A. More Background

A.1. Background on Gaussian Processes

In the repeated game, the attacker ($\mathcal{A}$) models its belief about its payoff function f_1 using a *Gaussian process* (GP) $\{f_1(\mathbf{x}_1, \mathbf{x}_2)\}_{\mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2}$. In particular, any finite subset of $\{f_1(\mathbf{x}_1, \mathbf{x}_2)\}_{\mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2}$ follows a multivariate Gaussian distribution (Rasmussen & Williams, 2006). A GP is fully specified by the prior mean $\mu(\mathbf{x}_1, \mathbf{x}_2)$ and kernel function $k([\mathbf{x}_1, \mathbf{x}_2], [\mathbf{x}'_1, \mathbf{x}'_2])$, and we assume w.l.o.g. that $\mu(\mathbf{x}_1, \mathbf{x}_2) = 0$ and $k([\mathbf{x}_1, \mathbf{x}_2], [\mathbf{x}'_1, \mathbf{x}'_2]) \leq 1$ for all $\mathbf{x}_1, \mathbf{x}'_1 \in \mathcal{X}_1$ and $\mathbf{x}_2, \mathbf{x}'_2 \in \mathcal{X}_2$. Given a set of T noisy observations $\mathbf{y}_T \triangleq [y_t]^\top_{t=1,\ldots,T}$ at inputs $[\mathbf{x}_{1,1}, \mathbf{x}_{2,1}], \ldots, [\mathbf{x}_{1,T}, \mathbf{x}_{2,T}]$, the posterior GP belief of f_1 at any input $[\mathbf{x}_1, \mathbf{x}_2]$ is a Gaussian distribution with the following posterior mean and variance:

$$\begin{aligned} \mu_T(\mathbf{x}_1, \mathbf{x}_2) &\triangleq \mathbf{k}_T(\mathbf{x}_1, \mathbf{x}_2)^\top (\mathbf{K}_T + \sigma^2 I)^{-1} \mathbf{y}_T \ , \\ \sigma^2_T(\mathbf{x}_1, \mathbf{x}_2) &\triangleq k([\mathbf{x}_1, \mathbf{x}_2], [\mathbf{x}_1, \mathbf{x}_2]) - \mathbf{k}_T(\mathbf{x}_1, \mathbf{x}_2)^\top (\mathbf{K}_T + \sigma^2 I)^{-1} \mathbf{k}_T(\mathbf{x}_1, \mathbf{x}_2) \end{aligned} \tag{9}$$

where $\mathbf{K}_T \triangleq \left[k([\mathbf{x}_{1,t}, \mathbf{x}_{2,t}], [\mathbf{x}_{1,t'}, \mathbf{x}_{2,t'}])\right]_{t,t'=1,\ldots,T}$ and $\mathbf{k}_T(\mathbf{x}_1, \mathbf{x}_2) \triangleq \left[k([\mathbf{x}_{1,t}, \mathbf{x}_{2,t}], [\mathbf{x}_1, \mathbf{x}_2])\right]^\top_{t=1,\ldots,T}$.

A.2. The GP-MW Algorithm

When $\mathcal{A}$ (the attacker) adopts the GP-MW algorithm as the level-0 strategy, after iteration t of the repeated game, $\mathcal{A}$ calculates the updated value of the GP-UCB acquisition function at every input in its entire domain $\mathcal{X}_1$ (while fixing the defender's input $\mathbf{x}_2$ at the value selected in iteration t: $\mathbf{x}_{2,t}$), plugs in the (negative) GP-UCB values as the loss vector (with the length of the vector being equal to the size of its domain: $|\mathcal{X}_1|$) in the widely used multiplicative-weight online learning algorithm to update the randomized/mixed strategy $\mathcal{P}^0_{1,t+1}$. Subsequently, the resulting updated distribution will be used to sample $\mathcal{A}$'s action in the next iteration $t+1$, i.e., $\mathbf{x}_{1,t+1} \sim \mathcal{P}^0_{1,t+1}$. Note that the proof of Theorem 1 results from a slight modification to the proof of GP-MW (Sessa et al., 2019), i.e., the work of Sessa et al. (2019) has assumed that the payoff function has bounded norm in a reproducing kernel Hilbert space, whereas we assume that the payoff function is sampled from a GP. Both assumptions are commonly used in the analysis of BO algorithms. Refer to the work of Sessa et al. (2019) for more details about the GP-MW algorithm.

