}$, which describes how the defender's *true* utility with respect to $\boldsymbol{q}$ depends on her strategy $\boldsymbol{x}^*$. Computing this term requires a *counterfactual* estimate of how the attacker would react to a different coverage vector than the historical one. Unfortunately, typical datasets only contain a set of sampled attacker responses to a particular historical defender mixed strategy. Previous work on decision-focused learning in other domains [7, 23] assumes that the historical data specifies the utility of *any* possible decision, but this assumption breaks down under the limited data available in SSGs. We show that common models like SUQR exhibit a crucial decomposition property that enables unbiased counterfactual estimates. We now explain both steps in more detail.

4.1 Decision-Focused Learning for Nonconvex Optimization

Under nonconvexity, all that we can (in general) hope for is a local optimum. Since there may be many local optima, it is unclear what it means to differentiate through the solution to the problem. We assume that we have black-box access to a nonconvex solver which outputs a fixed local optimum. We show that we can obtain derivatives of that particular optimum by differentiating through a convex quadratic approximation around the solver's output (since existing techniques apply to the quadratic approximation).

We prove that this procedure works for a wide range of nonconvex problems. Specifically, we consider the generic problem $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \theta)$ where f is a (potentially nonconvex) objective which depends on a learned parameter θ. $\mathcal{X}$ is a feasible set that is representable as $\{x : g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x}) \leq 0, h_1(\boldsymbol{x}), \ldots, h_\ell(\boldsymbol{x}) = 0\}$ for some convex functions $g_1, \ldots, g_m$ and affine functions $h_1, \ldots, h_\ell$. We assume there exists some $\boldsymbol{x} \in \mathcal{X}$ with $\boldsymbol{g}(\boldsymbol{x}) < 0$, where $\boldsymbol{g}$ is the vector of constraints. In SSGs, f is the defender objective DEU, θ is the attack function $\hat{\boldsymbol{q}}$, and $\mathcal{X}$ is the set of $\boldsymbol{x}$ satisfying C_d. We assume that f is twice continuously differentiable. These two assumptions capture smooth nonconvex problems over a nondegenerate convex feasible set.

Suppose that we can obtain a local optimum of f. Formally, we say that $\boldsymbol{x}$ is a *strict local minimizer* of f if (1) there exist $\boldsymbol{\mu} \in R^m_+$ and $\boldsymbol{\nu} \in R^\ell$ such that $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \theta) + \boldsymbol{\mu}^\top \nabla \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\nu}^\top \nabla \boldsymbol{h}(\boldsymbol{x}) = 0$ and $\boldsymbol{\mu} \odot \boldsymbol{g}(\boldsymbol{x}) = 0$ and (2) $\nabla^2 f(\boldsymbol{x}, \theta) \prec 0$. Intuitively, the first condition is first-order stationarity, where $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are dual multipliers for the constraints, while the second condition says that the objective is strictly convex at $\boldsymbol{x}$ (i.e., we have a strict local minimum, not a plateau or saddle point). We prove the following:

Theorem 4.1. *Let $\boldsymbol{x}$ be a strict local minimizer of f over $\mathcal{X}$. Then, except on a measure zero set, there exists a convex set $\mathcal{I}$ around $\boldsymbol{x}$ such that $\boldsymbol{x}^*_{\mathcal{I}}(\theta) = \arg\min_{x \in \mathcal{I} \cap \mathcal{X}} f(\boldsymbol{x}, \theta)$ is differentiable. The gradients of $\boldsymbol{x}^*_{\mathcal{I}}(\theta)$ with respect to θ are given by the gradients of solutions to the local quadratic approximation $\min_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{2}\boldsymbol{x}^\top \nabla^2 f(\boldsymbol{x}, \theta)\boldsymbol{x} + \boldsymbol{x}^\top \nabla f(\boldsymbol{x}, \theta)$.*

