End-to-End Game-Focused Learning of Adversary Behavior in Security Games

Andrew Perrault,[1] Bryan Wilder,[1] Eric Ewing,[2] Aditya Mate,[1] Bistra Dilkina,[2] Milind Tambe[1]
[1]Center for Research on Computation and Society, Harvard
[2]Center for Artificial Intelligence in Society, University of Southern California
aperrault@g.harvard.edu, bwilder@g.harvard.edu, ericewin@usc.edu,
aditya_mate@g.harvard.edu, dilkina@usc.edu, milind_tambe@harvard.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 learning a defense that *generalizes* well to a new set of targets with novel feature values and combinations. Traditionally, this problem has been addressed via a *two-stage* approach where an adversary model is trained to maximize predictive accuracy without considering the defender's optimization problem. We develop an end-to-end *game-focused* approach, where the adversary model is trained to maximize a surrogate for the defender's expected utility. We show both in theory and experimental results that our game-focused approach achieves higher defender expected utility than the two-stage alternative when there is limited data.

1 Introduction

Many real-world settings call for allocating limited defender resources against a strategic adversary, such as protecting public infrastructure (Gan, An, and Vorobeychik 2015), transportation networks (Okamoto, Hazon, and Sycara 2012), large public events (Yin, An, and Jain 2014), urban crime (Zhang, Sinha, and Tambe 2015), and green security (Fang, Stone, and Tambe 2015). *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 (Tambe 2011).

In many SSG settings, the adversary's preferences over targets are not known a priori. In early work, the adversary's preferences were estimated via the judgments of human experts (Tambe 2011). In domains where there are many interactions with the adversary, we can leverage this history using machine learning instead. This line of work, started by Letchford et al. (2009), has received extensive attention in recent years (see related work).

We use protecting wildlife from poaching (Fang, Stone, and Tambe 2015) as a motivating example. The adversary's (poacher's) behavior is observable because snares are left behind, which rangers aim to remove (see Fig. 1a). Various features such as animal counts, distance to the edge of the park, weather and time of year may affect how attractive a particular target is to the adversary. The training data consists of adversary behavior in the context of particular sets of targets, and our objective is to achieve a high defender utility when we are playing against the same adversary and new sets of targets. For the problem of poaching prevention, Gholami et al. (2018) use around 20 features per target and observe tens of thousands of distinct targets (i.e., combinations of feature values). 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.

The standard approach to the problem breaks it into two stages. In the first, the adversary model is fit to the historical data to minimize an accuracy-based loss function, and in the second, the defender covers the targets (via a mixed strategy) to maximize utility against the learned model. It is true that, in a worst-case analysis, a model that is more accurate in a global sense induces a better coverage (see Sinha et al. (2016) and Haghtalab et al. (2016)), but a model that accurately predicts the *relative* values of "important" targets may achieve high defender utility with weak global accuracy. For example, in a game with many low-value targets, the estimates of the values of the low-value targets can be wildly inaccurate and still yield a high defender utility (see Sec. 3 for an example).

In our *game-focused* approach, in contrast to a two-stage approach, we focus on learning a model that yields a high defender expected utility from the start. We train a predictive model end-to-end (i.e., considering the effects of the optimization problem) using an estimate of defender expected utility as our loss function. This approach has the advantage of focusing learning on "important" targets that have a large impact on the defender expected utility, and not being distracted by irrelevant targets (e.g., those with low value for both the attacker and defender). For example, in our human subject data experiments, two-stage achieves 2–20% lower cross entropy, but worse defender expected utility. Performing game-focused training requires us to overcome several technical challenges, including forming counterfactual estimates of the defender's expected utility and differentiating through the solution of a nonconvex optimization problem.

In summary, our contributions are: *First*, we provide a theoretical justification for why our game-focused approach can outperform two-stage approaches in SSGs. *Second*, we overcome technical challenges to develop a game-focused learning pipeline for SSGs. *Third*, we test our approach on a combination of synthetic and human subject data and show that game-focused learning outperforms a two-stage approach in settings where the amount of data available is small and when there is wide variation in the adversary's values for the targets.

Related Work. There is a rich literature on SSGs, ranging from uncertain observability (Korzhyk, Conitzer, and Parr 2011) to disguised defender resources (Guo et al. 2017) to extensive-form models (Cermak et al. 2016) to patrolling on graphs (Basilico, Gatti, and Amigoni 2012; Basilico, De Nittis, and Gatti 2017). In particular, learning to maximize the defender's payoff from repeated play has been a subject of extensive study. It is important to distinguish between the active learning case (Letchford, Conitzer, and Munagala 2009; Xu, Tran-Thanh, and Jennings 2016; Blum, Haghtalab, and Procaccia 2017), where the defender may gather information through her choice of strategy, and the passive case, where the defender does not have control over the training data. We consider each case to be valuable but focus on the passive case because we believe it is encountered more frequently in domains of interest. In the anti-poaching setting, parks often have historical data that far exceeds what can be actively collected in the short term.