B. Extension to Games Involving More than Two Agents

The R2-B2, as well as R2-B2-Lite, algorithm can be extended to repeated games involving more than two ($M > 2$) agents. A motivating scenario for this type of games with $M > 2$ agents is MARL, in which every individual agent attempts to maximize its own return (payoff). Here, we use $\mathcal{A}_1, \ldots, \mathcal{A}_M$ to represent the M agents.

Level-$k = 0$ Strategy. The extension of level-0 reasoning is trivial since level-0 strategies are agnostic with respect to the other agent's action selection strategies, and can thus treat all other agents as a single collective agent. As a result, if GP-MW is adopted as the level-0 strategy, the theoretical guarantee of Theorem 1 still holds.

Level-$k = 1$ Strategy. If the agent $\mathcal{A}_1$ thinks that all other agents ($\mathcal{A}_2, \ldots, \mathcal{A}_M$) reason at level 0 and knows the level-0 strategies of all other agents, $\mathcal{A}_1$ can reason at level 1 by:

$$\mathbf{x}^1_{1,t} = \arg\max_{\mathbf{x}_1 \in \mathcal{X}_1} \mathbb{E}_{\mathbf{x}^0_{2,t}, \ldots, \mathbf{x}^0_{M,t}} \left[\alpha_{1,t}(\mathbf{x}_1, \mathbf{x}^0_{2,t}, \ldots, \mathbf{x}^0_{M,t})\right], \tag{10}$$

in which the expectation is taken over the level-0 strategies of all other agents $\mathcal{A}_2, \ldots, \mathcal{A}_M$. R2-B2-Lite can also be applied:

$$\mathbf{x}^1_{1,t} = \arg\max_{\mathbf{x}_1 \in \mathcal{X}_1} \alpha_{1,t}(\mathbf{x}_1, \widetilde{\mathbf{x}}^0_{2,t}, \ldots, \widetilde{\mathbf{x}}^0_{M,t}), \tag{11}$$

in which $\widetilde{\mathbf{x}}^0_{2,t}, \ldots, \widetilde{\mathbf{x}}^0_{M,t}$ are sampled from the corresponding level-0 strategies of agents $\mathcal{A}_2, \ldots, \mathcal{A}_M$.

For level-1 reasoning, the actions of all other agents can be viewed as the joint action of a single collective agent, whose level-0 strategy (action distribution) factorizes across different agents. As a result, the theoretical guarantees of Theorems 2 and 4 are still valid.

Level-$k \geq 2$ Strategy. Level-$k \geq 2$ reasoning with $M > 2$ agents is significantly more complicated than the two-agent setting, mainly due to the fact that the other agents may not reason at the same level. For simplicity, we consider the scenario in which the agent $\mathcal{A}_1$ reasons at level 2, and thus all other agents reason at either level 1 or 0. This is a common scenario

since as discussed in Section 3.1.3 and will be explained at the end of this section, the agents have a strong tendency to reason at lower levels in the setting with $M > 2$ agents. Without loss of generality, we assume that agents 2 to M_0 reason at level 0, and agents $M_0 + 1$ to M reason at level 1 (by following the strategy of (10)). In this case, the level-2 action of agent $\mathcal{A}_1$ is selected by best-responding to the corresponding strategy of each of the other agents:

$$\mathbf{x}_{1,t}^2 = \arg\max_{\mathbf{x}_1 \in \mathcal{X}_1} \mathbb{E}_{\mathbf{x}_{2,t}^0, \ldots, \mathbf{x}_{M_0,t}^0} \left[\alpha_{1,t}(\mathbf{x}_1, \mathbf{x}_{2,t}^0, \ldots, \mathbf{x}_{M_0,t}^0, \mathbf{x}_{M_0+1,t}^1, \ldots, \mathbf{x}_{M,t}^1) \right]. \tag{12}$$

