The Risk of Attacker Behavioral Learning: Can Attacker Fool Defender under Uncertainty?

Thanh Hong Nguyen[1] and Amulya Yadav[2]

[1] University of Oregon, Eugene, OR 97403 thanhhng@cs.uoregon.edu
[2] Pennsylvania State University, University Park, PA 16802 amulya@psu.edu

Abstract. In security games, the defender often has to predict the attacker's behavior based on some observed attack data. However, a clever attacker can intentionally change its behavior to mislead the defender's learning, leading to an ineffective defense strategy. This paper investigates the attacker's *imitative behavior deception under uncertainty*, in which the attacker mimics a (deceptive) behavior model by consistently playing according to that model, given that it is uncertain about the defender's learning outcome. We have three main contributions. First, we introduce a new maximin-based algorithm to compute a robust attacker deception decision. Second, we propose a new counter-deception algorithm to tackle the attacker's deception. We show that there is a *universal* optimal defense solution, regardless of any private knowledge the defender has about the relation between his learning outcome and the attacker deception choice. Third, we conduct extensive experiments, demonstrating the effectiveness of our proposed algorithms.

1 Introduction

In many real-world security domains, security agencies (defender) attempt to predict the attacker's future behavior based on some collected attack data, and use the prediction result to determine effective defense strategies. A lot of existing work in security games has thus focused on developing different behavior models of the attacker [24,27,20]. Recently, the challenge of playing against a deceptive attacker has been studied, in which the attacker can manipulate the attack data (by changing its behavior) to *fool* the defender, making the defender learn a wrong behavior model of the attacker [18]. Such deceptive behavior by the attacker can lead to an ineffective defender strategy.

A key limitation in existing work is the assumption that the defender has full access to the attack data, which means the attacker knows exactly what the learning outcome of the defender would be. However, in many real-world domains, the defender often has limited access to the attack data, e.g., in wildlife protection, park rangers typically cannot find all the snares laid out by poachers in entire conservation areas [8]. As a result, the learning outcome the defender obtains (with limited attack data) may be different from the deception behavior model that the attacker commits to. Furthermore, the attacker (and the defender) may have imperfect knowledge about the relation between the deception choice of the attacker and the actual learning outcome of the defender.

We address this limitation by studying the challenge of attacker deception given such uncertainty about the deception-learning relationship. We consider a security game model in which the defender adopts Quantal Response (QR), a well-known behavior model which has been widely used in economics and game theory [15,16,27], to predict the attacker's behavior, where the model parameter $\lambda \in \mathbb{R}$ is trained based on some attack data. On the other hand, the attacker plays deceptively by mimicking a QR model with a different value of λ, denoted by λ^{dec}. In this work, we incorporate the deception-learning uncertainty into this game model, where the learning outcome of the defender (denoted by λ^{learnt}) can be any value within a range centered at λ^{dec}.

We provide the following key contributions. First, we present a new maximin-based algorithm to compute an optimal robust deception strategy for the attacker. At a high level, our algorithm works by maximizing the attacker's utility under the worst-case of uncertainty. The problem comprises of three nested optimization levels, which is not straightforward to solve. We thus propose an alternative single-level optimization problem based on partial discretization. Despite this simplification, the resulting optimization is still challenging to solve due to the non-convexity of the attacker's utility and the dependence of the uncertainty set on λ^{dec}. By exploiting the decomposibility of the deception space and the monotonicity of the attacker's utility, we show that the alternative relaxed problem can be solved optimally in polynomial time.

Second, we propose a new counter-deception algorithm, which generates an optimal defense function that outputs a defense strategy for each possible (deceptive) learning outcome. Our key finding is that there is a *universal* optimal defense function for the defender, regardless of any additional information he has about the relation between his learning outcome and the deception choice of the attacker (besides the common knowledge that the learning outcome is within a range around the deception choice). Importantly, this optimal defense function, which can be determined by solving a single non-linear program, only generates two different defense strategies despite the infinite-sized learning outcome space.

Third, we conduct extensive experiments to evaluate our proposed algorithms in different game settings. Our results show that (i) despite the uncertainty, the attacker still obtains a significantly higher utility by playing deceptively; and (ii) the defender can substantially diminish the impact of the attacker's deception when following our counter-deception algorithm.

2 Related Work

Parameterized models of attacker behavior such as Quantal Response, and other machine learning models have been studied for SSGs [8,13,1]. These models provide general techniques for modeling the attacker decision making. Prior work assumes that the attacker always plays truthfully. Thus, existing algorithms for generating defense strategies would be vulnerable against deceptive attacks by an attacker who is aware of the defender's learning. Our work addresses such a strategic deceptive attacker by planning counter-deception defense strategies.

Deception is widely studied in security research [3,5,11,28,10,6]. In SSG literature, a lot of prior work has studied deception by the defender, i.e., the defender exploits his knowledge regarding uncertainties to mislead the attacker's decision making [9,21,22,26]. Recently, deception on the attacker's side has been studied. Existing work focuses on situations in which the defender is uncertain about the attacker type [7,19,4]. Some study the attacker behavior deception problem [18,17]. They assume that the attacker knows exactly the learning outcome while in our problem, the attacker is uncertain about that learning outcome.

Our work is also related to poisoning attacks in adversarial machine learning in which an adversary can contaminate the training data to mislead ML algorithms [2,12,23,25]. Existing work in adversarial learning uses prediction accuracy as the measure to analyzing such attacks, while in our game setting, the final goals of players are to optimize their utility, given some learning outcome.

3 Background

Stackelberg Security Games (SSGs) . There is a set of $\mathbf{T} = \{1, 2, \ldots, T\}$ targets that a defender has to protect using $L < T$ security resources. A pure strategy of the defender is an allocation of these L resources over the T targets. A mixed strategy of the defender is a probability distribution over all pure strategies. In this work, we consider the no-scheduling-constraint game setting, in which each defender mixed strategy can be compactly represented as a coverage vector $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$, where $x_t \in [0, 1]$ is the probability that the defender protects target t and $\sum_t x_t \leq L$ [14]. We denote by $\mathbf{X}$ the set of all defense strategies. In SSGs, the defender plays first by committing to a mixed strategy, and the attacker responds against this strategy by choosing a single target to attack.

When the attacker attacks target t, it obtains a reward R_t^a while the defender receives a penalty P_t^d if the defender is not protecting that target. Conversely, if the defender is protecting t, the attacker gets a penalty $P_t^a < R_t^a$ while the defender receives a reward $R_t^d > P_t^d$. The expected utility of the defender, $U_t^d(x_t)$ (and attacker's, $U_t^a(x_t)$), if the attacker attacks target t are computed as follows:

$$U_t^d(x_t) = x_t R_t^d + (1 - x_t) P_t^d \qquad U_t^a(x_t) = x_t P_t^a + (1 - x_t) R_t^a$$

Quantal Response Model (QR). QR is a well-known behavioral model used to predict boundedly rational (attacker) decision making in security games [15,16,27]. Essentially, QR predicts the probability that the attacker attacks each target t using the following softmax function:

$$q_t(\mathbf{x}, \lambda) = \frac{e^{\lambda U_t^a(x_t)}}{\sum_{t'} e^{\lambda U_{t'}^a(x_{t'})}} \tag{1}$$

where λ is the parameter that governs the attacker's rationality. When $\lambda = 0$, the attacker attacks every target uniformly at random. When $\lambda = +\infty$, the attacker is perfectly rational. Given that the attacker follows QR, the defender

and attacker's expected utility is computed as an expectation over all targets:

$$U^d(\mathbf{x}, \lambda) = \sum_t q_t(\mathbf{x}, \lambda) U_t^d(x_t) \tag{2}$$

$$U^a(\mathbf{x}, \lambda) = \sum_t q_t(\mathbf{x}, \lambda) U_t^a(x_t) \tag{3}$$

The attacker's utility $U^a(\mathbf{x}, \lambda)$ was proved to be increasing in λ [18]. We leverage this monotonicity property to analyze the attacker's deception. In SSGs, the defender can learn λ based on some collected attack data, denoted by λ^{learnt}, and find an optimal strategy which maximizes his expected utility accordingly:

$$\max_{\mathbf{x} \in \mathbf{X}} U^d(\mathbf{x}, \lambda^{\text{learnt}})$$

4 Attacker Behavior Deception under Uncertainty

We first study the problem of imitative behavior deception in an uncertainty scenario in which the attacker is uncertain about the defender's learning outcome. Formally, if the attacker plays according to a particular parameter value of QR, denoted by λ^{dec}, the learning outcome of the defender can be any value within the interval $[\max\{\lambda^{\text{dec}} - \delta, 0\}, \lambda^{\text{dec}} + \delta]$, where $\delta > 0$ represents the extent to which the attacker is uncertain about the learning outcome of the defender. We term this interval, $[\max\{\lambda^{\text{dec}} - \delta, 0\}, \lambda^{\text{dec}} + \delta]$, as the *uncertainty range* of λ^{dec}. We are particularly interested in the research question:

Given uncertainty about learning outcomes, can the attacker still benefit from playing deceptively?

In this section, we consider the scenario when the attacker plays deceptively while the defender does not take into account the prospect of the attacker's deception. We aim at analyzing the attacker deception decision in this no-counter-deception scenario. We assume that the attacker plays deceptively by mimicking any λ^{dec} within the range $[0, \lambda^{max}]$.[3] The value λ^{max} represents the limit to which the attacker plays deceptively. When $\lambda^{max} \to \infty$, the deception range of the attacker covers the whole range of λ. We aim at examining the impact of λ^{max} on the deception outcome of the attacker later in our experiments. Given uncertainty about the learning outcome of the defender, the attacker attempts to find the optimal $\lambda^{\text{dec}} \in [0, \lambda^{max}]$ to imitate that maximizes its utility in the worst case scenario of uncertainty, which can be formulated as follows:

$$\begin{aligned}
(\mathbf{P}^{\text{dec}}) : \max_{\lambda^{\text{dec}}} \min_{\lambda^{\text{learnt}}} \; & U^a(\mathbf{x}(\lambda^{\text{learnt}}), \lambda^{\text{dec}}) \\
\text{s.t. } & \lambda^{\text{dec}} \in [0, \lambda^{max}] \\
& \max\{\lambda^{\text{dec}} - \delta, 0\} \leq \lambda^{\text{learnt}} \leq \lambda^{\text{dec}} + \delta \\
& \mathbf{x}(\lambda^{\text{learnt}}) \in \operatorname*{argmax}_{\mathbf{x}' \in \mathbf{X}} U^d(\mathbf{x}', \lambda^{\text{learnt}})
\end{aligned}$$

[3] In this work, we consider $\lambda \geq 0$ as this is the widely accepted range of the attacker's bounded rationality in the literature.

where $\mathbf{x}(\lambda^{\text{learnt}})$ is the defender's optimal strategy w.r.t his learning outcome λ^{learnt}. The objective $U^a(\mathbf{x}(\lambda^{\text{learnt}}), \lambda^{\text{dec}})$ is essentially the attacker's utility when the defender plays $\mathbf{x}(\lambda^{\text{learnt}})$ and the attacker mimics QR with λ^{dec} to play (see Equations (1 – 3) for the detailed computation).