Bounded rationality models are a critical component of the SSG literature because they allow the defender to achieve higher utilities against many realistic attackers. They have been the subject of extensive study since their introduction by Pita et al. (2010) (e.g., Cui and John (2014) and Abbasi et al. (2016), who develop a distinct line of work, inspired by psychology). We focus on the *quantal response (QR)* (Yang et al. 2013) model and especially the *subjective utility quantal response (SUQR)* model (Nguyen et al. 2013). SUQR is simple, widely used and has been shown to be effective in practice.

Sinha et al. (2016) and Haghtalab et al. (2016) provide probabilistic bounds on the learning error for two-stage approaches for generalized SUQR attackers in the passive and weakly active cases, respectively. Both works translate these bounds into the guarantees on the defender's expected utility in the worst case. Our focus is on the orthogonal issue of how to train *any* differentiable predictive model end-to-end with gradient descent, including deep learning architectures that are the state of the art for many learning tasks. These methods can scale to many features and complicated relationships and are one of the main appeals of two-stage approaches. We use SUQR implemented on a neural network as an illustrative example, but our approach can be applied to other bounded rationality models, as we discuss in Sec. 4.

Outside of SSGs, Ling et al. (2018; 2019) use a differentiable QR equilibrium solver to reconstruct the payoffs of both players in a game from observed play. Hartford et al. (2016) and Wright and Leyton-Brown (2017) study the problem of predicting play in unseen two-player simultaneous-move games with a small number of actions per player, and Hartford et al. (2016) build a deep learning architecture for this purpose. These works focus on prediction rather than optimization.

We briefly discuss related work in end-to-end learning for decision-making in non-game-theoretic contexts (see Donti and Kolter (2017) for a more complete discussion). New technical issues arise due to the presence of the adversary, such as counterfactual estimation and nonconvexity. In their study of parameter sensitivity, Rockafellar and Wets (2009) provide a comprehensive theoretical analysis of differentiating through optimization. Bengio (1997) was first to train a learning system for a more complex task by directly differentiating through the outcome of applying parameterized rules. Amos and Kolter (2017) provide analytical derivatives for constrained convex problems. This analytic approach is extended to stochastic optimization by Donti et al. (2017) and to submodular optimization by Wilder et al. (2019). Demirovic et al. (2019) provide a theoretically optimal framework for ranking problems with linear objectives.

2 Setting

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 (Tambe 2011). Formally, an SSG is a tuple $\{\mathcal{T}, \boldsymbol{u}_d, \boldsymbol{u}_a, C_d\}$, where $\mathcal{T}$ is a set of targets, $\boldsymbol{u}_d : \mathcal{T} \to \mathbb{R}_{\leq 0}$ is the defender's payoff if each target is successfully attacked, $\boldsymbol{u}_a : \mathcal{T} \to \mathbb{R}_{\geq 0}$ 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.

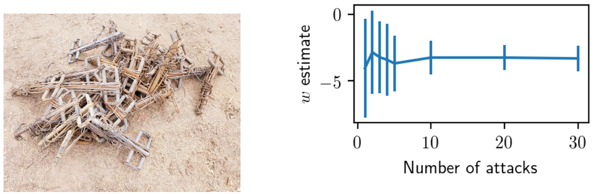
The game has two time steps: the defender computes a mixed strategy that satisfies the constraints C_d, which induces a *marginal coverage probability (or coverage)* $\boldsymbol{p} = \{\boldsymbol{p}_i : i \in \mathcal{T}\}$. The attacker's *attack function* $\boldsymbol{q}$ determines which target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

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

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 (McKelvey and Palfrey 1995) 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 (Nguyen et al. 2013) attacker attacks each target with probability proportional to the exponential of an *attractiveness function*:

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

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*. We focus our effort on learning ϕ because w can easily be learned using existing techniques, such as the maximum likelihood estimation (MLE) approach of Sinha et al. (2016), assuming we



(a) Snares removed by rangers in Srepok National Park, Cambodia.

(b) MLE estimate of w converges quickly to the true value of -4. Error bars indicate one standard deviation.

have the ability to play different defender strategies against the same set of targets. MLE estimates converge rapidly, as shown by Fig. 1b, which demonstrates learning in an eight-target game, averaged over 20 trials. Once we have an accurate w estimate, it can be transferred to all games against the same adversary.

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 attacks 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. 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 $\boldsymbol{q}$:

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

To achieve this, she can observe the attacker's behavior in the training data and learn how he values different combinations of features.