Specifically, the level-1 actions of those agents reasoning at level 1 ($\mathbf{x}_{M_0+1,t}^1, \ldots, \mathbf{x}_{M,t}^1$) can be calculated using (10), and the expectation in (12) is taken with respect to the level-0 strategies of those agents reasoning at level 0 ($\mathbf{x}_{2,t}^0, \ldots, \mathbf{x}_{M_0,t}^0$). Interestingly, the level-2 reasoning strategy of (12) enjoys the same regret upper bound as shown in Theorem 2 or Theorem 3, depending on whether there exists level-0 agents (see the detailed explanation and the proof in Appendix E). Unfortunately, the complexity of reasoning at levels $k \geq 3$ grows excessively. Firstly, every other agent reasoning at a lower level $k \geq 2$ may best-respond to the other agents in multiple ways. For example, if there are $M = 3$ agents in the environment and agent $\mathcal{A}_1$ reasons at level 2, $\mathcal{A}_1$ might choose its level-2 action in three different ways, with the corresponding reasoning levels of the 3 agents being $[2, 1, 1]$, $[2, 1, 0]$ or $[2, 0, 1]$. As a result, if Agent $\mathcal{A}_2$ chooses to reason at level 3, in addition to obtaining the information that agent $\mathcal{A}_1$ reasons at level 2, $\mathcal{A}_2$ also needs to additionally know in which of the three ways will the level-2 reasoning of $\mathcal{A}_1$ be performed. Therefore, when $M > 2$ agents are present, as the reasoning level increases, the reasoning complexity, as well as computational cost, grows significantly. As a consequence, compared with the agents in 2-agent games, the agents in games with $M > 2$ agents are expected to display a stronger preference to reasoning at low levels.

C. Proof of Theorems 2 and 3

Before proving the main theorems, we need the following lemma showing a high-probability uniform upper bound on the value of the payoff function.

Lemma 1. *Let $\delta \in (0, 1)$ and $\beta_t = 2\log(|\mathcal{X}_1|t^2\pi^2/3\delta)$, then with probability $\geq 1 - \delta$,*

$$|f_1(\mathbf{x}_1, \mathbf{x}_2) - \mu_{t-1}(\mathbf{x}_1, \mathbf{x}_2)| \leq \beta_t^{1/2} \sigma_{t-1}(\mathbf{x}_1, \mathbf{x}_2)$$

for all $\mathbf{x}_1 \in \mathcal{X}_1$, $\mathbf{x}_2 \in \mathcal{X}_2$, and $t \geq 1$.

The proof of Lemma 1 makes use of the Gaussian concentration inequality and the union bound, and the proof can be found in Lemma 5.1 of Srinivas et al. (2010). Note that a tighter confidence bound (i.e., a smaller value of $\beta_t = 2\log(|\mathcal{X}_1|t^2\pi^2/6\delta)$) is possible, however, the value of β_t in Lemma 1 is selected for convenience to match the requirement of GP-MW (Theorem 1).

C.1. Theorem 2

Denote the history of game plays for $\mathcal{D}$ (the defender) up to iteration $t-1$ as $\mathcal{H}_{t-1}$, which includes $\mathcal{D}$'s selected actions (inputs) and observed payoffs (outputs) in every iteration from 1 to $t-1$: $\mathcal{H}_{t-1} = [\mathbf{x}_{2,1}, y_{2,1}, \mathbf{x}_{2,2}, y_{2,2}, \ldots, \mathbf{x}_{2,t-1}, y_{2,t-1}]$. Again, we use superscripts to denote the reasoning level such that if $\mathcal{D}$ reasons at level 0, $\mathcal{H}_{t-1} = [\mathbf{x}_{2,1}^0, y_{2,1}^0, \mathbf{x}_{2,2}^0, y_{2,2}^0, \ldots, \mathbf{x}_{2,t-1}^0, y_{2,t-1}^0]$.

Here, we analyze the regret of the level-1 strategy, i.e., when $\mathcal{A}$ (the attacker) reasons at level $k = 1$ and $\mathcal{D}$ (the defender) reasons at level $k' = 0$. Note that in iteration t, the level-0 strategy of $\mathcal{D}$ (i.e., the distribution of $\mathbf{x}_{2,t}$) may depend on the history of input-output pairs of $\mathcal{D}$, i.e., $\mathcal{H}_{t-1}$, which is true for both the GP-MW and EXP3 strategies. Therefore, when analyzing $\mathcal{A}$'s expected regret in iteration t (with the expectation taken over the level-0 strategy of $\mathcal{D}$ in iteration t), we need to condition on $\mathcal{H}_{t-1}$. We denote the regret of $\mathcal{A}$ in iteration t as $r_{1,t}$, i.e., $R_{1,T} = \sum_{t=1}^{T} r_{1,t}$ in which $R_{1,T}$ represents external regret defined in (1). As a result, with probability of at least $1 - \delta$, the expected regret of $\mathcal{A}$ (the attacker) in iteration