Proof. By continuity, there exists an open ball around x on which $\nabla^2 f(x, \theta)$ is negative definite; let $\mathcal{I}$ be this ball. Restricted to $\mathcal{X} \cap \mathcal{I}$, the optimization problem is convex, and satisfies Slater's condition by our assumption on $\mathcal{X}$ combined with Lemma 4.2. Therefore, the KKT conditions are a necessary and sufficient description of $x^*_{\mathcal{I}}(\theta)$. Since the KKT conditions depend only on second-order information, $x^*_{\mathcal{I}}(\theta)$ is differentiable whenever the quadratic approximation is differentiable. Note that in the quadratic approximation, we can drop the requirement that $x \in \mathcal{I}$ since the minmizer over $x \in \mathcal{X}$ already lies in $\mathcal{I}$ by continuity. Using Theorem 1 of Amos and Kolter (2017), the quadratic approximation is differentiable except at a measure zero set, proving the theorem. □

Lemma 4.2. *Let $g_1 \ldots g_m$ be convex functions and consider the set $\mathcal{X} = \{x : \boldsymbol{g}(x) \leq 0\}$. If there is a point x^* which satisfies $\boldsymbol{g}(x) < 0$, then for any point $x' \in \mathcal{X}$, the set $\mathcal{X} \cap B(x', \delta)$ contains a point x_{int} satisfying $g(x) < x_{int}$ and $d(x_{int}, x') < \delta$.*

Proof. By convexity, for any $t \in [0, 1]$, the point $(1 - t)x^* + tx'$ lies in $\mathcal{X}$, and for $t < 1$, satisfies $g((1 - t)x^* + tx') < 0$. Moreoever, for t sufficiently large (but strictly less than 1), we must have $d((1 - t)x^* + tx', x') < \delta$, proving the existence of x_{int}. □

This states that the local minimizer within the region output by the nonconvex solver varies smoothly with θ, and we can obtain gradients of it by applying existing techniques [2] to the local quadratic approximation. It is easy to verify that the defender utility maximization problem for an SUQR attacker satisfies the assumptions of Theorem 4.1 since the objective is smooth and typical constraint sets for SSGs are polytopes with nonempty interior (see [25] for a

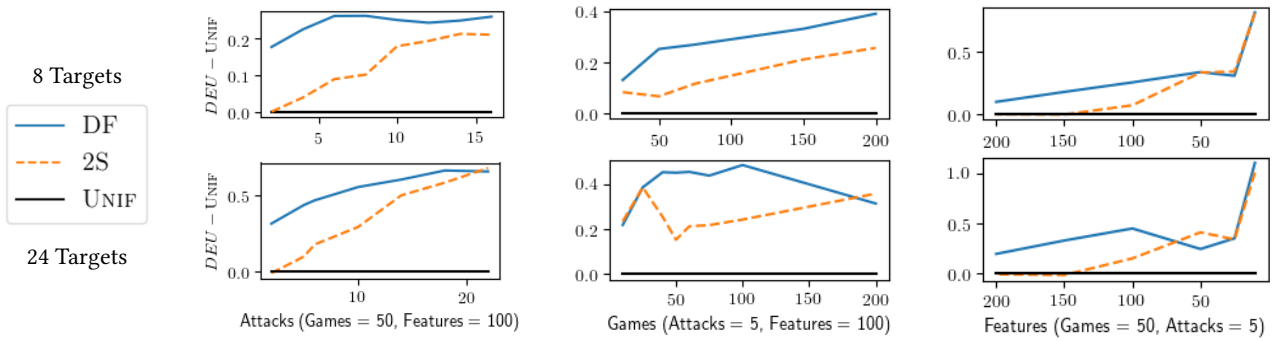

Figure 4: *DEU* − UNIF across the three strategies as we vary the number of features, number of training games and number of observed attacks per training game. When not varied, the parameter values are 100 features, 50 training games and 5 attacks per game. DF receives higher *DEU* than 2S for most parameter values.

list of examples). In fact, our approach is quite general and applies to a range of behavioral models such as QR, SUQR, and SHARP since the defender optimization problem remains smooth in all.