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, which is illustrated in the top of Fig. 2, 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 $\hat{\boldsymbol{q}}$:

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

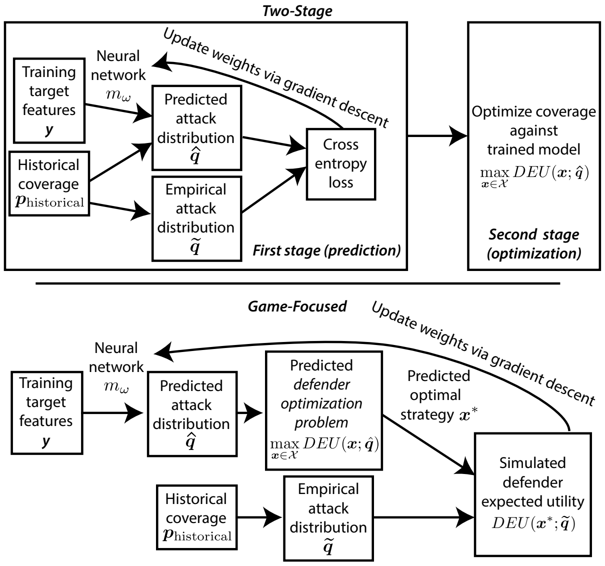

Two-Stage
Training target features $\boldsymbol{y}$
Neural network m_ω
Update weights via gradient descent
Predicted attack distribution $\hat{\boldsymbol{q}}$
Historical coverage $\boldsymbol{p}_{\text{historical}}$
Cross entropy loss
Empirical attack distribution $\tilde{\boldsymbol{q}}$
First stage (prediction)
Optimize coverage against trained model
$\max_{\boldsymbol{x}\in\mathcal{X}} DEU(\boldsymbol{x}; \hat{\boldsymbol{q}})$
Second stage (optimization)
Game-Focused
Update weights via gradient descent
Training target features $\boldsymbol{y}$
Neural network m_ω
Predicted attack distribution $\hat{\boldsymbol{q}}$
Predicted defender optimization problem
$\max_{\boldsymbol{x}\in\mathcal{X}} DEU(\boldsymbol{x}; \hat{\boldsymbol{q}})$
Predicted optimal strategy $\boldsymbol{x}^*$
Simulated defender expected utility
$DEU(\boldsymbol{x}^*; \tilde{\boldsymbol{q}})$
Historical coverage $\boldsymbol{p}_{\text{historical}}$
Empirical attack distribution $\tilde{\boldsymbol{q}}$

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

3 Impact of Two-Stage Learning on DEU

We begin by developing intuitions about when an inaccurate predictive model can lead to high defender expected utility. We study the rational attacker case for simplicity—results in the rational case can be directly translated to the QR case (which is a smooth version of rationality). Consider an SSG with three targets and a single defense resource. The defender has equal value for all three and the attacker has true values of $(0.4, 0.4, 0.2)$, yielding an optimal coverage of $p^* = (0.5, 0.5, 0.0)$. Suppose the defender estimates the attacker's target values to be $(0.5, 0.5, 0.0)$. This estimate yields the optimal coverage, despite overestimating the value of the first two targets by 25% and underestimating the value of the third by 100%. In contrast, the estimate $(0.4 - \epsilon, 0.4 - \epsilon, 0.2 + 2\epsilon)$ does not yield optimal coverage despite being within ϵ of the ground truth target values.

We characterize the extent to which two predictive models with the same accuracy-based loss can differ in terms of the defender's expected utility for rational attacker, two-target SSGs with both equal and zero-sum defender target values. From the perspective of a two-stage approach with an accuracy-based loss, any two models with the same loss are considered equally good. *In contrast, a game-focused model with an oracle for the defender's expected utility would automatically prefer a model with higher defender utility.* We additionally extend the latter result to QR attackers.

The theory shows two key points. First, the error in estimates of attacker's utilities can have highly variable effects on the defender's expected utility. As we saw in the example, estimation error can have no effect in certain cases. The defender's preference for the distribution of estimation error depends on both the relative values of the targets and the correlation between the target values of the attacker and defender. These properties are challenging to replicate in hand-tuned two-stage approaches. Second, game-focused learning is more beneficial when the attacker's true values across targets exhibit greater variance. We return to this intuition in our experiments.

We begin with the case where the defender values all targets equally (and recall that we assume that both the attacker and defender receive a payoff of zero for an unsuccessful attack). For complete proofs of all theorems, see the full version of the paper.

Theorem 1 (Equal defender values)**.** *Consider a two-target SSG with a rational attacker, equal defender values for each target, 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. Let the defender's values for the targets be -1 for each.*

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 $\epsilon^2 \leq (1 - z_0)^2$ and $\epsilon^2 \leq (z_0 - z_1)^2$, 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:*

$$\frac{z_0 + \epsilon}{z_1 + \epsilon} \tag{6}$$

Proof Sketch. There are two normalized estimates of the attacker's values that have MSE ϵ^2: $(z_0 + \epsilon, z_1 - \epsilon)$ and $(z_0 - \epsilon, z_1 + \epsilon)$. The attacker will attack the target whose value the defender underestimates. The defender prefers the latter case, where the attacker selects the higher value target, because this target has more coverage and successful attacks have the same cost on both targets. □

Thus, in the equal value case, it is generally better for the defender to underestimate the attacker's values for high-value targets. This dynamic is reversed in the zero-sum case.