t, given $\mathcal{H}_{t-1}$, can be analyzed as

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}^0_{2,t}}[r_{1,t}|\mathcal{H}_{t-1}] &= \mathbb{E}_{\mathbf{x}^0_{2,t}}\left[f_1\left(\mathbf{x}^*_1, \mathbf{x}^0_{2,t}\right) - f_1\left(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}\right)|\mathcal{H}_{t-1}\right] \\
&\overset{(a)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\alpha_{1,t}\left(\mathbf{x}^*_1, \mathbf{x}^0_{2,t}\right) - f_1\left(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}\right)|\mathcal{H}_{t-1}\right] \\
&\overset{(b)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\alpha_{1,t}\left(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}\right) - f_1\left(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}\right)|\mathcal{H}_{t-1}\right] \\
&\overset{(c)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\mu_{t-1}(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}) - f_1\left(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t}\right)|\mathcal{H}_{t-1}\right] \\
&\overset{(d)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}, \mathbf{x}^0_{2,t})|\mathcal{H}_{t-1}\right]
\end{aligned} \tag{13}$$

in which (a) results from Lemma 1 and the definition of the GP-UCB acquisition function (α) in Section 2, (b) follows from the definition of the level-1 strategy (3) as well as the linearity of the expectation operator, (c) results from the definition of the GP-UCB acquisition function, and (d) is again a consequence of Lemma 1.

Next, the expected external regret of $\mathcal{A}$ reasoning at level 1 can be upper-bounded:

$$\begin{aligned}
\mathbb{E}[R_{1,T}] &= \mathbb{E}_{\mathbf{x}^0_{2,1}, y^0_{2,1}, \ldots, \mathbf{x}^0_{2,T-1}, y^0_{2,T-1}, \mathbf{x}^0_{2,T}}[R_{1,T}] \\
&= \mathbb{E}_{\mathbf{x}^0_{2,1}, y^0_{2,1}, \ldots, \mathbf{x}^0_{2,T-1}, y^0_{2,T-1}, \mathbf{x}^0_{2,T}}\left[\sum_{t=1}^{T} r_{1,t}\right] \\
&\overset{(a)}{=} \mathbb{E}_{\mathbf{x}^0_{2,1}}\left[r_{1,1}\right] + \mathbb{E}_{\mathbf{x}^0_{2,1}, y^0_{2,1}, \mathbf{x}^0_{2,2}}\left[r_{1,2}\right] + \ldots + \mathbb{E}_{\mathbf{x}^0_{2,1}, y^0_{2,1}, \ldots, \mathbf{x}^0_{2,T-1}, y^0_{2,T-1}, \mathbf{x}^0_{2,T}}\left[r_{1,T}\right] \\
&\overset{(b)}{=} \mathbb{E}_{\mathbf{x}^0_{2,1}}\left[r_{1,1}\right] + \mathbb{E}_{\mathbf{x}^0_{2,1}, y^0_{2,1}}\left[\mathbb{E}_{\mathbf{x}^0_{2,2}}\left[r_{1,2}|\mathbf{x}^0_{2,1}, y^0_{2,1}\right]\right] + \ldots + \\
&\quad \mathbb{E}_{\mathbf{x}^0_{2,1}, y^0_{2,1}, \ldots, \mathbf{x}^0_{2,T-1}, y^0_{2,T-1}}\left[\mathbb{E}_{\mathbf{x}^0_{2,T}}\left[r_{1,T}|\mathbf{x}^0_{2,1}, y^0_{2,1}, \ldots, \mathbf{x}^0_{2,T-1}, y^0_{2,T-1}\right]\right] \\
&= \mathbb{E}_{\mathbf{x}^0_{2,1}}\left[r_{1,1}\right] + \mathbb{E}_{\mathcal{H}_1}\left[\mathbb{E}_{\mathbf{x}^0_{2,2}}\left[r_{1,2}|\mathcal{H}_1\right]\right] + \ldots + \mathbb{E}_{\mathcal{H}_{T-1}}\left[\mathbb{E}_{\mathbf{x}^0_{2,T}}\left[r_{1,T}|\mathcal{H}_{T-1}\right]\right] \\
&\overset{(c)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,1}}\left[2\beta_1^{1/2}\sigma_0(\mathbf{x}_{1,1}, \mathbf{x}_{2,1})\right] + \mathbb{E}_{\mathcal{H}_1}\left[\mathbb{E}_{\mathbf{x}^0_{2,2}}\left[2\beta_2^{1/2}\sigma_1(\mathbf{x}_{1,2}, \mathbf{x}_{2,2})|\mathcal{H}_1\right]\right] + \ldots + \\
&\quad \mathbb{E}_{\mathcal{H}_{T-1}}\left[\mathbb{E}_{\mathbf{x}^0_{2,T}}\left[2\beta_T^{1/2}\sigma_{T-1}(\mathbf{x}_{1,T}, \mathbf{x}_{2,T})|\mathcal{H}_{T-1}\right]\right] \\
&\overset{(d)}{=} \mathbb{E}_{\mathbf{x}^0_{2,1}}\left[2\beta_1^{1/2}\sigma_0(\mathbf{x}_{1,1}, \mathbf{x}_{2,1})\right] + \mathbb{E}_{\mathcal{H}_1, \mathbf{x}^0_{2,2}}\left[2\beta_2^{1/2}\sigma_1(\mathbf{x}_{1,2}, \mathbf{x}_{2,2})\right] + \ldots + \\
&\quad \mathbb{E}_{\mathcal{H}_{T-1}, \mathbf{x}^0_{2,T}}\left[2\beta_T^{1/2}\sigma_{T-1}(\mathbf{x}_{1,T}, \mathbf{x}_{2,T})\right] \\
&\overset{(e)}{=} \mathbb{E}_{\mathcal{H}_{T-1}, \mathbf{x}^0_{2,T}}\left[\sum_{t=1}^{T} 2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_{1,t}, \mathbf{x}_{2,t})\right] \\
&\overset{(f)}{\leq} \mathbb{E}_{\mathcal{H}_{T-1}, \mathbf{x}^0_{2,T}}\left[\sqrt{C_1 T \beta_T \gamma_T}\right] \\
&\overset{(g)}{=} \sqrt{C_1 T \beta_T \gamma_T}
\end{aligned} \tag{14}$$