4.2 Counterfactual Adversary Estimates

We now turn to the second challenge, that of estimating how well a different strategy would perform on the historical games. We focus here on the SUQR attacker, but the main idea extends more widely (as we discuss below). For SUQR, if the historical attractiveness values $\phi(y_i)$ were known, then $\frac{\partial DEU}{\partial \boldsymbol{x}^*}$ could be easily computed in closed form using Eq. 2. The difficulty is that we typically only observe samples from the attack distribution $\boldsymbol{q}$, where for SUQR, $\boldsymbol{q}_i \propto \exp(w\boldsymbol{p}_i + \phi(\boldsymbol{y}_i))$. $\phi(y_i)$ itself is not observed directly.

The crucial property enabling counterfactual estimates is that the attacker's behavior can be decomposed into his reaction to the defender's coverage ($w\boldsymbol{p}_i$) and the impact of target values ($\phi(y_i)$). Suppose that we know w and observe sampled attacks for a particular historical game. Because we can estimate $\boldsymbol{q}_i$ and the term $w\boldsymbol{p}_i$ is known, we can invert the exp function to obtain an estimate of $\phi(\boldsymbol{y}_i)$ (formally, this corresponds to the maximum likelihood estimator under the empirical attack distribution). Note that we do not know the *entire function* ϕ, only its value at $\boldsymbol{y}_i$, and that the inversion yields $\phi(\boldsymbol{y}_i)$ that is unique up to a constant additive factor. Having recovered $\phi(\boldsymbol{y}_i)$, we can then perform complete counterfactual reasoning for the defender on the historical games.

5 EXPERIMENTS

We compare the performance of decision-focused and two-stage approaches across a range of settings both simulated and real (using data from Nguyen et al. [19]). We find that decision-focused learning outperforms two-stage when the number of training games is low, the number of attacks observed on each training game is low, and the number of target features is high. We compare the following three defender strategies:

(1) *Decision-focused (DF)* is our decision-focused approach. For the prediction neural network, we use a single layer with ReLU activations with 200 hidden units on synthetic data and 10 hidden units on the simpler human subject data. We do not tune DF.
(2) *Two-stage (2S)* is a standard two-stage approach, where a neural network is fit to predict attacks, minimizing cross-entropy on the training data, using the same architecture as DF. We find that two-stage is sensitive to overfitting, and thus, we use Dropout and early stopping based on a validation set.
(3) *Uniform attacker values (UNIF)* is a baseline where the defender assumes that the attacker's value for all targets is equal and maximizes *DEU* under that assumption.

5.1 Experiments in Simulation

We perform experiments against an attacker with an SUQR target attractiveness function. Raw features values are sampled i.i.d. from the uniform distribution over [-10, 10]. Because it is necessary that the attacker target value function is a function of the features, we sample the attacker and defender target value functions by generating a random neural network for the attacker and defender. Our other parameter settings are chosen to align with Nguyen et al.'s [19] human subject data. We rescale defender values to be between -10 and 0.

We choose instance parameters to illustrate the differences in performance between decision-focused and two-stage approaches. We run 28 trials per parameter combination. Unless it is varied in an experiment, the parameters are:

(1) *Number of targets* = $|\mathcal{T}| \in \{8, 24\}$.
(2) *Features per target* = $|\boldsymbol{y}|/|\mathcal{T}| = 100$.
(3) *Number of training games* = $|D_{\text{train}}| = 50$. We fix the number of test games = $|D_{\text{test}}| = 50$.
(4) *Number of attacks per training game* = $|\mathcal{A}| = 5$.
(5) *Defender resources* is the number of defense resources available. We use 3 for 8 targets and 9 for 24.
(6) We fix the attacker's weight on defender coverage to be $w = -4$ (see Eq. 2), a value chosen because of its resemblance to observed attacker w in human subject experiments [19, 26]. All strategies receive access to this value, which would require the defender to vary her mixed strategies to learn.
(7) *Historical coverage* = $\boldsymbol{p}_{\text{historical}}$ is the coverage generated by UNIF, which is fixed for each training game.