Theorem 2 (Zero-sum)**.** *Consider the same setting as Thm. 1 except the utilities are zero-sum. 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:*

$$\frac{(1 - (z_1 - \epsilon))z_1}{(1 - (z_0 - \epsilon))z_0} \tag{7}$$

Proof Sketch. Similarly to Thm. 1, there are two value estimates with MSE ϵ^2. The defender prefers the case where she underestimates the attacker's value for the lower value target, inducing the attacker to attack it. The lower cost of failures outweighs the attacker getting caught less often. □

The theory can be extended to QR attackers. In the case of Thm. 2, 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 (Zero-sum, QR attacker)**.** *Consider the setting of Thm. 2, 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.*

4 Game-Focused Learning in SSGs

We now present our approach to game-focused learning in SSGs. The key idea is to embed the defender optimization problem into training and compute gradients of DEU with respect to the model's predictions, which requires us to overcome two technical challenges. First, in the previous section, we assumed we had access to an exact oracle for the defender's expected utility, but in practice, this is a counterfactual estimation problem. Second, our defender's optimization is nonconvex and new machinery is required to calculate the derivative of the solution w.r.t. its parameters. We illustrate our approach in the bottom of Fig. 2.

We begin with notation. As we have discussed, the standard two-stage 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 would like to learn a function m_ω which takes a set of targets and associated features as input and produces $\hat{\boldsymbol{q}}$ as output, which then induces a coverage with high expected utility. Note that from a utility-theoretic perspective, it does not matter how accurate $\hat{\boldsymbol{q}}$ is, only that the induced coverage has high expected utility. Let

$$\boldsymbol{x}^*(\hat{\boldsymbol{q}}) = \underset{\boldsymbol{x} \text{ satisfying } C_d}{\arg\max} \ \text{DEU}(\boldsymbol{x}; \hat{\boldsymbol{q}}) \tag{8}$$

be the optimal defender coverage function against an adversary with attack function $\hat{\boldsymbol{q}}$. Our goal is to find $\hat{\boldsymbol{q}}$ which maximizes

$$\text{DEU}(\hat{\boldsymbol{q}}) = \underset{\langle \mathcal{T}, \boldsymbol{y}, \boldsymbol{u}_d \rangle \sim D_{\text{test}}}{\mathbb{E}} \left[\text{DEU}(\boldsymbol{x}^*(\hat{\boldsymbol{q}}); \boldsymbol{q}) \right], \tag{9}$$

$\text{DEU}(\hat{\boldsymbol{q}})$ is the ground truth expected utility of coverage $\boldsymbol{x}^*(\hat{\boldsymbol{q}})$ (recall that $\boldsymbol{q}$ is the attacker's true response function). While we do not have access to D_{test}, we can estimate Expr. 9 using samples from D_{train}. We would like to calculate the derivative of Expr 9 w.r.t. $\hat{\boldsymbol{q}}$ to use in model training. Using the chain rule:

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

Here, $\frac{\partial \text{DEU}(\boldsymbol{x}^*(\hat{\boldsymbol{q}}); \boldsymbol{q})}{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}$ describes how the defender's true utility with respect to $\boldsymbol{q}$ changes as a function of her strategy $\boldsymbol{x}^*$, which is a *counterfactual* question because we only observe the defender playing a single strategy in this training game. $\frac{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}}$ describes how $\boldsymbol{x}^*$ depends on the estimated attack function $\hat{\boldsymbol{q}}$, which requires differentiating through the nonconvex optimization problem in Eq. 8. If we had a means of calculating both terms, we could then estimate $\frac{\partial \text{DEU}(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}}$ by sampling 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), the entire system may be trained end-to-end via gradient descent. We address each of the two terms separately.

Counterfactual Adversary Estimates

We want to calculate $\frac{\partial \text{DEU}(\boldsymbol{x}^*(\hat{\boldsymbol{q}});\boldsymbol{q})}{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}$ 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. We find that typical datasets only contain a set of sampled attacker responses to a particular historical defender mixed strategy or a small set of mixed strategies. Previous work on end-to-end learning for decision problems (Bengio 1997; Donti, Amos, and Kolter 2017; Wilder, Dilkina, and Tambe 2019; Demirovic et al. 2019) assumes that the historical data specifies the utility of *any* possible decision, but this assumption does not hold in SSGs because they are interactions between strategic agents.

Our approach relies on the adversary using a bounded rationality model that is stochastic and *decomposable*. It is generally the case that boundedly rational adversaries complicate the process of learning and optimizing in SSGs, e.g., because they cause the optimization to become nonconvex and they add uncertainty to the defender's adversary model. However, bounded rationality is critical to our counterfactual reasoning strategy because boundedly rational adversaries reveal information about their entire ranking of targets over repeated games against the same defender strategy. For example, consider a three-target game where the defender has covered all three targets equally. QR attackers attack each target proportionally to the expected utility it provides, eventually revealing the attacker's relative utilities across all of the targets under that particular defender coverage. Without the stochasticity, we would unable to learn anything other than the attacker's most preferred target.

The resulting target value estimates are in the context of one particular defender strategy. To estimate the attacker's response to *any* defender coverage, we need to substitute the historical coverage for an arbitrary one. At first glance, this may seem impossible for a stochastic, bounded rationality model because the attacker could have an arbitrary response to coverage. If we had a rational attacker instead, with known target values, we could compute his reaction to an arbitrary defender coverage, but we could not estimate his relative values for each target (as previously discussed). Here we exploit the decomposability of many bounded rationality models: the impact of the defender's coverage can be separated from the values of the targets.