4.1 A Polynomial-Time Deception Algorithm

$(\mathbf{P}^{\text{dec}})$ involves three-nested optimization levels which is not straightforward to solve. We thus propose to limit the possible learning outcomes of the defender by discretizing the domain of λ^{learnt} into a finite set $\Lambda^{\text{learnt}}_{\text{discrete}} = (\lambda^{\text{learnt}}_1, \lambda^{\text{learnt}}_2, \ldots, \lambda^{\text{learnt}}_K)$ where $\lambda^{\text{learnt}}_1 = 0$, $\lambda^{\text{learnt}}_K = \lambda^{max} + \delta$, and $\lambda^{\text{learnt}}_{k+1} - \lambda^{\text{learnt}}_k = \eta, \forall k < K$ where $\eta > 0$ is the discretization step size and $K = \frac{\lambda^{max}+\delta}{\eta} + 1$ is the number of discrete learning values.[4] For each deception choice λ^{dec}, the attacker's *uncertainty set* of defender's possible learning outcomes λ^{learnt} is now given by:

$$\Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}}) = \Lambda^{\text{learnt}}_{\text{discrete}} \cap [\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta]$$

For each $\lambda^{\text{learnt}}_k$, we can easily compute the corresponding optimal defense strategy $\mathbf{x}(\lambda^{\text{learnt}}_k)$ in advance [27]. We thus obtain a simplified optimization problem:

$$(\mathbf{P}^{\text{dec}}_{\text{discrete}}): \max_{\lambda^{\text{dec}} \in [0, \lambda^{max}]} U$$
$$\text{s.t. } U \leq U^a(\mathbf{x}(\lambda^{\text{learnt}}_k), \lambda^{\text{dec}}), \text{ for all } \lambda^{\text{learnt}}_k \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}})$$

Remark on computational challenge. Although $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ is a single-level optimization, solving it is still challenging due to (i) $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ is a non-convex optimization problem since the attacker's utility $U^a(\mathbf{x}(\lambda^{\text{learnt}}_k), \lambda^{\text{dec}})$ is non-convex in λ^{dec}; and (ii) the number of inequality constraints in $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ vary with respect to λ^{dec}, which complicates the problem further. By exploiting the decomposability property of the deception space $[0, \lambda^{max}]$ and the monotonicity of the attacker's utility function $U^a(\mathbf{x}(\lambda^{\text{learnt}}_k), \lambda^{\text{dec}})$, we show that $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ can be solved optimally in a polynomial time.[5]

Theorem 1 (Time complexity). *The problem ($\mathbf{P}^{dec}_{discrete}$) can be solved optimally in a polynomial time.*

Overall, the proof of Theorem 1 is derived based on (i) Lemma 1 — showing that the deception space can be divided into an $O(K)$ number of sub-intervals, and each sub-interval leads to the same uncertainty set; and (ii) Lemma 2 — showing that $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ can be divided into a $O(K)$ sub-problems which correspond to the decomposability of the deception space (as shown in Lemma 1), and each sub-problem can be solved in polynomial time.

[4] We use a uniform discretization for the sake of solution quality analysis (as we will describe later). Our approach can be generalized to any non-uniform discretization.

[5] All of our detailed proofs are in online appendix: https://www.dropbox.com/s/frebqe6etjns6c6/appendix.pdf?dl=0.

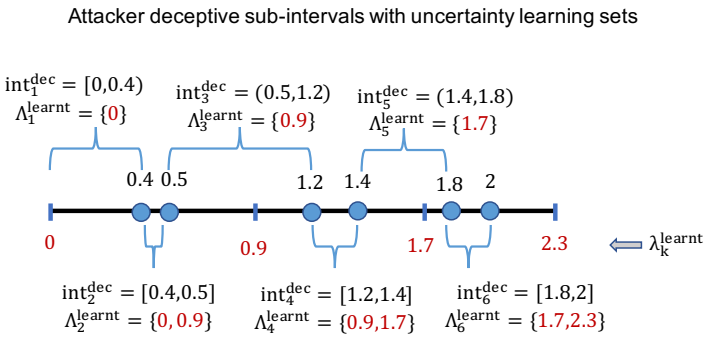


Fig. 1: An example of discretizing λ^{learnt}, $\Lambda^{\text{learnt}} = \{0, 0.9, 1.7, 2.3\}$, and the six resulting attacker sub-intervals and corresponding uncertainty sets, with $\lambda^{max} = 2, \delta = 0.5$. In particular, the first sub-interval of deceptive λ^{dec} is $int_1 = [0, 0.4)$ in which any λ^{dec} corresponds to the same uncertainty set of possible learning outcomes $\Lambda_1^{\text{learnt}} = \{0\}$.

Lemma 1 (Decomposability of deception space). *The attacker deception space $[0, \lambda^{max}]$ can be decomposed into a finite number of disjointed sub-intervals, denoted by int_j^{dec} where $j = 1, 2, \ldots,$ and $int_j^{dec} \cap int_{j'}^{dec} = \emptyset$ for all $j \neq j'$ and $\cup_j int_j^{dec} = [0, \lambda^{max}]$, such that each $\lambda^{dec} \in int_j^{dec}$ leads to the same uncertainty set of learning outcomes, denoted by $\Lambda_j^{learnt} \subseteq \Lambda_{discrete}^{learnt}$. Furthermore, these sub-intervals and uncertainty sets $(int_j^{dec}, \Lambda_j^{learnt})$ can be found in a polynomial time.*

An example of the deception-space decomposition is illustrated in Figure 1. Intuitively, although the deception space $[0, \lambda^{max}]$ is infinite, the total number of possible learning-outcome uncertainty sets is at most 2^K (i.e., the number of subsets of the discrete learning space $\Lambda_{\text{discrete}}^{\text{learnt}}$). Therefore, the deception space can be divided into a finite number of disjoint subsets such that any deception value λ^{dec} within each subset will lead to the same uncertainty set. Moreover, each of these deception subsets form a sub-interval of $[0, \lambda^{max}]$, which is derived from the following observation:

Observation 1 *Given two deception values $\lambda_1^{dec} < \lambda_2^{dec} \in [0, \lambda^{max}]$, if the learning uncertainty sets corresponding to these two values are the same, i.e., $\Lambda_{discrete}^{learnt}(\lambda_1^{dec}) \equiv \Lambda_{discrete}^{learnt}(\lambda_2^{dec})$, then for any deception value $\lambda_1^{dec} < \lambda^{dec} < \lambda_2^{dec}$, its uncertainty set is also the same, that is:*

$$\Lambda_{discrete}^{learnt}(\lambda^{dec}) \equiv \Lambda_{discrete}^{learnt}(\lambda_1^{dec}) \equiv \Lambda_{discrete}^{learnt}(\lambda_2^{dec})$$

The remaining analysis for Lemma 1 is to show that these deception sub-intervals can be found in polynomial time, which is obtained based on Observation 2:

Observation 2 *For each learning outcome λ_k^{learnt}, there are at most two deception sub-intervals such that λ_k^{learnt} is the smallest learning outcome in the corresponding learning uncertainty set. As a result, the total number of deception sub-intervals is $O(K)$, which is polynomial.*

Since there is a $O(K)$ number of deception sub-intervals, we now can develop a polynomial-time algorithm (Algorithm 1) which iteratively divides the deceptive range $[0, \lambda^{max}]$ into multiple intervals, denoted by $\{int_j^{\text{dec}}\}_j$. Each of these intervals, int_j^{dec}, corresponds to the same uncertainty set of possible learning outcomes for the defender, denoted by $\Lambda_j^{\text{learnt}}$. In this algorithm, for each $\lambda_k^{\text{learnt}}$,

Algorithm 1: Imitative behavior deception — Decomposition of QR parameter domain into sub-intervals

Input: $\Lambda^{\text{learnt}} = \{\lambda_1^{\text{learnt}}, \lambda_2^{\text{learnt}}, \dots, \lambda_K^{\text{learnt}}\}$;
Initialize interval index: $j = 1$; $start = 0$; $open = false$;
  $\Lambda_j^{\text{learnt}} = \{\lambda_k^{\text{learnt}} \in \Lambda^{\text{learnt}} : \lambda_k^{\text{learnt}} \in [start - \delta, start + \delta]\}$;
while $\Lambda_j^{learnt} \neq \emptyset$ **do**
  Set the max index: $k_j^{max} = \max_k\{\lambda_k^{\text{learnt}} \in \Lambda_j^{\text{learnt}}\}$;
  Set the min index: $k_j^{min} = \min_k\{\lambda_k^{\text{learnt}} \in \Lambda_j^{\text{learnt}}\}$;
  if $k_j^{max} < K$ & $lb_{k_j^{max}+1} \leq ub_{k_j^{min}}$ **then**
    if *open* **then** Set $int_j^{\text{dec}} = (start, lb_{k_j^{max}+1})$;
    else Set $int_j^{\text{dec}} = [start, lb_{k_j^{max}+1})$;
    Update $start = lb_{k_j^{max}+1}$; $open = false$;
    $\Lambda_{j+1}^{\text{learnt}} = \{\lambda_k^{\text{learnt}} \in \Lambda^{\text{learnt}} : \lambda_k^{\text{learnt}} \in [start - \delta, start + \delta]\}$;
  else
    if *open* **then** Set $int_j^{\text{dec}} = (start, ub_{k_j^{min}}]$;
    else Set $int_j^{\text{dec}} = [start, ub_{k_j^{min}}]$;
    Update $start = ub_{k_j^{min}}$; $open = true$;
    $\Lambda_{j+1}^{\text{learnt}} = \{\lambda_k^{\text{learnt}} \in \Lambda^{\text{learnt}} : \lambda_k^{\text{learnt}} \in (start - \delta, start + \delta]$;
  Update $j = j + 1$;
  return $\{(int_j^{\text{dec}}, \lambda_j^{\text{learnt}})\}$;

we denote by $lb_k = \lambda_k^{\text{learnt}} - \delta$ and $ub_k = \lambda_k^{\text{learnt}} + \delta$ the smallest and largest possible values of λ^{dec} so that $\lambda_k^{\text{learnt}}$ belongs to the uncertainty set of λ^{dec}. In Algorithm 1, *start* is the variable which represents the left bound of each interval int_j^{dec}. The variable *open* indicates if int_j^{dec} is left-open ($open = true$) or not ($open = false$). If *start* is known for int_j^{dec}, the uncertainty set $\Lambda_j^{\text{learnt}}$ can be determined as follows:

$$\Lambda_j^{\text{learnt}} = \{\lambda_k^{\text{learnt}} : \lambda_k^{\text{learnt}} \in [start - \delta, start + \delta]\} \text{ if } int_j^{\text{dec}} \text{ is left-closed}$$
$$\Lambda_j^{\text{learnt}} = \{\lambda_k^{\text{learnt}} : \lambda_k^{\text{learnt}} \in (start - \delta, start + \delta]\} \text{ if } int_j^{\text{dec}} \text{ is left-open}$$

Initially, *start* is set to 0 which is the lowest possible value of λ^{dec} such that the uncertainty range $[\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta]$ contains $\lambda_1^{\text{learnt}}$ and $open = false$. Given *start* and its uncertainty range $[start - \delta, start + \delta]$, the first interval int_1^{dec} of

λ^{dec} corresponds to the uncertainty set determined as follows:

$$\Lambda_1^{\text{learnt}} = \{\lambda_k^{\text{learnt}} \in \Lambda^{\text{learnt}} : \lambda_k^{\text{learnt}} \in [start - \delta, start + \delta]\}$$

At each iteration j, given the left bound $start$ and the uncertainty set $\Lambda_j^{\text{learnt}}$ of the interval int_j^{dec}, Algorithm 1 determines the right bound of int_j^{dec}, the left bound of the next interval int_{j+1}^{dec} (by updating $start$), and the uncertainty set $\Lambda_{j+1}^{\text{learnt}}$, (lines (6–15)). The correctness of Algorithm 1 is proved in the appendix.

Lemma 2 (Divide-and-conquer). *The problem ($\mathbf{P}_{discrete}^{dec}$) can be decomposed into $O(K)$ sub-problems $\{(\mathbf{P}_{\mathbf{j}}^{dec})\}$ according to the decomposibility of the deception space. Each of these sub-problems can be solved in polynomial time.*

Indeed, we can now divide the problem ($\mathbf{P}_{\text{discrete}}^{\text{dec}}$) into multiple sub-problems which correspond to the decomposition of the deception space. Essentially, each sub-problem optimizes λ^{dec} (and λ^{learnt}) over the deception sub-interval int_j^{dec} (and its corresponding uncertainty set $\Lambda_j^{\text{learnt}}$):

$$(\mathbf{P}_{\mathbf{j}}^{\text{dec}}) : \max_{\lambda^{\text{dec}} \in int_j^{\text{dec}}} U^a$$
$$\text{s.t. } U^a \leq U^a(\mathbf{x}(\lambda_k^{\text{learnt}}), \lambda^{\text{dec}}), \forall \lambda_k^{\text{learnt}} \in \Lambda_j^{\text{learnt}}$$

which maximizes the attacker's worst-case utility w.r.t uncertainty set $\Lambda_j^{\text{learnt}}$. Note that the defender strategies $\mathbf{x}(\lambda_k^{\text{learnt}})$ can be pre-computed for every outcome $\lambda_k^{\text{learnt}}$. Each sub-problem ($\mathbf{P}_{\mathbf{j}}^{\text{dec}}$) has a constant number of constraints, but still remain non-convex. Our Observation 3 shows that despite of the non-convexity, the optimal solution for ($\mathbf{P}_{\mathbf{j}}^{\text{dec}}$) is actually straightforward to compute.