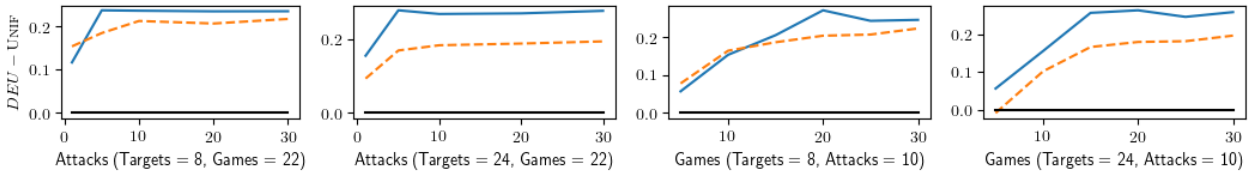

Figure 5: *DEU* − UNIF from human subject data for 8 and 24 targets, as the number of attacks per training game is varied and number of training games is varied. DF receives higher *DEU* for most settings, especially for 24-target games.

Results (Simulations). Figure 4 shows the results of the experiments in simulation, comparing DF and 2S across a variety of problem types. DF yields higher DEU than 2S across most tested parameter settings and DF especially excels in problems where learning is more difficult: more features, fewer training games and fewer attacks. The vertical axis of each graph is median *DEU* minus the *DEU* achieved by UNIF. Because UNIF does not perform learning, its *DEU* is unaffected by the horizontal axis parameter variation, which only affects the difficulty of the learning problem, not the difficulty of the game. The average $DEU(\text{UNIF}) = -2.5$ for 8 targets and $DEU(\text{UNIF}) = -4.2$ for 24.

The left column of Figure 4 compares DF to 2S as the number of attacks observed per game increases. For both 8 and 24 targets, DF receives higher *DEU* than 2S across the tested range. 2S fails to outperform UNIF at 2 attacks per target, whereas DF receives 75% of the *DEU* it receives at 15 attacks per target.

The center column of Figure 4 compares *DEU* as the number of training games increases. Note that without training games, no learning is possible and $DEU(\text{2S}) = DEU(\text{DF}) = DEU(\text{UNIF})$. DF receives equal or higher *DEU* than 2S, except for 24 targets and 200 training games.

The right column of Figure 4 compares *DEU* as the number of features decreases. A larger number of features results in a harder learning problem, as each feature increases the complexity of the attacker's value function. Of the the parameters we vary, features has the largest impact on the relative performance of DF and 2S. DF performs better than 2S for more than 50 features (for 8 targets) and 100 features (for 24 targets). For more than 150 features, 2S fails to learn for both 8 and 24 targets and performs extremely poorly.

5.2 Experiments on Human Subject Data

We use data from human subject experiments performed by Nguyen et al. [19]. The data consists of an 8-target setting with 3 defender resources and a 24-target setting with 9. Each setting has 44 games. Historical coverage is the optimal coverage assuming a QR attacker with $\lambda = 1$. For each game, 30-45 attacks by human subjects are recorded.

We use the attacker coverage parameter w calculated by Nguyen et al. [19]: -8.23. We use maximum likelihood estimation to calculate the ground truth target values for the test games. There are four features for each target: attacker's reward and defender's penalty for a successful attack, attacker's penalty and defender's reward for a failed attack. Note that to be consistent with the rest of the paper, we assume the defender receives a reward of 0 if she successfully prevents an attack.