We develop an illustrative example of the pipeline for SUQR. We observe samples from the attack distribution $\boldsymbol{q}$, where for SUQR, $\boldsymbol{q}_i \propto \exp(w\boldsymbol{p}_i + \phi(\boldsymbol{y}_i))$. Because we can estimate $\boldsymbol{q}_i$ from the empirical attack frequencies and the term $w\boldsymbol{p}_i$ is known (see Sec. 2), we can invert the exp function to obtain an estimate of $\phi(\boldsymbol{y}_i)$. Formally, this corresponds to setting $\hat{\phi}(\boldsymbol{y}_i)$ to the MLE under the empirical attack distribution:

$$\hat{\phi}(\boldsymbol{y}_i) = \underset{\boldsymbol{\phi}}{\text{argmax}} \prod_{a \in \mathcal{A}} \Pr(\text{Categorical}(\exp(w\boldsymbol{p}_i + \boldsymbol{\phi})) = a).$$

By exploiting decomposability, we derive relative target value estimates that can be used to estimate the attacker's behavior under an arbitrary coverage. Our estimates have two key limitations. First, they do not provide us with any information about the ϕ function for values other than $\boldsymbol{y}_i$ and second, they are unique only up to a constant additive factor. Despite these limitations, they suffice to allow us to simulate the defender's expected utility for any training data point $\langle \mathcal{T}, \boldsymbol{y}, \mathcal{A}, \boldsymbol{u}_d, \boldsymbol{p}_{\text{historical}} \rangle$ as

$$\sum_{i \in T} (1 - \boldsymbol{x}^*(\hat{\boldsymbol{q}})_i) \exp(w\boldsymbol{x}^*(\hat{\boldsymbol{q}})_i + \hat{\phi}(\boldsymbol{y}_i)) \boldsymbol{u}_d(i). \tag{10}$$

We briefly discuss two issues that arise when applying this procedure to other bounded rationality models. First, the model needs to provide meaningful $\hat{\phi}$ estimates, which is where the rational attacker model fails. Second, the model needs to be decomposable into the effects of coverage and the inherent attractiveness of the targets, and the parameters of this decomposition need to be easily estimable (as we show is the case for SUQR in Sec. 2). Most models satisfy this condition, including SHARP (Kar et al. 2016), PT and QBRM (Abbasi et al. 2015).

Gradients of Nonconvex Optimization

The optimization problem which produces $\boldsymbol{x}^*(\hat{\boldsymbol{q}})$ is typically nonconvex when the adversary is boundedly rational. This complicates the process of differentiating through the defender problem to obtain $\frac{\partial \boldsymbol{x}^*(\hat{\boldsymbol{q}})}{\partial \hat{\boldsymbol{q}}}$, as previous approaches rely on either a convex optimization problem (Donti, Amos, and Kolter 2017) or a cleverly chosen convex surrogate for a nonconvex problem (Wilder, Dilkina, and Tambe 2019). In contrast, our approach produces correct gradients for many nonconvex problems. The key idea is to fit a quadratic program around the optimal point returned by a blackbox nonconvex solver. Intuitively, this works well when the local neighborhood is, in fact, convex, and fortunately, this is the case for many optimization problems against boundedly rational attackers.

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. *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 $\min_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{2} \boldsymbol{x}^\top \nabla^2 f(\boldsymbol{x}, \theta) \boldsymbol{x} + \boldsymbol{x}^\top \nabla f(\boldsymbol{x}, \theta)$.*

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 (Amos and Kolter 2017) to the local quadratic approximation. It is easy to verify that the defender utility maximization problem for an SUQR attacker satisfies the assumptions of Thm. 4 since the objective is smooth and typical constraint sets for SSGs are polytopes with nonempty interior (see (Xu 2016) for a list of examples). 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.

5 Experiments

We begin by comparing the performance of game-focused and two-stage approaches across a range of settings both simulated and real. We find that game-focused learning outperforms two-stage when the number of training games is low, the number of attacks observed in each training game is low, and the number of target features is high. As the amount of training data increases, two-stage starts catching up as it is able to reconstruct the attacker model accurately. We dedicate the second part of the experiments section to investigating three hypotheses for why game-focused achieves superior performance.

Defender Strategies. We compare the following three defender strategies: *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. *Game-focused (*GF*)* is our game-focused approach. *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. *Game-tuned two-stage (2S-GT)* is a regularized approach that aims to maximize the defender's expected utility when the amount of data is small. It uses Dropout (Srivastava et al. 2014) and a validation set for early stopping. All three methods use the same architecture for the prediction neural network: a fully-connected single-layer network with 200 hidden units on the synthetic data and 10 hidden units on the simpler human subject data.

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 (2013) 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 (Nguyen et al. 2013; Yang et al. 2014). 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.

Results Fig. 3 shows the results of the experiments in simulation, comparing the defender strategies across a variety of problem types. GF yields higher DEU than the other methods across most tested parameter settings and GF 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 $\text{DEU}(\text{UNIF}) = -2.5$ for 8 targets and $\text{DEU}(\text{UNIF}) = -4.2$ for 24.