Observation 3 *The optimal solution of λ^{dec} for each sub-problem, $\mathbf{P}_{\mathbf{j}}^{dec}$, is the (right) upper limit of the corresponding deception sub-interval int_j^{dec}.*

This observation is derived based on the fact that the attacker's utility, $U^a(\mathbf{x}, \lambda)$, is an increasing function of λ [18]. Therefore, in order to solve ($\mathbf{P}_{\text{discrete}}^{\text{dec}}$), we only need to iterate over right bounds of int_j^{dec} and select the best j such that the attacker's worst-case utility (i.e., the objective of ($\mathbf{P}_{\mathbf{j}}^{\text{dec}}$)), is the highest among all sub-intervals. Since there are $O(K)$ sub-problems, ($\mathbf{P}_{\text{discrete}}^{\text{dec}}$) can be solved optimally in a polynomial time, concluding our proof for Theorem 1.

4.2 Solution Quality Analysis

We now focus on analyzing the solution quality of our method presented in Section 4.1 to approximately solve the deception problem ($\mathbf{P}^{\text{dec}}$).

Theorem 2. *For any arbitrary $\epsilon > 0$, there always exists a discretization step size $\eta > 0$ such that the optimal solution of the corresponding ($\mathbf{P}_{discrete}^{dec}$) is ϵ-optimal for ($\mathbf{P}^{dec}$).*

Intuitively, let us denote by λ_*^{dec} the optimal solution of $(\mathbf{P}^{\text{dec}})$ and $U^a_{\text{worst-case}}(\lambda_*^{\text{dec}})$ is the corresponding worst-case utility of the attacker under the uncertainty of learning outcomes in $(\mathbf{P}^{\text{dec}})$. We also denote by $\lambda^{\text{dec}}_{\text{discrete}}$ the optimal solution of $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$. Then, Theorem 2 states that:

$$U^a_{\text{worst-case}}(\lambda_*^{\text{dec}}) \geq U^a_{\text{worst-case}}(\lambda^{\text{dec}}_{\text{discrete}}) \geq U^a_{\text{worst-case}}(\lambda_*^{\text{dec}}) - \epsilon$$

Heuristic to improve discretization. According to Theorem 2, we can obtain a high-quality solution for $(\mathbf{P}^{\text{dec}})$ by having a fine discretization of the learning outcome space with a small step size η. In practice, it is not necessary to have a fine discretization over the entire learning space right from the begining. Instead, we can start with a coarse discretization and solve the corresponding $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ to obtain a solution of $\lambda^{\text{dec}}_{\text{discrete}}$. We then refine the discretization *only* within the uncertainty range of the current solution, $[\lambda^{\text{dec}}_{\text{discrete}} - \delta, \lambda^{\text{dec}}_{\text{discrete}} + \delta]$. We keep doing that until the uncertainty range of the latest deception solution reaches the step-size limit which guarantees the ϵ-optimality. Practically, by doing so, we will obtain a much smaller discretized learning outcome set (aka. smaller K). As a result, the computational time for solving $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$ is substantially faster while the solution quality remains the same.

5 Defender Counter-Deception

In order to counter the attacker's imitative deception, we propose to find a counter-deception defense function $\mathcal{H} : [0, \lambda^{max} + \delta] \rightarrow \mathbf{X}$ which maps a learnt parameter λ^{learnt} to a strategy $\mathbf{x}$ of the defender. In designing an effective $\mathcal{H}$, we need to take into account that the attacker will also adapt its deception choice accordingly, denoted by $\lambda^{\text{dec}}(\mathcal{H})$. Essentially, the problem of finding an optimal defense function which maximizes the defender's utility against the attacker's deception can be abstractly represented as follows:

$$\max_{\mathcal{H}} U^d(\mathcal{H}, \lambda^{\text{dec}}(\mathcal{H}))$$

where $\lambda^{\text{dec}}(\mathcal{H})$ is the deception choice of the attacker with respect to the defense function $\mathcal{H}$ and U^d is the defender's utility corresponding to $(\mathcal{H}, \lambda^{\text{dec}}(\mathcal{H}))$. Finding an optimal $\mathcal{H}$ is challenging since the domain $[0, \lambda^{max} + \delta]$ of λ^{learnt} is continuous and there is no explicit closed-form expression of $\mathcal{H}$ as a function of λ^{learnt}. For the sake of our analysis, we divide the entire domain $[0, \lambda^{max} + \delta]$ into a number of sub-intervals $\mathbf{I} = \{I_1^d, I_2^d, \ldots, I_N^d\}$ where $I_1^d = [\lambda_1^{\text{def}}, \lambda_2^{\text{def}}]$, $I_2^d = (\lambda_2^{\text{def}}, \lambda_3^{\text{def}}]$, $\ldots$, $I_N^d = (\lambda_N^{\text{def}}, \lambda_{N+1}^{\text{def}}]$ with $0 = \lambda_1^{\text{def}} \leq \lambda_2^{\text{def}} \leq \cdots \leq \lambda_{N+1}^{\text{def}} = \lambda^{max} + \delta$, and N is the number of sub-intervals. We define a defense function with respect to the interval set: $\mathcal{H}^{\mathbf{I}} : \mathbf{I} \rightarrow \mathbf{X}$ which maps each interval $I_n^d \in \mathbf{I}$ to a single defense strategy $\mathbf{x}_n$, i.e., $\mathcal{H}^{\mathbf{I}}(I_n^d) = \mathbf{x}_n \in \mathbf{X}$, for all $n \leq N$. We denote the set of these strategies by $\mathbf{X}^{\text{def}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Intuitively, all $\lambda^{\text{learnt}} \in I_n^d$ will lead to a single strategy $\mathbf{x}_n$. Our counter-deception problem now becomes finding an optimal defense function $\mathcal{H}_* = (\mathbf{I}_*, \mathcal{H}_*^{\mathbf{I}_*})$ that comprises of (i) an optimal interval set $\mathbf{I}_*$; and (ii) corresponding defense strategies determined by the defense function $\mathcal{H}_*^{\mathbf{I}_*}$

with respect to $\mathbf{I}_*$, taking into account the attacker's deception adaptation. Essentially, $(\mathbf{I}_*, \mathcal{H}^{\mathbf{I}_*}_*)$ is the optimal solution of the following optimization problem:

$$\max_{\mathbf{I},\mathcal{H}^{\mathbf{I}}} U^d(\mathcal{H}^{\mathbf{I}}, \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})) \tag{4}$$

$$\text{s.t. } \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}) \in \operatorname*{argmax}_{\lambda^{\text{dec}} \in [0, \lambda^{max}]} \min_{\mathbf{x} \in \mathbf{X}(\lambda^{\text{dec}})} U^a(\mathbf{x}, \lambda^{\text{dec}}) \tag{5}$$

where $\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})$ is the maximin deception choice of the attacker. Here, $\mathbf{X}(\lambda^{\text{dec}}) = \{\mathbf{x}_n : I^d_n \cap [\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta] \neq \emptyset\}$ is the *uncertainty set* of the attacker when playing λ^{dec}. This uncertainty set contains all possible defense strategy outcomes with respect to the deceptive value λ^{dec}.

Main Result. So far, we have not explicitly defined the utility objective function, $U^d(\mathcal{H}^{\mathbf{I}}, \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}))$, except that we know this utility depends on the defense function $\mathcal{H}^{\mathbf{I}}$ and the attacker's deception response $\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})$. Now, since $\mathcal{H}^{\mathbf{I}}$ maps each possible learning outcome λ^{learnt} to a defense strategy, we know that if $\lambda^{\text{learnt}} \in I^d_n$, then $U^d(\mathcal{H}^{\mathbf{I}}, \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})) = U^d(\mathbf{x}_n, \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}))$, which can be computed using Equation (3). However, due to the deviation of λ^{learnt} from the attacker's deception choice, $\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})$, different possible learning outcomes λ^{learnt} within $[\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}) - \delta, \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}) + \delta]$ may belong to different intervals I^d_n (which correspond to different strategies $\mathbf{x}_n$), leading to different utility outcomes for the defender. One may argue that to cope with this deception-learning uncertainty, we can apply the maximin approach to determine the defender's worst-case utility if the defender only has the common knowledge that $\lambda^{\text{learnt}} \in [\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}) - \delta, \lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}}) + \delta]$. And perhaps, depending on any additional (private) knowledge the defender has regarding the relation between the attacker's deception and the actual learning outcome of the defender, we can incorporate such knowledge into our model and algorithm to obtain an even better utility outcome for the defender. Interestingly, we show that there is, in fact, a *universal* optimal defense function for the defender, $\mathcal{H}_*$, regardless of any additional knowledge that he may have. That is, the defender obtains the highest utility by following this defense function, and additional knowledge besides the common knowledge cannot make the defender do better. Our main result is formally stated in Theorem 3.

Theorem 3. *There is a* universal *optimal defense function, regardless of any additional information (besides the common knowledge) he has about the relation between his learning outcome and the deception choice of the attacker. Formally, let's consider the following optimization problem:*

$$\begin{aligned} (\mathbf{P}^{counter}) : \max_{\mathbf{x},\lambda} \; & U^d(\mathbf{x}, \lambda) \\ s.t. \; & U^a(\mathbf{x}, \lambda) \geq \min_{\mathbf{x}' \in \mathbf{X}} U^a(\mathbf{x}', \lambda^{max}) \\ & 0 \leq \lambda \leq \lambda^{max}, \mathbf{x} \in \mathbf{X} \end{aligned}$$

Denote by $(\mathbf{x}^*, \lambda^*)$ an optimal solution of $(\mathbf{P}^{counter})$, then an optimal solution of (4), $\mathcal{H}_*$ can be determined as follows:

- *If* $\lambda^* = \lambda^{max}$*, choose the interval set* $\mathbf{I}_* = \{I_1^d\}$ *with* $I_1^d = [0, \lambda^{max} + \delta]$ *covering the entire learning space, and function* $\mathcal{H}_*^{\mathbf{I}_*}(I_1^d) = \mathbf{x}_1$ *where* $\mathbf{x}_1 = \mathbf{x}^*$.
- *If* $\lambda^* < \lambda^{max}$*, choose the interval set* $\mathbf{I}_* = \{I_1^d, I_2^d\}$ *with* $I_1^d = [0, \lambda^* + \delta]$, $I_2^d = (\lambda^* + \delta, \lambda^{max} + \delta]$. *In addition, choose the defender strategies* $\mathbf{x}_1 = \mathbf{x}^*$ *and* $\mathbf{x}_2 \in \operatorname{argmin}_{\mathbf{x} \in \mathbf{X}} U^a(\mathbf{x}, \lambda^{max})$ *correspondingly.*

The attacker's optimal deception against this defense function is to mimic λ^*. *As a result, the defender always obtains the highest utility,* $U^d(\mathbf{x}^*, \lambda^*)$, *while the attacker receives the maximin utility of* $U^a(\mathbf{x}^*, \lambda^*)$.

Corollary 1. *When* $\lambda^{max} = +\infty$, *the defense function* $\mathcal{H}_*$ *(specified in Theorem 3) gives the defender a utility which is no less than his Strong Stackelberg equilbrium (SSE) utility.*

The proof of Corollary 1 is straightforward. Since $(\mathbf{x}^{\mathtt{sse}}, \lambda^{max} = +\infty)$ is a feasible solution of $(\mathbf{P}^{\text{counter}})$, the optimal utility of the defender $U^d(\mathbf{x}^*, \lambda^*)$ is thus no less than $U^d(\mathbf{x}^{\mathtt{sse}}, \lambda^{max})$ ($\mathbf{x}^{\mathtt{sse}}$ denotes the defender's SSE strategy).

Now the rest of this section will be devoted to prove Theorem 3. The full proof of Theorem 3 can be decomposed into three main parts: (i) We first analyze the attacker deception adapted to the defender's counter deception; (ii) Based on the result of the attacker adaptation, we provide theoretical results on computing the defender optimal defense function given a fixed set of sub-intervals $\mathbf{I}$; and (iii) finally, we complete the proof of the theorem leveraging the result in (ii).

5.1 Analyzing Attacker Deception Adaptation

In this section, we aim at understanding the behavior of the attacker deception against $\mathcal{H}^{\mathbf{I}}$. Overall, as discussed in the previous section, since the attacker is uncertain about the actual learning outcome of the defender, the attacker can attempt to find an optimal deception choice $\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})$ that maximizes its utility under the worst case of uncertainty. Essentially, $\lambda^{\text{dec}}(\mathcal{H}^{\mathbf{I}})$ is an optimal solution of the following maximin problem:

$$\max_{\lambda^{\text{dec}} \in [0, \lambda^{max}]} \min_{\mathbf{x} \in \mathbf{X}(\lambda^{\text{dec}})} U^a(\mathbf{x}, \lambda^{\text{dec}})$$

where: $\mathbf{X}(\lambda^{\text{dec}}) = \{\mathbf{x}_n : I_n^d \cap [\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta] \neq \emptyset\}$ is the *uncertainty set* of the attacker with respect to the defender's sub-intervals $\mathbf{I}$. In this problem, the uncertainty set $\mathbf{X}(\lambda^{\text{dec}})$ depends on λ^{dec} that we need to optimize, making this problem not straightforward to solve.