in which $C_1 = 8/\log(1+\sigma_1^{-2})$, β_T is defined in Lemma 1, and γ_T is the maximum information gain about the function f_1 obtained from any set of observations of size T. Steps (a) and (e) both result from the fact that $r_{1,t}$ only depends on the level-0 strategy of iteration t and the history up to iteration $t-1$ (through the level-0 strategy of iteration t), and is thus independent of those input actions and output observations in future iterations $t+1, \ldots, T$. (b) and (d) both follow from the law of total expectation, (c) results from (13), (f) follows from Lemmas 5.3 and 5.4 of Srinivas et al. (2010), (g) follows since all terms inside the expectation are independent of the history of input-output pairs. Note that the expectation

in (14) is taken over the history of selected actions and observed payoffs of $\mathcal{D}$. Note that an upper bound on the regret can be easily derived using the upper bound on the expected regret (14) through Markov's inequality, which suggests that level-1 reasoning achieves no regret asymptotically.

Of note, in the scenario in which more than two ($M > 2$) agents are present (Appendix B), with the modified level-1 policy given by (10), the proofs of (13) and (14) still go through by simply replacing $\mathbf{x}_{2,t}^0$ with the concatenated vector of $[\mathbf{x}_{2,t}^0, \ldots, \mathbf{x}_{M,t}^0]$ (i.e., the concatenation of the level-0 actions of all other agents) in every step of the proof. Similarly, the expectation of the regret would be taken over the history of input-output pairs of all other agents $2, \ldots, M$.

C.2. Theorem 3

For level-$k \geq 2$ reasoning, i.e., when $\mathcal{A}$ reasons at level k (for $k \geq 2$) and $\mathcal{D}$ reasons at level $k' = k - 1 \geq 1$, the regret of $\mathcal{A}$ in iteration t can be analyzed as:

$$\begin{aligned} r_{1,t} &= f_1\left(\mathbf{x}_1^*, \mathbf{x}_{2,t}\right) - f_1\left(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}\right) \\ &= f_1\left(\mathbf{x}_1^*, \mathbf{x}_{2,t}^{k-1}\right) - f_1\left(\mathbf{x}_{1,t}^k, \mathbf{x}_{2,t}^{k-1}\right) \\ &\overset{\text{(a)}}{\leq} \alpha_{1,t}\left(\mathbf{x}_1^*, \mathbf{x}_{2,t}^{k-1}\right) - f_1\left(\mathbf{x}_{1,t}^k, \mathbf{x}_{2,t}^{k-1}\right) \\ &\overset{\text{(b)}}{\leq} \alpha_{1,t}\left(\mathbf{x}_{1,t}^k, \mathbf{x}_{2,t}^{k-1}\right) - f_1\left(\mathbf{x}_{1,t}^k, \mathbf{x}_{2,t}^{k-1}\right) \\ &\leq 2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}) \end{aligned} \tag{15}$$

in which (a) follows from Lemma 1, (b) results from the fact that $\mathbf{x}_{1,t}^k$ is selected by maximizing the GP-UCB acquisition function α with respect to $\mathbf{x}_{2,t}^{k-1}$ according to (6). (15) also holds with probability of at least $1 - \delta$.

Next, the external regret can be upper bounded in a similar way as (14):

$$R_{1,T} = \sum_{t=1}^{T} r_{1,t} \overset{\text{(a)}}{\leq} \sum_{t=1}^{T} 2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}) \overset{\text{(b)}}{\leq} \sqrt{C_1 T \beta_T \gamma_T} \tag{16}$$

in which (a) results from (15), and (b) again follows from Lemmas 5.3 and 5.4 of Srinivas et al. (2010).

D. Proof of Theorem 4

Note that the level-1 action selected by $\mathcal{A}$ (the attacker) following R2-B2-Lite (8) is stochastic, instead of being deterministic as in R2-B2 (3). In the following, we denote the level-1 action of $\mathcal{A}$ following R2-B2-Lite as $\mathbf{x}_{1,t}^1(\widetilde{\mathbf{x}}_{2,t}^0)$ since, conditioned on all the game history up to iteration $t-1$, the selected level-1 action is a deterministic function of $\mathcal{A}$'s simulated action of $\mathcal{D}$ (the defender) at level 0 ($\widetilde{\mathbf{x}}_{2,t}^0$). Note that, in contrast to the corresponding definition in Appendix C.1, the history of game plays $\mathcal{H}'_{t-1}$ we define here additionally includes $\mathcal{A}$'s simulated action of $\mathcal{D}$ in every iteration: $\mathcal{H}'_{t-1} = [\mathbf{x}_{2,1}^0, \widetilde{\mathbf{x}}_{2,1}^0, y_{2,1}^0, \mathbf{x}_{2,2}^0, \widetilde{\mathbf{x}}_{2,2}^0, y_{2,2}^0, \ldots, \mathbf{x}_{2,t-1}^0, \widetilde{\mathbf{x}}_{2,t-1}^0, y_{2,t-1}^0]$. We use $\Sigma_{2,t}$ to denote the covariance matrix of the level-0 mixed strategy of $\mathcal{D}$ in iteration t ($\mathcal{P}_{2,t}$), and use $\mathrm{Tr}(\Sigma_{2,t})$ to represent its trace. As a result, the expected regret of $\mathcal{A}$ in

iteration t can be analyzed as:

$$
\begin{aligned}
\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}[r_{1,t}|\mathcal{H}'_{t-1}] &= \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[f_1\left(\mathbf{x}^*_1,\mathbf{x}^0_{2,t}\right)-f_1\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]\\
&\overset{(a)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\alpha_{1,t}\left(\mathbf{x}^*_1,\mathbf{x}^0_{2,t}\right)-f_1\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]\\
&\overset{(b)}{=} \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\alpha_{1,t}\left(\mathbf{x}^*_1,\widetilde{\mathbf{x}}^0_{2,t}\right)-f_1\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]\\
&\overset{(c)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\alpha_{1,t}\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\widetilde{\mathbf{x}}^0_{2,t}\right)-f_1\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]\\
&\overset{(d)}{=} \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\Big[\underline{\alpha_{1,t}\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^{(0,1)}_{2,t}),\widetilde{\mathbf{x}}^0_{2,t}\right)-\alpha_{1,t}\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)}\\
&\quad+\underline{\alpha_{1,t}\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)-f_1\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)}|\mathcal{H}'_{t-1}\Big]\\
&\overset{(e)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[L_{\alpha_1}\left\|\widetilde{\mathbf{x}}^0_{2,t}-\mathbf{x}^0_{2,t}\right\|_2|\mathcal{H}'_{t-1}\right]\\
&\quad+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\alpha_{1,t}\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)-f_1\left(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]\\
&\overset{(f)}{\leq} \mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[L_{\alpha_1}\sqrt{\left\|\widetilde{\mathbf{x}}^0_{2,t}-\mathbf{x}^0_{2,t}\right\|^2_2}|\mathcal{H}'_{t-1}\right]+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&\overset{(g)}{\leq} L_{\alpha_1}\sqrt{\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\left\|\widetilde{\mathbf{x}}^0_{2,t}-\mathbf{x}^0_{2,t}\right\|^2_2|\mathcal{H}'_{t-1}\right]}+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&= L_{\alpha_1}\sqrt{\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\left(\widetilde{\mathbf{x}}^0_{2,t}-\mathbf{x}^0_{2,t}\right)^\top\left(\widetilde{\mathbf{x}}^0_{2,t}-\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]}+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&= L_{\alpha_1}\sqrt{\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[\left(\widetilde{\mathbf{x}}^0_{2,t}\right)^\top\left(\widetilde{\mathbf{x}}^0_{2,t}\right)+\left(\mathbf{x}^0_{2,t}\right)^\top\left(\mathbf{x}^0_{2,t}\right)-2\left(\widetilde{\mathbf{x}}^0_{2,t}\right)^\top\left(\mathbf{x}^0_{2,t}\right)|\mathcal{H}'_{t-1}\right]}+\\
&\quad\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&\overset{(h)}{=} L_{\alpha_1}\sqrt{\mathbb{E}_{\widetilde{\mathbf{x}}^0_{2,t}}\left[\left(\widetilde{\mathbf{x}}^0_{2,t}\right)^\top\left(\widetilde{\mathbf{x}}^0_{2,t}\right)\right]+\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\left(\mathbf{x}^0_{2,t}\right)^\top\left(\mathbf{x}^0_{2,t}\right)\right]-2\mathbb{E}_{\widetilde{\mathbf{x}}^0_{2,t}}\left[\widetilde{\mathbf{x}}^0_{2,t}\right]^\top\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\mathbf{x}^0_{2,t}\right]}+\\
&\quad\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&\overset{(i)}{=} L_{\alpha_1}\sqrt{\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\left(\mathbf{x}^0_{2,t}\right)^\top\left(\mathbf{x}^0_{2,t}\right)\right]+\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\left(\mathbf{x}^0_{2,t}\right)^\top\left(\mathbf{x}^0_{2,t}\right)\right]-2\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\mathbf{x}^0_{2,t}\right]^\top\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\mathbf{x}^0_{2,t}\right]}+\\
&\quad\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&= \sqrt{2}L_{\alpha_1}\sqrt{\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\left(\mathbf{x}^0_{2,t}\right)^\top\left(\mathbf{x}^0_{2,t}\right)\right]-\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\mathbf{x}^0_{2,t}\right]^\top\mathbb{E}_{\mathbf{x}^0_{2,t}}\left[\mathbf{x}^0_{2,t}\right]}+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&\overset{(j)}{=} \sqrt{2}L_{\alpha_1}\sqrt{\mathrm{Tr}\left(\Sigma_{2,t}\right)}+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right]\\
&\overset{(k)}{\leq} \sqrt{2}L_{\alpha_1}\sqrt{\omega_t}+\mathbb{E}_{\mathbf{x}^0_{2,t},\widetilde{\mathbf{x}}^0_{2,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^1_{1,t}(\widetilde{\mathbf{x}}^0_{2,t}),\mathbf{x}^0_{2,t})|\mathcal{H}'_{t-1}\right] \qquad (17)
\end{aligned}
$$

in which (a) results from Lemma 1; (b) holds because, conditioned on $\mathcal{H}_{t-1}$, $\mathbf{x}^0_{2,t}$ and $\widetilde{\mathbf{x}}^{(0,1)}_{2,t}$ are sampled from the same distribution and thus identically distributed; (c) follows from the way in which $\mathbf{x}^1_{1,t}$ is selected using the R2-B2-Lite algorithm (8), i.e., by deterministically best-responding to $\widetilde{\mathbf{x}}^0_{2,t}$ in terms of the GP-UCB acquisition function; (d) simply subtracts and adds the same GP-UCB term; (e) follows from the Lipschitz continuity of the GP-UCB acquisition function,