The left column of Fig. 3 compares DEU as the number of attacks observed per game increases. For both 8 and 24 targets, GF receives higher DEU than the other methods across the tested range. We provide the results of paired sample T-test between GF and 2S-GT in the appendix, which shows that the differences are statistically significant at $p < 0.05$. The center column of Fig. 3 compares DEU as the number of training games increases. Likewise, GF outperforms the other methods. The right column of Fig. 3 compares DEU as the number of features decreases. Here we see GF outperforming 2S-GT when the number of features is large (and thus, the learning problem is harder) and vice versa when the number of features is small.

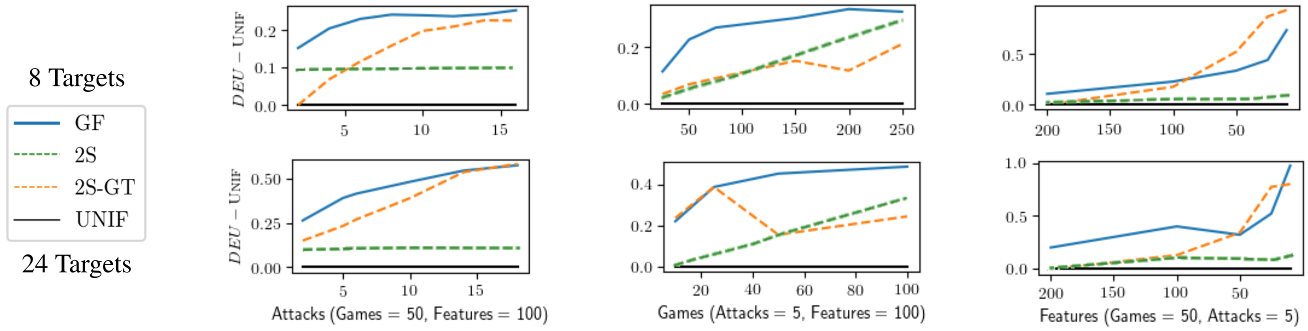


Figure 3: DEU − UNIF across the three strategies as we vary the number of features, number of training games and number of observed attacks per training game. When not varied, the parameter values are 100 features, 50 training games and 5 attacks per game. GF receives higher DEU than 2S for most parameter values.

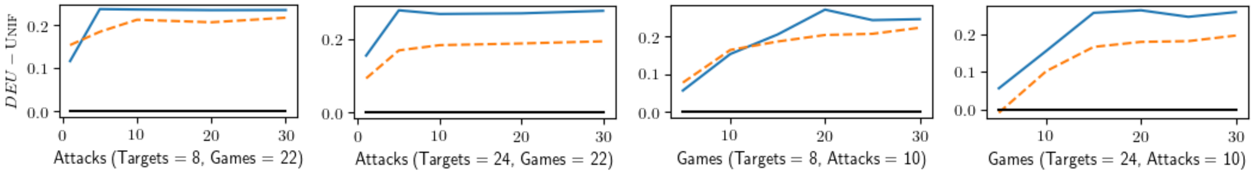


Figure 4: DEU − UNIF from human subject data for 8 and 24 targets, as the number of attacks per training game is varied and number of training games is varied. GF receives higher DEU for most settings, especially for 24-target games.

In all three columns, 2S-GT catches up to GF as the learning problem becomes easier. With enough data, the gap between the two methods closes as both approach optimality. This fact is reflected by Thms. 1 and 2: as the model error ϵ decreases, the DEU difference between the best and worst model with error ϵ decreases. We observe a standard relationship between 2S and 2S-GT: the model that is tuned for small data mostly performs better. The untuned 2S model benefits from increasing the number of training games, but increasing the number of attacks or decreasing the number of features has little effect.

Experiments on Human Subject Data

We use data from human subject experiments performed by Nguyen et al. (2013). 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. (2013): -8.23. We use MLE 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.

Results We find that GF receives higher DEU than 2S-GT on the human subject data (as 2s-GT outperformed 2S in the synthetic data case, we do not present results for 2s here). The differences are statistically significant at $p < 0.05$ in the 24-target case and at the border of significance in the 8-target case. Fig. 4 summarizes our results as the number of training attacks per target and games are varied. Varying the number of attacks, for 8 targets, GF achieves its highest percentage improvement in DEU at 5 attacks where it receives 28% more than 2S-GT. For 24 targets, GF achieves its largest improvement of 66% more DEU than 2S at 1 attack. Varying the number of games, GF outperforms 2S-GT 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-GT barely outperforms UNIF.

Unlike in the synthetic data experiments, we do not observe 2S-GT catching up to GF using the data that we allocate for training. A key difference between the human subject data experiments and the synthetic data experiments is the presence of noise in the former. In the latter, there exists a ground-truth attractiveness function that, if learned, would reproduce the attack distribution exactly. With human subject data, we do not expect this to be the case: there are other features that are not available to the model such as the position of the target on the screen and learning effects.