First, given $\mathcal{H}^{\mathbf{I}}$, we show that we can divide the range of λ^{dec} into several intervals, each interval corresponds to the same uncertainty set. This characteristic of the attacker uncertainty set is, in fact, similar to the no-counter-deception scenario as described in previous section. We propose Algorithm 2 to determine these intervals of λ^{dec}, which works in a similar fashion as Algorithm 1. The main difference is that in the presence of the defender's defense function, the

Algorithm 2: Counter-deception — Decomposition of QR parameter into sub-intervals

Input: $\mathbf{I} = \{I_1^d, I_2^d, \ldots, I_N^d\}$ and $\mathbf{X}^{def} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
Initialize attacker interval index $j = 1$;
Initialize $start = 0$; uncertainty set $\mathbf{X}_j^{\text{def}} = \{x_n : I_n^d \cap [start - \delta, start + \delta] \neq \emptyset\}$;
while $\mathbf{X}_j^{def} \neq \emptyset$ **do**
    Set the max index: $n_j^{max} = \max_n\{x_n \in \mathbf{X}_j^{\text{def}}\}$;
    Set the min index $n_j^{min} = \min_n\{x_n \in \mathbf{X}_j^{\text{def}}\}$;
    if $n_j^{max} < k$ & $lb_{n_j^{max}+1} \leq ub_{n_j^{min}+1}$ **then**
        Set $end = lb_{n_j^{max}+1}$;
    else Set $end = ub_{n_j^{min}+1}$;
    if $j = 1$ **then** Set $int_j^{\text{dec}} = [start, end]$;
    else Set $int_j^{\text{dec}} = (start, end]$;
    Update $start = end$; $j = j + 1$;
    Set $\mathbf{X}_j^{\text{def}} = \{x_n : I_n^d \cap (start - \delta, start + \delta] \neq \emptyset\}$;
return $\{int_j^{\text{dec}}, \mathbf{X}_j^{\text{def}}\}$

attacker's uncertainty set $\mathbf{X}(\lambda^{\text{dec}})$ is determined based on whether the uncertainty range of the attacker $[\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta]$ is overlapped with the defender's intervals $\mathbf{I} = \{I_n^d\}$ or not.

Essentially, similar to Algorithm 1, Algorithm 2 also iteratively divides the range of λ^{dec} into multiple intervals, (with an abuse of notation) denoted by $\{int_j^{\text{dec}}\}$. Each of these intervals, int_j^{dec}, corresponds to the same uncertainty set of $\mathbf{x}_n$, denoted by $\mathbf{X}_j^{\text{def}}$. In this algorithm, for each interval of the defender I_n^d, $lb_n = \lambda_n^{\text{def}} - \delta$ and $ub_{n+1} = \lambda_{n+1}^{\text{def}} + \delta$ represent the smallest and largest possible deceptive values of λ^{dec} so that $I_n^d \cap [\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta] \neq \emptyset$. In addition, n_j^{min} and n_j^{max} denote the smallest and largest indices of the defender's strategies in the set $\mathbf{X}^{\text{def}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ that belongs to $\mathbf{X}_j^{\text{def}}$. Algorithm 2 relies on Observations 4 and 5. Note that Algorithm 2 does not check if each interval int_j^{dec} of λ^{dec} is left-open or not since all intervals of the defender I_n^d is left-open (except for $n = 1$), making all int_j^{dec} left-closed (except for $j = 1$).

Observation 4 *Given a deceptive λ^{dec}, for any $n_1 < n_2$ such that $\mathbf{x}_{n_1}, \mathbf{x}_{n_2} \in \mathbf{X}(\lambda^{dec})$, then $\mathbf{x}_n \in \mathbf{X}(\lambda^{dec})$ for any $n_1 < n < n_2$.*

Observation 5 *For any λ^{dec} such that $lb_n < \lambda^{dec} \leq ub_{n+1}$,[6] the uncertainty range of λ^{dec} overlaps with the defender's interval I_n^d, i.e., $I_n^d \cap [\lambda^{dec} - \delta, \lambda^{dec} + \delta] \neq \emptyset$, or equivalently, $\mathbf{x}_n \in \mathbf{X}(\lambda)$. Otherwise, if $\lambda^{dec} \leq lb_n$ or $\lambda^{dec} > ub_{n+1}$, then $\mathbf{x}_n \notin \mathbf{X}(\lambda^{dec})$.*

Essentially, this algorithm divides the range of λ^{dec} into multiple intervals, (with an abuse of notation) denoted by $\{int_j^{\text{dec}}\}$. Each of these intervals, int_j^{dec},

[6] Observation 5 is stated for the general case $n > 1$ when the defender's interval I_n^d is left-open. When $n = 1$ with the left bound is included, we have $lb_n \leq \lambda^{\text{dec}} \leq ub_{n+1}$.

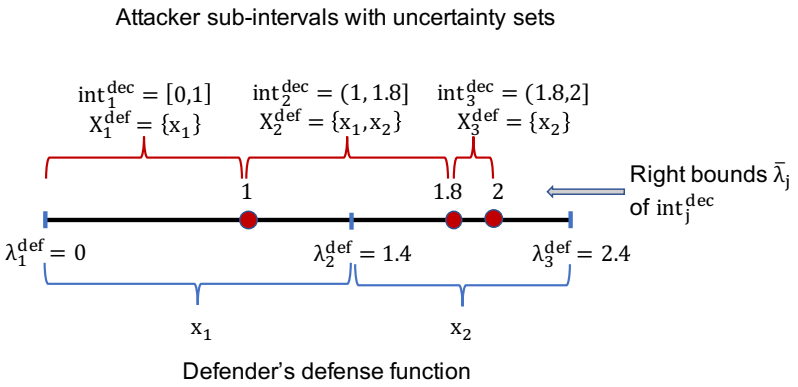


Fig. 2: An example of a defense function with corresponding sub-intervals and uncertainty sets of the attacker, where $\lambda^{max} = 2.0$ and $\delta = 0.4$. The defense function is determined as: $I_1^d = [0, 1.4]$, $I_2^d = (1.4, 2.4]$ with corresponding defense strategies $\{\mathbf{x}_1, \mathbf{x}_2\}$. Then the deception range of the attacker can be divided into three sub-intervals: $int_1^{\text{dec}} = [0, 1], int_2^{\text{dec}} = (1, 1.8], int_3^{\text{dec}} = (1.8, 2]$ with corresponding uncertainty sets $\mathbf{X}_1^{\text{def}} = \{\mathbf{x}_1\}, \mathbf{X}_2^{\text{def}} = \{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{X}_3^{\text{def}} = \{\mathbf{x}_2\}$. For example, if the attacker plays any $\lambda^{\text{dec}} \in int_2^{\text{dec}}$, it will lead the defender to play either $\mathbf{x}_1$ or $\mathbf{x}_2$, depending on the actual learning outcome of the defender.

corresponds to the same uncertainty set of $\mathbf{x}_n$, denoted by $\mathbf{X}_j^{\text{def}}$. An example of decomposing the deceptive range of λ^{dec} is shown in Figure 2.

We denote by M the number of attacker intervals. Given the division of the attacker's deception range $\{int_j^{\text{dec}}\}$, we can divide the problem of attacker deception into M sub-problems. Each corresponds to a particular int_j^{dec} where $j \in \{1, \ldots, M\}$, as follows:

$$(\bar{\mathbf{P}}_{\mathbf{j}}^{\text{dec}}) : U_j^{a,*} = \max_{\lambda^{\text{dec}} \in int_j^{\text{dec}}} \min_{x_n \in \mathbf{X}_j^{\text{def}}} U^a(x_n, \lambda^{\text{dec}})$$

Lemma 3. *For each sub-problem ($\bar{\mathbf{P}}_{\mathbf{j}}^{dec}$) with respect to the deception sub-interval int_j^{dec}, the attacker optimal deception is to imitate the right-bound of int_j^{dec}, denoted by $\bar{\lambda}_j^{dec}$.*

The proof of Lemma 3 is derived based on the fact that the attacker's utility $U^a(\mathbf{x}_n, \lambda^{\text{dec}})$ is increasing in λ^{dec}. As a result, the attacker only has to search over the right bounds, $\{\bar{\lambda}_j^{\text{dec}}\}$, of all intervals $\{int_j^{\text{dec}}\}$ to find the best one among the sub-problems that maximizes the attacker's worst-case utility. We consider these bounds $\bar{\lambda}_j^{\text{dec}}$ to be the deception candidates of the attacker. Let's assume j^{opt} is the best deception choice for the attacker among these candidates, that is, the attacker will mimic the $\bar{\lambda}_{j^{opt}}^{\text{dec}}$. We obtain the following observations about important properties of the attacker's optimal deception, which we leverage to determine an optimal defense function later.

Our following Observation 6 says that any non-optimal deception candidate for the attacker, $\bar{\lambda}_j^{\text{dec}} \neq \bar{\lambda}_{j^{opt}}^{\text{dec}}$, such that the max index of the defender strategy in the corresponding uncertainty set $\mathbf{X}_j^{\text{def}}$, denoted by n_j^{max}, satisfies $n_j^{max} \leq n_{j^{opt}}^{max}$, then the deception candidate $\bar{\lambda}_j^{\text{dec}}$ is strictly less than $\bar{\lambda}_{j^{opt}}^{\text{dec}}$, or equivalently, $j < j^{opt}$. Otherwise, j^{opt} cannot be a best deception response.

Observation 6 *For any $j \neq j^{opt}$ such that $n_j^{max} \leq n_{j^{opt}}^{max}$, then $\bar{\lambda}_j^{dec} < \bar{\lambda}_{j^{opt}}^{dec}$, or equivalently, $j < j^{opt}$.*

Note that we have right bounds of attacker intervals, denoted by $\{\bar{\lambda}_1^{\text{dec}}, \ldots, \bar{\lambda}_M^{\text{dec}} = \lambda^{max}\}$. Our next Observation 7 says that if the max index of the defender strategy $n_{j^{opt}}^{max}$ in the uncertainty set $\mathbf{X}_{j^{opt}}$ is equal to the max index of the whole defense set, N, then $\bar{\lambda}_{j^{opt}}^{\text{dec}}$ has to be equal to the highest value of the entire deception range, that is $\bar{\lambda}_{j^{opt}}^{\text{dec}} = \bar{\lambda}_M = \lambda^{max}$, or equivalently, $j^{opt} = M$.

Observation 7 *If $n_{j^{opt}}^{max} = N$, then $j^{opt} = M$.*

Remark. According to Observations 6 and 7, we can easily determine which deception choices among the set $\{\bar{\lambda}_1^{\text{dec}}, \ldots, \bar{\lambda}_M^{\text{dec}}\}$ cannot be an optimal attacker deception, regardless of defense strategies $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. These non-optimal choices are determined as follow: the deception choice $\bar{\lambda}_j$ can not be optimal for:

- Any j such that there is a $j' > j$ with $n_{j'}^{max} \leq n_j^{max}$
- Any $j < M$ such that $n_j^{max} = N$

For any other choices $\bar{\lambda}_j^{\text{dec}}$, there always exists defense strategies $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ such that $\bar{\lambda}_j^{\text{dec}}$ is an optimal attacker deception.