Results (Human Subject Data). We find that DF receives higher *DEU* than 2S on the human subject data. Figure 5 summarizes our results as the number of training attacks per target and games are varied. Varying the number of attacks, for 8 targets, DF achieves its highest percentage improvement in *DEU* at 5 attacks where it receives 28% more than 2S. For 24 targets, DF achieves its largest improvement of 66% more *DEU* than 2S at 1 attack.

Varying the number of games, DF outperforms 2S except for fewer than 10 training games in the 8-target case. The percentage advantage is greatest for 8-target games at 20 training games (33%) and at 2 training games for 24-target games, where 2S barely outperforms UNIF.

The theorems of Section 3 suggest that models with higher *DEU* may not have higher predictive accuracy. We find that, indeed, this can occur. The effect is most pronounced in the human subject experiments, where 2S has lower test cross entropy than DF by 2–20%. Note that we measure test cross entropy against the attacks generated by UNIF, the same defender strategy used to generate the training data and that 2S received extensive hyperparameter to improve validation cross entropy and DF did not.

6 CONCLUSION

We present a decision-focused approach to adversary modeling in security games. We provide a theoretical justification as to why training an attacker model to maximize *DEU* can provide higher *DEU* than training the model to maximize predictive accuracy. We extend past work in decision-focused learning to smooth nonconvex objectives, accounting for the defender's optimization in SSGs against many attacker types, including SUQR. We show empirically, in both synthetic and human subject data, that our decision-focused approach outperforms standard two stage approaches.

We conclude that improving predictive accuracy does not guarantee increased *DEU* in SSGs. We believe this conclusion has important consequences for future research and that our decision-focused approach can be extended to a variety of SSG models where smooth nonconvex objectives and polytope feasible regions are common.

REFERENCES

[1] Yasaman Dehghani Abbasi, Noam Ben-Asher, Cleotilde Gonzalez, Don Morrison, Nicole Sintov, and Milind Tambe. 2016. Adversaries Wising Up: Modeling Heterogeneity and Dynamics of Behavior. In *Proceedings of the 14th International Conference on Cognitive Modeling*. University Park, PA, 79–85.
[2] Brandon Amos and J. Zico Kolter. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the Thirty-fourth International Conference on Machine Learning (ICML-17)*. Sydney, 136–165.
[3] Nicola Basilico, Giuseppe De Nittis, and Nicola Gatti. 2017. Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence* 246 (2017), 220–257.
[4] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. 2012. Patrolling security games: Definition and algorithms for solving large instances with single patroller

and single intruder. *Artificial Intelligence* 184 (2012), 78–123.

[5] Jiri Cermak, Branislav Bosansky, Karel Durkota, Viliam Lisy, and Christopher Kiekintveld. 2016. Using correlated strategies for computing Stackelberg equilibria in extensive-form games. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*. Phoenix, 439–445.

[6] Jinshu Cui and Richard S John. 2014. Empirical comparisons of descriptive multi-objective adversary models in Stackelberg security games. In *Proceedings of the Fifth Conference on Decision and Game Theory for Security (GameSec-14)*. Los Angeles, 309–318.

[7] Priya Donti, Brandon Amos, and J. Zico Kolter. 2017. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems 30 (NIPS-17)*. Long Beach, 5484–5494.

[8] Fei Fang, Peter Stone, and Milind Tambe. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI-15)*. Buenos Aires, 2589–2595.

[9] Benjamin Ford, Thanh Nguyen, Milind Tambe, Nicole Sintov, and Francesco Delle Fave. 2015. Beware the soothsayer: From attack prediction accuracy to predictive reliability in security games. In *Proceedings of the Sixth Conference on Decision and Game Theory for Security(GameSec-15)*. London, 35–56.

[10] Shahrzad Gholami, Sara McCarthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, Joshua Mabonga, et al. 2018. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers. In *Proceedings of the Seventeenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-18)*. Stockholm, 823–831.