Discussion

Our experimental results establish that GF outperforms the two-stage approaches under a variety of instance parameter settings. We now focus on understanding why GF produces predictions that lead to superior defender utility. We study three hypotheses. We test our hypothesis with 24 targets, 100 features, 100 games, and 5 attacks unless specified otherwise.

Hypothesis 1: GF makes better predictions. A natural starting point is whether the differences in performance can be explained purely by the quality of predictions. This is the standard position in the literature—better predictive adversary models lead to higher defender expected utility. It would be surprising if this hypothesis were true because GF does not explicitly optimize for prediction accuracy and two-stage does.

The human subject experiments produce strong evidence against this hypothesis. Even when GF has a large advantage in defender expected utility, it has test cross entropy that is 2–20% higher than 2s-GT.

Hypothesis 2: GF handles model uncertainty better. From a Bayesian perspective, the training data induce a posterior distribution over the potential adversary attractiveness functions. Thus, the defender's ideal optimization, i.e., the one that yields the highest expected utility, is stochastic over this distribution of attackers. Because GF is an end-to-end approach, it may handle the uncertainty over attacker models better by learning to represent the distribution as a point estimate that induces the correct solution. We test this hypothesis by using the 2S test cross entropy as a surrogate for the uncertainty in the attacker model. Low cross entropy indicates that the model learned by 2S was close to the true model, and this indicates that there is little advantage to taking model uncertainty into account in the optimization. We would hypothesize that GF would be weaker in comparison when this occurs and stronger when 2S cross entropy is higher.

The results are shown on the left side of Fig. 5. The x-axis shows 2S test cross entropy and the y-axis is the gap between GF and 2S. This hypothesis fails: when there is less model uncertainty, GF performs better relative to 2S.

Hypothesis 3: GF learns more accurate models for more important targets. The different loss function used by GF may induce a different distribution of the errors across targets. Because errors on more important targets have a greater impact on the defender's expected utility, we hypothesize that GF will make smaller errors on important targets and larger errors on unimportant ones relative to 2S.

Fig. 6 supports this hypothesis. The x-axis is the target's predicted contribution to DEU under the coverage selected by the defender strategy, and the y-axis shows the absolute error in the predicted probability that the attacker attacks that target. GF has larger errors for targets that contribute less to DEU and smaller errors for targets that contribute more.

6 Conclusion

We advance the state of the art in learning adversary models in SSGs with the goal of maximizing defender expected utility. In contrast to past approaches, our approach allows modern deep learning architectures to be trained, and we outperform even two-stage approaches that have been tuned to maximize the defender's expected utility. We investigate empirically and theoretically why our game-focused approach outperforms two-stage and find that harder decision problems lead to better game-focused performance. We believe that our conclusions have important consequences for future research and that our game-focused approach can be extended to a variety of SSG models where smooth nonconvex objectives and polytope feasible regions are common.

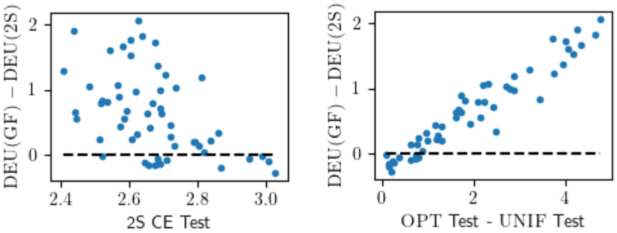


Figure 5: DEU(GF) − DEU(2S). Each point represents one trial. GF performs better when 2S has lower test cross entropy and when target values are less uniform.

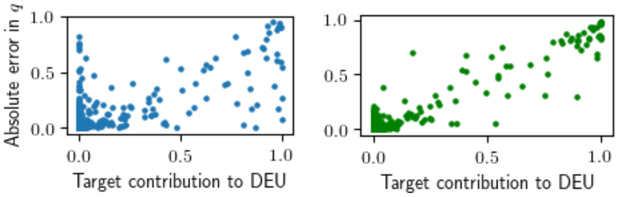


Figure 6: Target contribution to DEU vs. the absolute error in the predicted attacker q. GF (left) has lower estimation errors for targets with high DEU contributions and higher errors for targets with low DEU contributions. 2S (right) estimation errors do not vary with target importance.

Acknowledgments. Perrault was supported by the Center for Research on Computation and Society. This work was supported by the Army Research Office (MURI W911NF1810208).

References

Abbasi, Y. D.; Short, M.; Sinha, A.; Sintov, N.; Zhang, C.; and Tambe, M. 2015. Human adversaries in opportunistic crime security games: evaluating competing bounded rationality models. In *Proc. of Advances in Cognitive Systems*.

Abbasi, Y. D.; Ben-Asher, N.; Gonzalez, C.; Morrison, D.; Sintov, N.; and Tambe, M. 2016. Adversaries wising up: Modeling heterogeneity and dynamics of behavior. In *Proc. of Intl. Conf. on Cognitive Modeling*.

Amos, B., and Kolter, J. Z. 2017. OptNet: Differentiable optimization as a layer in neural networks. In *ICML-17*.