5.2 Finding Optimal Defense Function $\mathcal{H}^I$ Given Fixed I: Divide-and-Conquer

Given a set of sub-intervals $\mathbf{I}$, we aim at finding optimal defense function $\mathcal{H}^I$ or equivalently, strategies $\mathbf{X}^{\text{def}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ corresponding to these sub-intervals. According to previous analysis on the attacker's deception adaptation, since the attacker's best deception is one of the bounds $\{\bar{\lambda}_1^{\text{dec}}, \ldots, \bar{\lambda}_M^{\text{dec}}\}$, we propose to decompose the problem of finding an optimal defense function $\mathcal{H}^I$ into multiple sub-problems $\mathbf{P}_{\mathbf{j}}^{\text{counter}}$, each corresponds to a particular best deception choice for the attacker. In particular, for each sub-problem $\mathbf{P}_{\mathbf{j}}^{\text{counter}}$, we attempt to find $\mathcal{H}^{\mathbf{I}}$ such that $\bar{\lambda}_j^{\text{dec}}$ is the best response of the attacker. As discussed in the remark of previous section, we can easily determine which sub-problem $\mathbf{P}_{\mathbf{j}}^{\text{counter}}$ is not feasible. For any *feasible* optimal deception candidate j^{fea}, i.e., $\mathbf{P}_{\mathbf{j}^{\text{fea}}}^{\text{counter}}$ is feasible, $\mathbf{P}_{\mathbf{j}^{\text{fea}}}^{\text{counter}}$ can be formulated as follows:

$$(\mathbf{P}_{\mathbf{j}^{\text{fea}}}^{\text{counter}}) : \max_{\mathcal{H}^{\mathbf{I}}} U^d(\mathcal{H}^{\mathbf{I}}, \bar{\lambda}_{j^{\text{fea}}}^{\text{dec}})$$
$$\text{s.t.} \min_{\mathbf{x} \in \mathbf{X}_{j^{\text{fea}}}^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_{j^{\text{fea}}}^{\text{dec}}) \geq \min_{\mathbf{x} \in \mathbf{X}_j^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_j^{\text{dec}}), \forall j$$

where $U^d(\mathcal{H}^{\mathbf{I}}, \bar{\lambda}^{\text{dec}}_{j^{\text{fea}}})$ is the defender's utility when the defender commits to $\mathcal{H}^{\mathbf{I}}$ and the attacker plays $\bar{\lambda}^{\text{dec}}_{j^{\text{fea}}}$. The constraints in $(\mathbf{P}^{\text{counter}}_{\mathbf{j}^{\text{fea}}})$ guarantee that the attacker's worst-case utility for playing $\bar{\lambda}^{\text{dec}}_{j^{\text{fea}}}$ is better than playing other $\bar{\lambda}^{\text{dec}}_{j}$. Finally, our Propositions 1 and 2 determine an optimal solution for $(\mathbf{P}^{\text{counter}}_{\mathbf{j}^{\text{fea}}})$.

Proposition 1 (Sub-problem $\mathbf{P}^{\text{counter}}_{\mathbf{j}^{\text{fea}}}$). *If $n^{max}_{j^{fea}} < N$, the best defense function for the defender is determined as follows:*

- *For all $n > n^{max}_{j^{fea}}$, choose $\mathbf{x}_n = \mathbf{x}^*_>$ where $\mathbf{x}^*_>$ is an optimal solution of the following optimization problem:*

$$\min_{\mathbf{x}\in\mathbf{X}} U^a(\mathbf{x}, \lambda^{max})$$

- *For all $n \leq n^{max}_{j^{fea}}$, choose $\mathbf{x}_n = \mathbf{x}^*_<$ where $\mathbf{x}^*_<$ is the optimal solution of the following optimization problem:*

$$\begin{aligned} U^d_* = \max_{\mathbf{x}\in\mathbf{X}} & U^d(\mathbf{x}, \bar{\lambda}^{dec}_{j^{fea}}) \\ s.t.\ & U^a(\mathbf{x}, \bar{\lambda}^{dec}_{j^{fea}}) \geq U^a(\mathbf{x}^*_>, \lambda^{max}) \end{aligned}$$

By following the above defense function, an optimal deception of the attacker is to mimic $\bar{\lambda}^{dec}_{j^{fea}}$, and the defender obtains an utility of U^d_*.

Proposition 2 (Sub-problem $\mathbf{P}^{\text{counter}}_{\mathbf{j}^{\text{fea}}}$). *If $n^{max}_{j^{fea}} = N$, the best counter-deception of the defender can be determined as follows: for all n, we set: $\mathbf{x}_n = \hat{\mathbf{x}}$ where $\hat{\mathbf{x}}$ is an optimal solution of*

$$\max_{\mathbf{x}\in\mathbf{X}} U^d(\mathbf{x}, \lambda^{max})$$

By following this defense function, the attacker's best deception is to mimic λ^{max} and the defender obtains an utility of $U^d(\hat{\mathbf{x}}, \lambda^{max})$.

Based on Propositions 1 and 2, we can easily find the optimal counter-deception of the defender by choosing the solution of the sub-problem that provides the highest utility for the defender.

5.3 Completing the Proof of Theorem 3

According to Propositions 1 & 2, given an interval set $\mathbf{I}$, the resulting defense function will only lead the defender to play either $\{\mathbf{x}^*_>, \mathbf{x}^*_<\}$ or $\{\hat{\mathbf{x}}\}$, whichever provides a higher utility for the defender. Based on this result, our Theorem 3 then identifies an optimal interval set, and corresponding optimal defense strategies, as we prove below.

First, we will show that if the defender follows the defense function specified in Theorem 3, then the attacker's optimal deception is to mimic λ^*. Indeed, if $\lambda^* = \lambda^{max}$, then since the defender always plays $\mathbf{x}^*$, the attacker's optimal deception is to play $\lambda^* = \lambda^{max}$ to obtain a highest utility $U^a(\mathbf{x}^*, \lambda^{max})$.

On the other hand, if $\lambda^* < \lambda^{max}$, we consider two cases:

Case 1, if $\lambda^{max} - 2\delta \leq \lambda^* < \lambda^{max}$, then the intervals of the attackers are $int_1^{\text{dec}} = [0, \lambda^*]$ and $int_2^{\text{dec}} = (\lambda^*, \lambda^{max}]$. The corresponding uncertainty sets are $\mathbf{X}_1^{\text{def}} = \{\mathbf{x}_1\}$ and $\mathbf{X}_2^{\text{def}} = \{\mathbf{x}_1, \mathbf{x}_2\}$. In this case, the attacker's optimal deception is to mimic λ^*, since:

$$\min_{\mathbf{x} \in \mathbf{X}_1^{\text{def}}} U^a(\mathbf{x}, \lambda^*) = U^a(\mathbf{x}^*, \lambda^*)$$
$$\geq U^a(\mathbf{x}_2, \lambda^{max}) \geq \min_{\mathbf{x} \in \mathbf{X}_2^{\text{def}}} U^a(\mathbf{x}, \lambda^{max})$$

Case 2, if $\lambda^* < \lambda^{max} - 2\delta$, then the corresponding intervals for the attacker are $int_1^{\text{dec}} = [0, \lambda^*]$, $int_2^{\text{dec}} = (\lambda^*, \lambda^* + 2\delta]$, and $int_3^{\text{dec}} = (\lambda^* + 2\delta, \lambda^{max}]$. These intervals of the attacker have uncertainty sets $\mathbf{X}_1^{\text{def}} = \{\mathbf{x}_1\}$, $\mathbf{X}_2^{\text{def}} = \{\mathbf{x}_1, \mathbf{x}_2\}$, and $\mathbf{X}_3^{\text{def}} = \{\mathbf{x}_2\}$, respectively. The attacker's best deception is thus to mimic λ^*, since the attacker's worst-case utility is $\min_{\mathbf{x} \in \mathbf{X}_1^{\text{def}}} U^a(\mathbf{x}, \lambda^*) = U^a(\mathbf{x}^*, \lambda^*)$, and

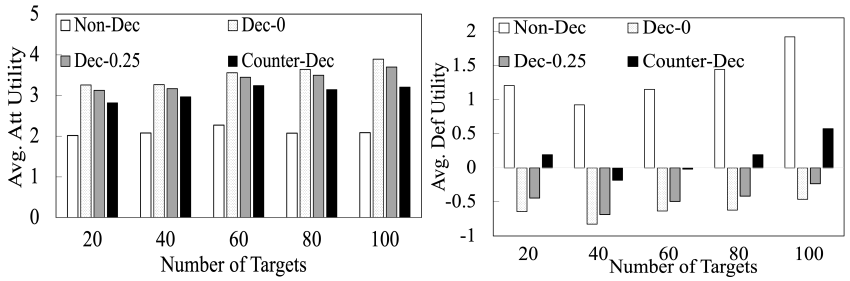
$$U^a(\mathbf{x}^*, \lambda^*) \geq U^a(\mathbf{x}_2, \lambda^{max}) \geq \min_{\mathbf{x} \in \mathbf{X}_2} U^a(\mathbf{x}, \lambda^* + 2\delta)$$
$$U^a(\mathbf{x}^*, \lambda^*) \geq U^a(\mathbf{x}_2, \lambda^{max}) = \min_{\mathbf{x} \in \mathbf{X}_3} U^a(\mathbf{x}, \lambda^{max})$$

Now, since the attacker's best deception is to mimic λ^*, according to the above analysis, the uncertainty set is $\mathbf{X}_1^{\text{def}} = \{\mathbf{x}_1 = \mathbf{x}^*\}$, thus the defender will play $\mathbf{x}^*$ in the end, leading to an utility of $U^d(\mathbf{x}^*, \lambda^*)$. This is the highest possible utility that the defender can obtain since both optimization problems presented in Propositions 1 and 2 are special cases of ($\mathbf{P}^{\text{counter}}$) when we fix the variable $\lambda = \lambda^{max}$ (for Proposition 2) or $\lambda = \bar{\lambda}_{j^{\text{fea}}}$ (for Proposition 1).

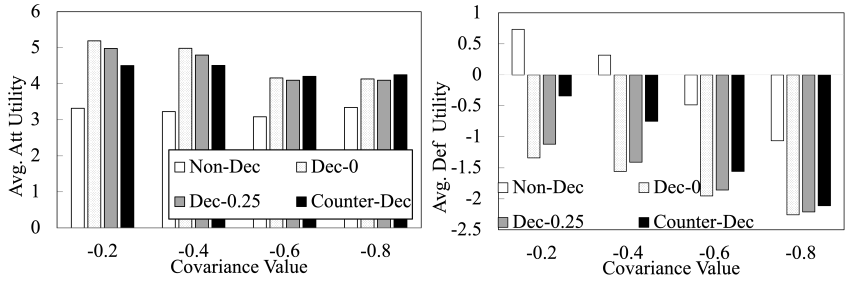
6 Experimental Evaluation

Our experiments are run on a 2.8 GHz Intel Xeon processor with 256 GB RAM. We use `Matlab` (https://www.mathworks.com) to solve non-linear programs and `Cplex` (https://www.ibm.com/analytics/cplex-optimizer) to solve `MILP`s involved in the evaluated algorithms. We use a value of $\lambda^{max} = 5$ in all our experiments (except in Figures 3(g)(h)), and discretize the range $[0, \lambda^{max}]$ using a step size of 0.2: $\lambda \in \{0, 0.2, \ldots, \lambda^{max}\}$. We use the covariance game generator, `GAMUT` (http://gamut.stanford/edu) to generate rewards and penalties of players within the range of $[1, 10]$ (for attacker) and $[-10, -1]$ (for defender). GAMUT takes as input a covariance value $r \in [-1, 0]$ which controls the correlations between the defender and the attacker's payoff. Our results are averaged over 50 runs. All our results are statistically significant under bootstrap-t ($p = 0.05$).

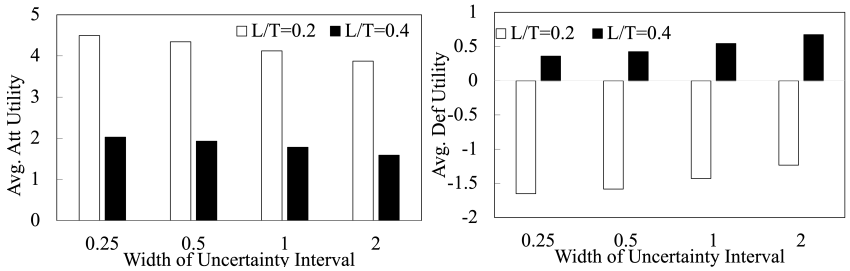
Algorithms. We compare three cases: (i) `Non-Dec`: the attacker is non deceptive and the defender also assumes so. As a result, both play Strong Stackelberg equilibrium strategies; (ii) `Dec-`δ: the attacker is deceptive, while the defender does not handle the attacker's deception (Section 4). We examine different uncertainty ranges by varying values of δ; and (iii) `Dec-Counter`: the attacker is deceptive while the defender tackle the attacker's deception (Section 5).