[11] Qingyu Guo, Boyuan An, Branislav Bosanskỳ, and Christopher Kiekintveld. 2017. Comparing Strategic Secrecy and Stackelberg Commitment in Security Games.. In *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*. Melbourne, 3691–3699.

[12] Nika Haghtalab, Fei Fang, Thanh Hong Nguyen, Arunesh Sinha, Ariel D Procaccia, and Milind Tambe. 2016. Three Strategies to Success: Learning Adversary Models in Security Games. In *Proceedings of the Twenty-five International Joint Conference on Artificial Intelligence (IJCAI-16)*. New York, 308–314.

[13] Jason S. Hartford, James R. Wright, and Kevin Leyton-Brown. 2016. Deep learning for predicting human strategic behavior. In *Advances in Neural Information Processing Systems 29 (NIPS-16)*. Barcelona, 2424–2432.

[14] Debarun Kar, Fei Fang, Francesco M. Delle Fave, Nicole Sintov, Milind Tambe, and Arnaud Lyet. 2016. Comparing human behavior models in repeated Stackelberg security games: An extended study. *Artificial Intelligence* 240 (2016), 65–103.

[15] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2011. Solving Stackelberg games with uncertain observability. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-11)*. Taipei, 1013–1020.

[16] Chun Kai Ling, Fei Fang, and J. Zico Kolter. 2018. What game are we playing? End-to-end learning in normal and extensive form games. In *Proceedings of the Twenty-seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*. Stockholm, 396–402.

[17] Chun Kai Ling, Fei Fang, and J. Zico Kolter. 2019. Large Scale Learning of Agent Rationality in Two-Player Zero-Sum Games. In *Proceedings of the Thirty-third AAAI Conference on Artificial Intelligence (AAAI-19)*. Honolulu.

[18] Richard D. McKelvey and Thomas R. Palfrey. 1995. Quantal response equilibria for normal form games. *Games and Economic Behavior* 10, 1 (1995), 6–38.

[19] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. 2013. Analyzing the Effectiveness of Adversary Modeling in Security Games. In *Proceedings of the Twenty-seventh AAAI Conference on Artificial Intelligence (AAAI-13)*. Bellevue, Washington, 718–724.

[20] Steven Okamoto, Noam Hazon, and Katia Sycara. 2012. Solving non-zero sum multiagent network flow security games with attack costs. In *Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-12)*. Valencia, 879–888.

[21] Arunesh Sinha, Debarun Kar, and Milind Tambe. 2016. Learning adversary behavior in security games: A PAC model perspective. In *Proceedings of the Fifteenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-16)*. Singapore, 214–222.

[22] Milind Tambe. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.

[23] Bryan Wilder, Bistra Dilkina, and Milind Tambe. 2019. Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization. In *Proceedings of the Thirty-third AAAI Conference on Artificial Intelligence (AAAI-19)*. Honolulu.

[24] James R. Wright and Kevin Leyton-Brown. 2017. Predicting human behavior in unrepeated, simultaneous-move games. *Games and Economic Behavior* 106 (2017), 16–37.

[25] Haifeng Xu. 2016. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, Maastricht, 497–514.

[26] Rong Yang, Benjamin Ford, Milind Tambe, and Andrew Lemieux. 2014. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the Thirteenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-14)*. Paris, 453–460.

[27] Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Richard John. 2011. Improving resource allocation strategy against human adversaries in security games. In *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*. Barcelona, 458–464.

[28] Yue Yin, Bo An, and Manish Jain. 2014. Game-theoretic resource allocation for protecting large public events. In *Proceedings of the Twenty-eighth AAAI Conference on Artificial Intelligence (AAAI-14)*. Quebec City, 826–833.

[29] Chao Zhang, Arunesh Sinha, and Milind Tambe. 2015. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *Proceedings of the Fourteenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-15)*. Istanbul, 1351–1359.