Basilico, N.; De Nittis, G.; and Gatti, N. 2017. Adversarial patrolling with spatially uncertain alarm signals. *AIJ* 246:220–257.

Basilico, N.; Gatti, N.; and Amigoni, F. 2012. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *AIJ* 184:78–123.

Bengio, Y. 1997. Using a financial training criterion rather than a prediction criterion. *Intl. J. of Neural Systems* 8:433–443.

Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2017. *Learning to Play Stackelberg Security Games*. Cambridge University Press. 604–626.

Cermak, J.; Bosansky, B.; Durkota, K.; Lisy, V.; and Kiekintveld, C. 2016. Using correlated strategies for computing Stackelberg equilibria in extensive-form games. In *AAAI-16*.

Cui, J., and John, R. S. 2014. Empirical comparisons of descriptive multi-objective adversary models in Stackelberg security games. In *GameSec-14*, 309–318.

Demirovic, E.; Stuckey, P. J.; Bailey, J.; Chan, J.; Leckie, C.; Ramamohanarao, K.; and Guns, T. 2019. Predict+optimise with ranking objectives: Exhaustively learning linear functions. In *IJCAI-19*, 1078–1085.

Donti, P.; Amos, B.; and Kolter, J. Z. 2017. Task-based end-to-end model learning in stochastic optimization. In *NIPS-17*, 5484–5494.

Fang, F.; Stone, P.; and Tambe, M. 2015. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI-15*, 2589–2595.

Gan, J.; An, B.; and Vorobeychik, Y. 2015. Security games with protection externalities. In *AAAI-15*, 914–920.

Gholami, S.; McCarthy, S.; Dilkina, B.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Nsubaga, M.; Mabonga, J.; et al. 2018. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers. In *AAMAS-18*, 823–831.

Guo, Q.; An, B.; Bosanskỳ, B.; and Kiekintveld, C. 2017. Comparing strategic secrecy and Stackelberg commitment in security games. In *IJCAI-17*, 3691–3699.

Haghtalab, N.; Fang, F.; Nguyen, T. H.; Sinha, A.; Procaccia, A. D.; and Tambe, M. 2016. Three strategies to success: Learning adversary models in security games. In *IJCAI-16*, 308–314.

Hartford, J. S.; Wright, J. R.; and Leyton-Brown, K. 2016. Deep learning for predicting human strategic behavior. In *NIPS-16*, 2424–2432.

Kar, D.; Fang, F.; Fave, F. M. D.; Sintov, N.; Tambe, M.; and Lyet, A. 2016. Comparing human behavior models in repeated Stackelberg security games: An extended study. *AIJ* 240:65–103.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2011. Solving Stackelberg games with uncertain observability. In *AAMAS-11*, 1013–1020.

Letchford, J.; Conitzer, V.; and Munagala, K. 2009. Learning and approximating the optimal strategy to commit to. In Mavronicolas, M., and Papadopoulou, V. G., eds., *Algorithmic Game Theory*, 250–262. Berlin, Heidelberg: Springer Berlin Heidelberg.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2018. What game are we playing? End-to-end learning in normal and extensive form games. In *IJCAI-18*, 396–402.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2019. Large scale learning of agent rationality in two-player zero-sum games. In *AAAI-19*.

McKelvey, R. D., and Palfrey, T. R. 1995. Quantal response equilibria for normal form games. *Games and Economic Behavior* 10(1):6–38.

Nguyen, T. H.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the effectiveness of adversary modeling in security games. In *AAAI-13*.

Okamoto, S.; Hazon, N.; and Sycara, K. 2012. Solving non-zero sum multiagent network flow security games with attack costs. In *AAMAS-12*, 879–888. Valencia.

Pita, J.; Jain, M.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2010. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *AIJ* 174:1142–1171.

Rockafellar, R. T., and Wets, R. J.-B. 2009. *Variational Analysis*, volume 317. Springer Science & Business Media.

Sinha, A.; Kar, D.; and Tambe, M. 2016. Learning adversary behavior in security games: A PAC model perspective. In *AAMAS-16*, 214–222.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* 15(1):1929–1958.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.

Wilder, B.; Dilkina, B.; and Tambe, M. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *AAAI-19*, 1658–1666.

Wright, J. R., and Leyton-Brown, K. 2017. Predicting human behavior in unrepeated, simultaneous-move games. *Games and Economic Behavior* 106:16–37.

Xu, H.; Tran-Thanh, L.; and Jennings, N. R. 2016. Playing repeated security games with no prior knowledge. In *AAMAS-16*, 104–112.

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

Yang, R.; Kiekintveld, C.; Ordez, F.; Tambe, M.; and John, R. 2013. Improving resource allocation strategies against human adversaries in security games: An extended study. *AIJ* 195:440 – 469.

Yang, R.; Ford, B.; Tambe, M.; and Lemieux, A. 2014. Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS-14*, 453–460.

Yin, Y.; An, B.; and Jain, M. 2014. Game-theoretic resource allocation for protecting large public events. In *AAAI-14*, 826–833.

Zhang, C.; Sinha, A.; and Tambe, M. 2015. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *AAMAS-15*, 1351–1359.