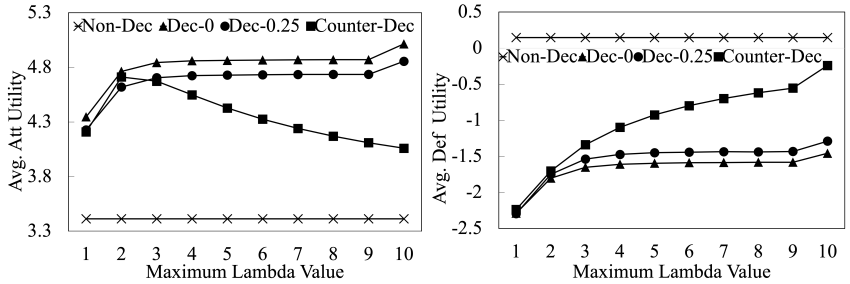
(a) Attacker utility, vary #target (b) Defender utility, vary #target



(c) Att. utility, vary covariance (d) Def. utility, vary covariance



(e) Att. utility, vary interval (f) Def. utility, vary interval



(g) Att. utility, vary λ^{max} (h) Def. utility, vary λ^{max}

Fig. 3: Evaluations on player utilities

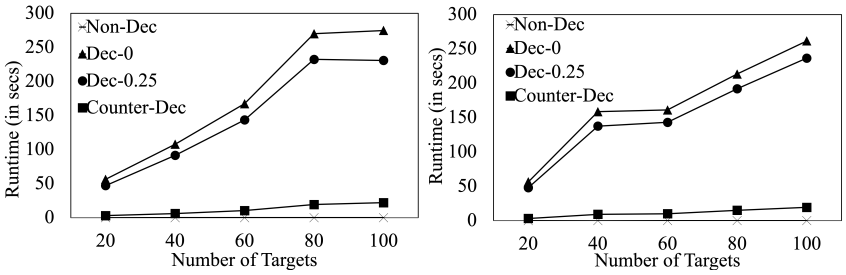
Figures 3(a)(b) compare the performance of our algorithms with increasing number of targets. These figures show that (i) the attacker benefits by playing deceptively (`Dec-0` achieves 61% higher attacker utility than `Non-Dec`); (ii) the benefit of deception to the attacker is reduced when the attacker is uncertain about the defender's learning outcome. In particular, `Dec-0.25` achieves 4% lesser attacker utility than `Dec-0`; (iii) the defender suffers a substantial utility loss due to the attacker's deception and this utility loss is reduced in the presence of the attacker's uncertainty; and finally, (iv) the defender benefits significantly (in terms of his utility) by employing counter-deception against a deceptive attacker.

In Figures 3(c)(d), we show the performance of our algorithms with varying r (i.e., covariance) values. In zero-sum games (i.e., $r = -1$), the attacker has no incentive to be deceptive [18]. Therefore, we only plot the results of $r \in [-0.2, -0.8]$ with a step size of 0.2. This figure shows that when r gets closer to -1.0 (which implies zero-sum behavior), the attacker's utility with deception (i.e., `Dec-0` and `Dec-0.25`) gradually moves closer to its utility with `Non-Dec`, reflecting that the attacker has less incentive to play deceptively. Furthermore, the defender's average utility in all cases gradually decreases when the covariance value gets closer to -1.0. This results show that in `SSG`s, the defender's utility is always governed by the *adversarial* level (i.e., the payoff correlations) between the players, regardless of whether the attacker is deceptive or not.

Figure 3(e)(f) compare the attacker and defender utilities with varying uncertainty range, i.e., δ values, on 60-target games. These figures show that attacker utilities decrease linearly with increasing values of δ. On the other hand, defender utilities increase linearly with increasing values of δ. This is reasonable as increasing δ corresponds to a greater width of the uncertainty interval that the attacker has to contend with. This increased uncertainty forces the attacker to play more conservatively, thereby leading to decreased utilities for the attacker and increased utilities for the defender.

In Figures 3(g)(h), we analyze the impact of varying λ^{max} on the players' utilities in 60-target games. These figures show that (i) with increasing values of λ^{max}, the action space of a deceptive attacker increases, hence, the attacker utility increases as a result (`Dec-0`, `Dec-0.25` in both sub-figures); (ii) When this λ^{max} is close to zero, the attacker is limited to a less-strategic-attack zone and thus the defender's strategies have less influence on how the attacker would response. The defender thus receives a lower utility when λ^{max} gets close to zero; and (iii) most importantly, the attacker utility against a counter-deceptive defender decreases with increasing values of λ^{max}. This result shows that when the defender plays counter-deception, the attacker can actually gain more benefit by committing to a more limited deception range.

Finally, we evaluate the runtime performance of our algorithms in Figure 4. We provide results for resource-to-target ratio $\frac{L}{T} = 0.3$ and 0.5. This figure shows that (i) even on 100 target games, `Dec-0` finishes in $\sim$5 minutes. (ii) Due to the simplicity of the proposed counter-deception algorithm, `Counter-Dec` finishes in 13 seconds on 100 target games.



(a) Runtime, ratio = 0.3 (b) Runtime, ratio = 0.5

Fig. 4: Runtime performance

7 Summary

This paper provides a comprehensive analysis of the attacker deception and defender counter-deception under uncertainty. Our algorithms are developed based on the decomposibility of the attacker's deception space and the discretization of the defender's learning outcome. Our key finding is that the optimal counter-deception defense solution only depends on the common knowledge of players about the uncertainty range of the defender's learning outcome. Finally, our extensive experiments show the effectiveness of our counter-deception solutions in handling the attacker's deception.

References

1. An, B., Shieh, E., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G.: A deployed quantal response based patrol planning system for the us coast guard. In Interfaces **43**(5), 400–420 (2013)
2. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
3. Carroll, T.E., Grosu, D.: A game theoretic investigation of deception in network security. Security and Communication Networks **4**(10), 1162–1172 (2011)
4. Estornell, A., Das, S., Vorobeychik, Y.: Deception through half-truths. In: AAAI (2020)
5. Fraunholz, D., Anton, S.D., Lipps, C., Reti, D., Krohmer, D., Pohl, F., Tammen, M., Schotten, H.D.: Demystifying deception technology: A survey. arXiv preprint arXiv:1804.06196 (2018)
6. Fugate, S., Ferguson-Walter, K.: Artificial intelligence and game theory models for defending critical networks with cyber deception **40**, 49–62 (Mar 2019). https://doi.org/10.1609/aimag.v40i1.2849, https://www.aaai.org/ojs/index.php/aimagazine/article/view/2849
7. Gan, J., Xu, H., Guo, Q., Tran-Thanh, L., Rabinovich, Z., Wooldridge, M.: Imitative follower deception in stackelberg games. arXiv preprint arXiv:1903.02917 (2019)
8. Gholami, S., Yadav, A., Tran-Thanh, L., Dilkina, B., Tambe, M.: Don't put all your strategies in one basket: Playing green security games with imperfect prior knowledge. In: AAMAS. pp. 395–403. AAMAS (2019)

9. Guo, Q., An, B., Bosansky, B., Kiekintveld, C.: Comparing strategic secrecy and Stackelberg commitment in security games. In: IJCAI (2017)
10. Han, X., Kheir, N., Balzarotti, D.: Deception techniques in computer security: A research perspective. ACM Computing Surveys (CSUR) **51**(4), 1–36 (2018)
11. Horák, K., Zhu, Q., Bošanskỳ, B.: Manipulating adversary's belief: A dynamic game approach to deception by design for proactive network security. In: GameSec. pp. 273–294. Springer (2017)
12. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.D.: Adversarial machine learning. In: AISec. pp. 43–58. ACM (2011)
13. Kar, D., Nguyen, T.H., Fang, F., Brown, M., Sinha, A., Tambe, M., Jiang, A.X.: Trends and applications in stackelberg security games. Handbook of Dynamic Game Theory pp. 1–47 (2017)
14. Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., Tambe, M.: Computing optimal randomized resource allocations for massive security games. In: AAMAS. pp. 689–696. AAMAS (2009)
15. McFadden, D., et al.: Conditional logit analysis of qualitative choice behavior (1973)
16. McKelvey, R.D., Palfrey, T.R.: Quantal response equilibria for normal form games. Games and economic behavior **10**(1), 6–38 (1995)
17. Nguyen, T.H., Sinha, A., He, H.: Partial adversarial behavior deception in security games. In: IJCAI (2020)
18. Nguyen, T.H., Vu, N., Yadav, A., Nguyen, U.: Decoding the imitation security game: Handling attacker imitative behavior deception. In: ECAI (2020)
19. Nguyen, T.H., Wang, Y., Sinha, A., Wellman, M.P.: Deception in finitely repeated security games. In: AAAI (2019)
20. Nguyen, T.H., Yang, R., Azaria, A., Kraus, S., Tambe, M.: Analyzing the effectiveness of adversary modeling in security games. In: AAAI (2013)
21. Rabinovich, Z., Jiang, A.X., Jain, M., Xu, H.: Information disclosure as a means to security. In: AAMAS. pp. 645–653 (2015)
22. Sinha, A., Fang, F., An, B., Kiekintveld, C., Tambe, M.: Stackelberg security games: Looking beyond a decade of success. In: IJCAI. pp. 5494–5501 (2018)
23. Steinhardt, J., Koh, P.W.W., Liang, P.S.: Certified defenses for data poisoning attacks. In: NeurIPS. pp. 3517–3529 (2017)
24. Tambe, M.: Security and game theory: algorithms, deployed systems, lessons learned. Cambridge university press (2011)
25. Tong, L., Yu, S., Alfeld, S., et al.: Adversarial regression with multiple learners. In: ICML. pp. 4946–4954 (2018)
26. Xu, H., Rabinovich, Z., Dughmi, S., Tambe, M.: Exploring information asymmetry in two-stage security games. In: AAMAS. pp. 1057–1063 (2015)
27. Yang, R., Kiekintveld, C., Ordonez, F., Tambe, M., John, R.: Improving resource allocation strategy against human adversaries in security games. In: IJCAI (2011)
28. Zhuang, J., Bier, V.M., Alagoz, O.: Modeling secrecy and deception in a multiple-period attacker–defender signaling game. European Journal of Operational Research **203**(2), 409–418 (2010)

Appendix A. Remaining proofs for Theorem 1

Appendix A.1: Proof of Observation 1

For any $\lambda^{\text{learnt}} \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \equiv \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}})$, we have:

$$\lambda_1^{\text{dec}} - \delta \leq \lambda^{\text{learnt}} \leq \lambda_1^{\text{dec}} + \delta$$
$$\lambda_2^{\text{dec}} - \delta \leq \lambda^{\text{learnt}} \leq \lambda_2^{\text{dec}} + \delta$$

Since $\lambda^{\text{dec}} \in (\lambda_1^{\text{dec}}, \lambda_2^{\text{dec}})$, we obtain:

$$\lambda^{\text{dec}} - \delta \leq \lambda^{\text{learnt}} \leq \lambda^{\text{dec}} + \delta$$

which implies $\lambda^{\text{learnt}} \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}})$. As a result,

$$\Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \equiv \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}}) \subseteq \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}}) \qquad (*)$$

On the other hand, let's consider a $\lambda^{\text{learnt}} \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}})$, or equivalently, $\lambda^{\text{dec}} - \delta \leq \lambda^{\text{learnt}} \leq \lambda^{\text{dec}} + \delta$. We are going to show that this $\lambda^{\text{learnt}} \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \equiv \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}})$ as well. Indeed, let's assume $\lambda^{\text{learnt}} \notin \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \equiv \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}})$. It means the following inequalities must hold true:

$$\lambda_1^{\text{dec}} + \delta < \lambda^{\text{learnt}} < \lambda_2^{\text{dec}} - \delta$$

which means that the uncertainty ranges with respect to λ_1^{dec} and λ_2^{dec} are not overlapped, i.e.,

$$[\lambda_1^{\text{dec}} - \delta, \lambda_1^{\text{dec}} + \delta] \cap [\lambda_2^{\text{dec}} - \delta, \lambda_2^{\text{dec}} + \delta] \equiv \emptyset$$

or equivalently, $\Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \cap \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}}) \equiv \emptyset$, which is contradictory. Therefore, $\lambda^{\text{learnt}} \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \equiv \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}})$, meaning that:

$$\Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}}) \subseteq \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_1^{\text{dec}}) \equiv \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda_2^{\text{dec}}) \qquad (**)$$

The combination of (*) and (**) concludes our proof.

Appendix A.2: Proof of Observation 2

First, although the deception space $[0, \lambda^{max}]$ is infinite, the total number of possible learning-outcome uncertainty sets is at most 2^K (i.e., the number of subsets of the discrete learning space $\Lambda^{\text{learnt}}_{\text{discrete}}$). Therefore, the deception space can be divided into a finite number of disjoint subsets such that any deception value λ^{dec} within each subset will lead to the same uncertainty set. Moreover, each of these deception subsets form a sub-interval of $[0, \lambda^{max}]$, which is a result of Observation 1.

Now, in order to prove that the number of disjoint sub-intervals is $O(K)$, we will show that for each learning outcome $\lambda_k^{\text{learnt}}$, there are at most two deception sub-intervals such that $\lambda_k^{\text{learnt}}$ is the smallest learning outcome in the

corresponding learning uncertainty set. Let's assume there is a deception sub-interval $[\lambda_1^{\text{dec}}, \lambda_2^{\text{dec}}]$ which leads to an uncertainty set $\{\lambda_k^{\text{learnt}}, \lambda_{k+1}^{\text{learnt}}, \ldots, \lambda_{k'}^{\text{learnt}}\}$ for some $k' \geq k$. We will prove that the following inequalities must hold:

$$\frac{2\delta}{\eta} - 2 < k' - k \leq \frac{2\delta}{\eta} \tag{6}$$

where η is the discretization step size. Indeed, for any $\lambda^{\text{dec}} \in [\lambda_1^{\text{dec}}, \lambda_2^{\text{dec}}]$, we have:

$$\begin{aligned}
&\lambda^{\text{dec}} - \delta \leq \lambda_k^{\text{learnt}} \leq \lambda^{\text{dec}} + \delta \\
&\lambda^{\text{dec}} - \delta \leq \lambda_{k'}^{\text{learnt}} \leq \lambda^{\text{dec}} + \delta \\
&\lambda_{k-1}^{\text{learnt}} < \lambda^{\text{dec}} - \delta \text{ and } \lambda_{k'+1}^{\text{learnt}} > \lambda^{\text{dec}} + \delta
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\lambda_{k'}^{\text{learnt}} - \lambda_k^{\text{learnt}} \leq 2\delta \implies k' - k \leq \frac{2\sigma}{\eta} \\
&\lambda_{k'+1}^{\text{learnt}} - \lambda_{k-1}^{\text{learnt}} > 2\delta \implies k' - k > \frac{2\sigma}{\eta} - 2
\end{aligned}$$

which concludes (6). Now, according to (6), for every k, then $k' = k + \lceil \frac{2\sigma}{\eta} \rceil - 2$ or $k' = k + \lfloor \frac{2\sigma}{\eta} \rfloor$, which means that there are at most two deception sub-intervals such that $\lambda_k^{\text{learnt}}$ is the smallest learning outcome in their learning uncertainty sets.

Appendix A.3: Imitative Behavior Deception: Correctness of Algorithm 1 for Decomposing Deception Range

Finally, we prove the correctness of Algorithm 1 by presenting Proposition 3, which shows that for any λ^{dec} within each interval int_j^{dec}, the corresponding uncertainty interval $[\lambda^{\text{dec}} - \delta, \lambda^{\text{dec}} + \delta]$ covers the same uncertainty set $\Lambda_j^{\text{learnt}}$.

Proposition 3. *Each iteration j of Algorithm 1 returns an interval int_j^{dec} such that each $\lambda^{dec} \in int_j^{dec}$ leads to the same uncertainty set:*

$$\Lambda_j^{learnt} = \{\lambda_{k_j^{min}}^{learnt}, \ldots, \lambda_{k_j^{max}}^{learnt}\}$$

Proof. At each iteration j, Algorithm 1 considers two cases:

Case 1: $k_j^{max} < K$ and $lb_{k_j^{max}+1} \leq ub_{k_j^{min}}$. In this case, the interval int_j^{dec} is determined as follows:

$$\begin{aligned}
&int_j^{\text{dec}} = [start, lb_{k_j^{max}+1}) \text{ if } open = false \\
&int_j^{\text{dec}} = (start, lb_{k_j^{max}+1}) \text{ if } open = true
\end{aligned}$$

Note that, since $\Lambda_j^{\text{learnt}}$ is the uncertainty set of *start* with the smallest and largest indices of (k_j^{min}, k_j^{max}), we have: $lb_{k_j^{min}} \leq lb_{k_j^{max}} \leq start$ and $ub_{k_j^{min}-1} < start$. Therefore, for any $\lambda^{\text{dec}} \in int_j^{\text{dec}}$, we obtain:

$$lb_{k_j^{min}} \leq start \leq \lambda^{\text{dec}} \text{ and } \lambda^{\text{dec}} < lb_{k_j^{max}+1} \leq ub_{k_j^{min}}$$
$$lb_{k_j^{max}} \leq start \leq \lambda^{\text{dec}} \text{ and } \lambda < ub_i \leq ub_{k_j^{max}}$$
$$\lambda^{\text{dec}} < lb_{k_j^{max}+1} \text{ and } \lambda^{\text{dec}} \geq start > ub_{k_j^{min}-1}$$

which means $\lambda_{k_j^{min}}^{\text{learnt}}$ and $\lambda_{k_j^{max}}^{\text{learnt}}$ belongs to the uncertainty set of λ^{dec} while $\lambda_{k_j^{min}-1}^{\text{learnt}}$ and $\lambda_{k_j^{max}+1}^{\text{learnt}}$ do not. Thus, $\Lambda_j^{\text{learnt}}$ is the uncertainty set of λ^{dec}. Since int_j^{dec} is open-right, the left bound of int_{j+1}^{dec} is $start = lb_{m_j+1}$ and $open = false$, and $\Lambda_{j+1}^{\text{learnt}}$ is determined accordingly.

Case 2: $k_j^{max} = K$ or $lb_{k_j^{max}+1} > ub_{k_j^{min}}$. In this case, the interval int_j^{dec} is determined as follows:

$$int_j^{\text{dec}} = [start, ub_{k_j^{min}}] \text{ if } open = false$$
$$int_j^{\text{dec}} = (start, ub_{k_j^{min}}] \text{ if } open = true$$

The argument for this case is similar. For the sake of analysis, since $k_j^{max} = K$ which is the largest index of λ^{learnt} in the entire set Λ^{learnt}, we set $lb_{k_j^{max}+1} = \infty$. For any $\lambda^{\text{dec}} \in int_j^{\text{dec}}$, we have:

$$lb_{k_j^{min}} \leq start \leq \lambda^{\text{dec}} \leq ub_{k_j^{min}}$$
$$lb_{k_j^{max}} \leq start \leq \lambda^{\text{dec}} \leq ub_{k_j^{min}} \leq ub_{k_j^{max}}$$
$$\lambda^{\text{dec}} \leq ub_{k_j^{min}} < lb_{k_j^{max}+1} \text{ and } \lambda^{\text{dec}} \geq start > ub_{k^{min}-1}$$

which implies $\Lambda_j^{\text{learnt}}$ is the uncertainty set of λ^{dec}. Since int_j^{dec} is closed-right, the left bound of int_{j+1}^{dec} is $start = ub_{k_j^{min}}$ and $open = true$, concluding our proof.

Appendix B: Proof of Theorem 2

Let's denote by λ_*^{dec} the optimal solution of $(\mathbf{P}^{\text{dec}})$. Then the worst-case utility of the attacker is determined as follows:

$$U^{\text{worst}}(\lambda_*^{\text{dec}}) = \min_{\lambda^{\text{learnt}} \in [\lambda_*^{\text{dec}} - \delta, \lambda_*^{\text{dec}} + \delta]} U^a(\mathbf{x}(\lambda^{\text{learnt}}), \lambda_*^{\text{dec}})$$

On the other hand, let's denote by $\lambda_{\text{discrete}}^{\text{dec}}$ the optimal solution of $(\mathbf{P}_{\text{discrete}}^{\text{dec}})$. Then the discretized worst-case utility of the attacker is determined as follows:

$$U_{\text{discrete}}^{\text{worst}}(\lambda_{\text{discrete}}^{\text{dec}}) = \min_{\lambda^{\text{learnt}} \in \Lambda_{\text{discrete}}^{\text{learnt}}(\lambda_{\text{discrete}}^{\text{dec}})} U^a(\mathbf{x}(\lambda^{\text{learnt}}), \lambda_{\text{discrete}}^{\text{dec}})$$

Note that, $U^{\text{worst}}_{\text{discrete}}(\lambda^{\text{dec}}_{\text{discrete}})$ is not the actual worst-case utility of the attacker for mimicking $\lambda^{\text{dec}}_{\text{discrete}}$ since it is computed based on the discrete uncertainty set, rather than the original continuous uncertainty set. In fact, the actual attacker worst-case utility is $U^{\text{worst}}(\lambda^{\text{dec}}_{\text{discrete}})$. We will show that for any $\epsilon > 0$, there exists a discretization step size η such that:

$$U^{\text{worst}}(\lambda^{\text{dec}}_{*}) \geq U^{\text{worst}}(\lambda^{\text{dec}}_{\text{discrete}}) \geq U^{\text{worst}}(\lambda^{\text{dec}}_{*}) - \epsilon \tag{7}$$

Observe that the first inequality is easily obtained since λ^{dec}_{*} the optimal solution of $(\mathbf{P}^{\text{dec}})$. Therefore, we will focus on the second inequality. First, we obtain the following inequalities:

$$U^{\text{worst}}(\lambda^{\text{dec}}_{*}) \leq U^{\text{worst}}_{\text{discrete}}(\lambda^{\text{dec}}_{*}) \leq U^{\text{worst}}_{\text{discrete}}(\lambda^{\text{dec}}_{\text{discrete}})$$

The first inequality is obtained based on the fact that the discretized uncertainty set is a subset of the actual continuous uncertainty range $\Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}}_{*}) \subset [\lambda^{\text{dec}}_{*} - \delta, \lambda^{\text{dec}}_{*} + \delta]$. The second inequality is derived from the fact that $\lambda^{\text{dec}}_{\text{discrete}}$ is the optimal solution of $(\mathbf{P}^{\text{dec}}_{\text{discrete}})$. Therefore, in order to obtain the second inequality of (7), we are going to prove that for any $\epsilon > 0$, there exists $\eta > 0$ such that:

$$U^{\text{worst}}(\lambda^{\text{dec}}_{\text{discrete}}) + \epsilon \geq U^{\text{worst}}_{\text{discrete}}(\lambda^{\text{dec}}_{\text{discrete}}) \tag{8}$$

Let's denote by $\lambda^{\text{learnt}}_{*}$ the worst-case learning outcome with respect to $\lambda^{\text{dec}}_{\text{discrete}}$ within the uncertainty range $[\lambda^{\text{dec}}_{\text{discrete}} - \delta, \lambda^{\text{dec}}_{\text{discrete}} + \delta]$. That is,

$$U^{\text{worst}}(\lambda^{\text{dec}}_{\text{discrete}}) = U^{a}(\mathbf{x}(\lambda^{\text{learnt}}_{*}), \lambda^{\text{dec}}_{\text{discrete}})$$

Since $\Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}}_{\text{discrete}})$ is a discretization of $[\lambda^{\text{dec}}_{\text{discrete}} - \delta, \lambda^{\text{dec}}_{\text{discrete}} + \delta]$, there exist a $\lambda^{\text{learnt}}_{k} \in \Lambda^{\text{learnt}}_{\text{discrete}}(\lambda^{\text{dec}}_{\text{discrete}})$ such that $|\lambda^{\text{learnt}}_{k} - \lambda^{\text{learnt}}_{*}| \leq \eta$. Now, according to the definition of the discretized worst-case utility of the attacker, we have:

$$U^{\text{worst}}_{\text{discrete}}(\lambda^{\text{dec}}_{\text{discrete}}) \leq U^{a}(\mathbf{x}(\lambda^{\text{learnt}}_{k}), \lambda^{\text{dec}}_{\text{discrete}})$$

Therefore, proving (8) now induces to proving $\exists\eta$:

$$U^{a}(\mathbf{x}(\lambda^{\text{learnt}}_{k}), \lambda^{\text{dec}}_{\text{discrete}}) - U^{a}(\mathbf{x}(\lambda^{\text{learnt}}_{*}), \lambda^{\text{dec}}_{\text{discrete}}) \leq \epsilon$$

where $|\lambda^{\text{learnt}}_{k} - \lambda^{\text{learnt}}_{*}| \leq \eta$. First, according to [17], for any λ, the defender's corresponding optimal strategy $\mathbf{x}(\lambda)$ is a differentiable function of λ. Second, the attacker's utility $U^{a}(\mathbf{x}, \lambda^{\text{dec}}_{\text{discrete}})$ is a differentiable function of the defender's strategy $\mathbf{x}$ for any $\lambda^{\text{dec}}_{\text{discrete}}$. Therefore, $U^{a}(\mathbf{x}(\lambda), \lambda^{\text{dec}}_{\text{discrete}})$ is differentiable (and thus continuous) at λ. According to the continuity property, for any $\epsilon > 0$, there always exists $\eta > 0$ such that:

$$U^{a}(\mathbf{x}(\lambda), \lambda^{\text{dec}}_{\text{discrete}}) - U^{a}(\mathbf{x}(\lambda^{\text{learnt}}_{*}), \lambda^{\text{dec}}_{\text{discrete}}) \leq \epsilon$$

for all λ such that $|\lambda - \lambda^{\text{learnt}}_{*}| \leq \eta$, concluding our proof.

Appendix C: Proof of Theorem 3

We first provide a detailed computation and analysis on the attacker's deception response and then the optimal defense function $\mathcal{H}^{\mathbf{I}}$ given a *fixed* set of intervals $\mathbf{I}$. We leverages these results to complete the proof of Theorem 3 at the end.

Appendix C.1: Analyzing Attacker Deception Adaptation

Proof of Observation 6 Observation 6 can be proved by contradiction as follows. Let's assume if there is $j > j^{opt}$ such that $n_j^{max} \leq n_{j^{opt}}^{max}$. According to Algorithm 2, for any attacker interval indices $j > j'$, we have the min and max indices of the defender's strategies in corresponding uncertainty sets must satisfy: $n_j^{min} \geq n_{j'}^{min}$ and $n_j^{max} \geq n_{j'}^{max}$, and they can not be both equal. That is because the intervals $\{int_j^{\text{dec}}\}$ returned by Algorithm 2 are sorted in a strictly increasing order. Therefore, if there is $j > j^{opt}$ such that $n_j^{max} \leq n_{j^{opt}}^{max}$, it means $n_j^{min} > n_{j^{opt}}^{min}$ and $n_j^{max} = n_{j^{opt}}^{max}$. In other words, the uncertainty set $\mathbf{X}_j^{\text{def}} \subset \mathbf{X}_{j^{opt}}^{def}$. Thus, we have the attacker's optimal worst-case utility with respect to deception intervals j and j^{opt} must satisfy:

$$U_{j^{opt}}^{a,*} = \min_{\mathbf{x} \in \mathbf{X}_{j^{opt}}^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_{j^{opt}}^{\text{dec}}) \leq \min_{\mathbf{x} \in \mathbf{X}_j^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_{j^{opt}}^{\text{dec}})$$
$$< \min_{\mathbf{x} \in \mathbf{X}_j^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_j^{\text{dec}}) = U_j^{a,*}$$

since $U^a(\mathbf{x}, \lambda)$ is a strictly increasing function of λ.[7]This strict inequality shows that j^{opt} cannot be an optimal deception for the attacker, concluding our proof for Observation 6.

Note that we have right bounds of attacker intervals, denoted by $\{\bar{\lambda}_1^{\text{dec}}, \ldots, \bar{\lambda}_M^{\text{dec}} = \lambda^{max}\}$.

Proof of Observation 7 We also prove this observation using contradiction. Let's assume that $j^{opt} < M$. Again, according to Algorithm 2, for any $j > j'$, we have $n_j^{min} \geq n_{j'}^{min}$ and $n_j^{max} \geq n_{j'}^{max}$, and they can not be both equal. Therefore, if $n_{j^{opt}}^{max} = N$, then for all $j > j^{opt}$, we have: $n_j^{max} = N$ and $n_j^{min} > n_{j^{opt}}^{min}$, which means $\mathbf{X}_j^{\text{def}} \subset \mathbf{X}_{j^{opt}}^{\text{def}}$. Therefore, if $j^{opt} < M$, then we obtain:

$$U_{j^{opt}}^{a,*} = \min_{\mathbf{x} \in \mathbf{X}_{j^{opt}}^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_{j^{opt}}^{\text{dec}}) \leq \min_{\mathbf{x} \in \mathbf{X}_M^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_{j^{opt}}^{\text{dec}})$$
$$< \min_{\mathbf{x} \in \mathbf{X}_M^{\text{def}}} U^a(\mathbf{x}, \bar{\lambda}_M^{\text{dec}}) = U_M^{a,*}$$

[7]There is a degenerate case in which $U^a(\mathbf{x}, \lambda)$ is constant for all λ, when the defense strategy $\mathbf{x}$ leads to an identical expected utility for the attacker across all targets. To avoid this case, we can add a small noise to such defense strategy $\mathbf{x}$ so that these attacker expected utilities vary across the targets, while ensuring that this noise only leads to a small change in the defender's utility.

which shows that j^{opt} cannot be an optimal deception of the attacker, concluding the proof of Observation 7.

Appendix C.2: Finding Optimal Defense Function $\mathcal{H}^{\mathbf{I}}$ Given Fixed I: Divide-and-Conquer

Proof of Proposition 1 First, we show that the attacker optimal deception response is to $\bar{\lambda}^{\text{dec}}_{j^{\text{fea}}}$. Indeed, we have the uncertainty set $\mathbf{X}^{\text{def}}_{j^{\text{fea}}} \equiv \{\mathbf{x}^*_<\}$ because the defender plays $\mathbf{x}_n = \mathbf{x}^*_<$ for all $n \leq n^{max}_{j^{\text{fea}}}$. In addition, for all j such that $n^{max}_j > n^{max}_{j^{\text{fea}}}$, the uncertainty set $\mathbf{X}^{\text{def}}_j$ contains $\mathbf{x}^*_>$. Therefore, we have the attacker worst-case utility satisfying:

$$U^{a,*}_j \leq U^a(\mathbf{x}^*_>, \bar{\lambda}^{\text{dec}}_j) \leq U^a(\mathbf{x}^*_>, \lambda^{max}) \leq U^a(\mathbf{x}^*_<, \bar{\lambda}^{\text{dec}}_{j^{\text{fea}}}) = U^{a,*}_{j^{\text{fea}}}$$

Furthermore, for all j such that $n^{max}_j \leq n^{max}_{j^{\text{fea}}}$, we have $j \leq j^{\text{fea}}$ according to Observation 6. Thus, we obtain:

$$U^{a,*}_j = U^a(\mathbf{x}^*_<, \bar{\lambda}^{\text{dec}}_j) \leq U^a(\mathbf{x}^*_<, \bar{\lambda}_{j^{\text{fea}}}) = U^{a,*}_{j^{\text{fea}}}$$

Based on the above defense function and the fact that the attacker will choose $\bar{\lambda}^{\text{dec}}_{j^{\text{fea}}}$, the defender receives an utility of U^d_*. Next, we prove that this is the best the defender can obtain by showing that any defense function $\{x'_1, \ldots, x'_N\}$ such that j^{fea} is the attacker's best response will lead to a defender utility less than U^d_*. Indeed, since $n^{max}_{j^{\text{fea}}} < N$, it means $j^{\text{fea}} < M$ or in other words, $\bar{\lambda}^{\text{dec}}_{j^{\text{fea}}} < \bar{\lambda}_M = \lambda^{max}$. On other hand, since $\bar{\lambda}^{\text{dec}}_{j^{\text{fea}}}$ is the best choice of the attacker, the following inequality must hold:

$$U^{a,*}_{j^{\text{fea}}} \geq U^{a,*}_M = \min_{\mathbf{x} \in \mathbf{X}^{\text{def}}_M} U^a(\mathbf{x}, \lambda^{max}) \geq \min_{\mathbf{x} \in \mathbf{X}} U^a(\mathbf{x}, \lambda^{max})$$

This means that any defense function $\{x'_1, \ldots, x'_k\}$ such that j^{fea} is the attacker's best response has to satisfy the above inequality. As defined, U^d_* is the highest utility for the defender among these defense functions that satisfy the above inequality.

Proof of Proposition 2 First, we observe that given $\hat{\mathbf{x}}$, $\bar{\lambda}_{j^{\text{fea}}}$ is the best response of the attacker. Indeed, since $j^{\text{fea}} = M$ or equivalently $\bar{\lambda}_{j^{\text{fea}}} = \lambda^{max}$ according to Observation 7, we have:

$$U^{a,*}_{j^{\text{fea}}} = U^a(\hat{\mathbf{x}}, \lambda^{max}) \geq U^a(\hat{\mathbf{x}}, \bar{\lambda}^{\text{dec}}_j) = U^{a,*}_j, \forall j$$

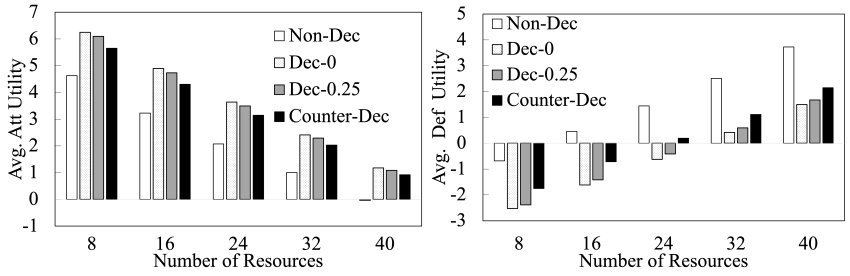
Second, since $\bar{\lambda}_{j^{\text{fea}}} = \lambda^{max}$, then for any defense function such that $\bar{\lambda}_{j^{\text{fea}}}$ is the best deception choice of the attacker, the resulting utility for the defender must be no more than:

$$\max_{\mathbf{x} \in \mathbf{X}^{\text{def}}_{j^{\text{fea}}}} U^d(\mathbf{x}, \lambda^{max}) \leq \max_{\mathbf{x} \in \mathbf{X}} U^d(\mathbf{x}, \lambda^{max})$$

regardless of the learning outcome $\lambda^{\text{learnt}} \in [\lambda^{max} - \delta, \lambda^{max} + \delta]$. This is because the defender eventually plays one of the defense strategies in the set $\mathbf{X}^{\text{def}}_{j^{\text{fea}}}$. The RHS is the defender's utility obtained by playing the counter-deception specified by the proposition.

Appendix D: Additional Experiment Results

We provide additional set of experiments.

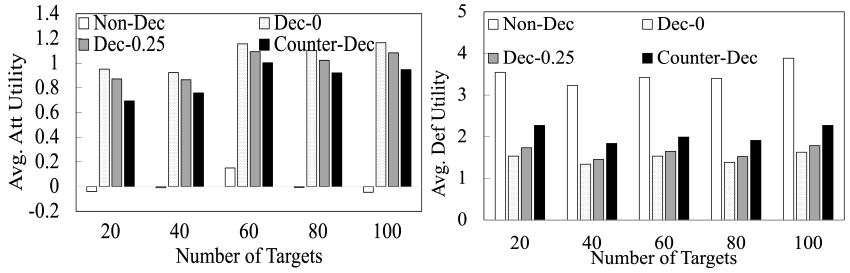


(a) Attacker utility, 80 targets (b) Defender utility, 80 targets

Fig. 5: Player Utilities with Varying Number of Resources

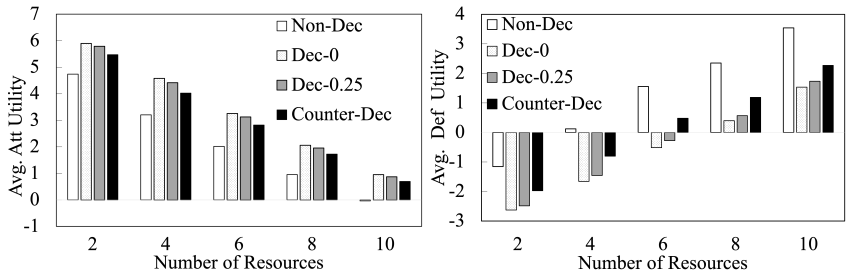
Figure 5 shows the performance of our algorithms as we vary the number of resources L on 80-target games. This figure shows that the benefits of deception and counter-deception to the players are observed consistently when varying L. It shows that (i) the defender (attacker) utilities steadily increase (decrease) with increasing L; and (ii) the trends observed between the different algorithms in Figure 5 are observed consistently at different values of L.

In Figure 6, we compare different algorithms with increasing number of targets when $\frac{L}{T} = 0.5$. In Figure 7, we compare different algorithms with increasing number of security resources on 20-target games.



(a) Attacker utility, ratio = 0.5 (b) Defender utility, ratio = 0.5

Fig. 6: Player Utilities with Varying Number of Targets



(a) Attacker utility, 20 targets (b) Defender utility, 20 targets

Fig. 7: Player Utilities with Varying Number of Resources