whose Lipschitz constant (denoted as L_{α_1}) has been shown to be finite in (Kim & Choi, 2019); (f) is a result of the definition of the GP-UCB acquisition function (Section 2) and Lemma 1; (g) results from the concavity of the square root function; (h) follows from the linearity of expectation and the fact that $\widetilde{\mathbf{x}}^0_{2,t}$ and $\mathbf{x}^0_{2,t}$ are independent; (i) again results from the fact that $\widetilde{\mathbf{x}}^0_{2,t}$ and $\mathbf{x}^0_{2,t}$ are identically distributed; (j) follows from the definition of $\Sigma_{2,t}$, i.e., the covariance matrix of the level-0 mixed strategy of the defender in iteration t; (k) follows from our assumption in Theorem 4 that the trace of $\Sigma_{2,t}$ is upper-bounded by the sequence $\{\omega_t\}$ for all $t \geq 1$. Note that all expectations in (17) are conditioned on $\mathcal{D}'_{t-1}$, and some of the conditioning are omitted to shorten the expression.

Next, the expected external regret can be upper-bounded in a similar way as (14):

$$\begin{aligned}\mathbb{E}_{\mathbf{x}^0_{2,1},\widetilde{\mathbf{x}}^0_{2,1},y^0_{2,1},\ldots,\mathbf{x}^0_{2,T-1},\widetilde{\mathbf{x}}^0_{2,T-1},y^0_{2,T-1},\mathbf{x}^0_{2,T},\widetilde{\mathbf{x}}^0_{2,T}}[R_{1,T}] &= \mathbb{E}_{\mathbf{x}^0_{2,1},\widetilde{\mathbf{x}}^0_{2,1},y^0_{2,1},\ldots,\mathbf{x}^0_{2,T-1},\widetilde{\mathbf{x}}^0_{2,T-1},y^0_{2,T-1},\mathbf{x}^0_{2,T},\widetilde{\mathbf{x}}^0_{2,T}}\left[\sum_{t=1}^{T} r_{1,t}\right] \\ &\leq \sqrt{2}L_{\alpha_1}\sum_{t=1}^{T}\sqrt{\omega_t} + \sqrt{C_1 T\beta_T\gamma_T}\end{aligned} \tag{18}$$

Note that compared with Theorem 2, the expectation in Theorem 4 is additionally taken over $\mathcal{A}$'s simulated action of $\mathcal{D}$ in all iterations, i.e., $\widetilde{\mathbf{x}}^0_{2,1},\ldots,\widetilde{\mathbf{x}}^0_{2,T}$. Finally, Theorem 4 follows:

$$\mathbb{E}[R_{1,T}] \leq \mathcal{O}\left(\sum_{t=1}^{T}\sqrt{\omega_t} + \sqrt{T\beta_T\gamma_T}\right) \tag{19}$$

Similar to the analysis of R2-B2, in the scenario where more than two ($M > 2$) agents are involved, with the modified level-1 R2-B2-Lite algorithm given by (11), the proofs given above still go through by simply replacing $\mathbf{x}^0_{2,t}$ with the concatenated vector of $[\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M,t}]$ (and replacing $\widetilde{\mathbf{x}}^0_{2,t}$ with the concatenated vector of $[\widetilde{\mathbf{x}}^0_{2,t},\ldots,\widetilde{\mathbf{x}}^0_{M,t}]$) in every step of the proof. Again, the expectation of the regret of agent $\mathcal{A}_1$ is taken over the history of input-output pairs of all other agents, as well as $\mathcal{A}_1$'s simulated level-0 actions of all other agents in every iteration.

E. Proof of Theorems 2 and 3 for $M > 2$ Agents

We prove here that the regret upper bound in Theorems 2 and 3 also hold in games with $M > 2$ agents. We only give the proof for level-$k \geq 2$ strategy since the proofs for level-0 and level-1 strategies are straightforward as explained in Appendices B and C. For simplicity, we only focus on the scenario in which agent $\mathcal{A}_1$ reasons at level 2, whereas all other agents reason at either level 0 or level 1. However, the proof can be generalized to the settings in which agent $\mathcal{A}_1$ reasons at a higher level $k > 2$. Following the notations of Appendix B, the expected regret of $\mathcal{A}_1$ in iteration t can be upper bounded as:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t}}[r_{1,t}|\mathcal{H}_{t-1}] =& \mathbb{E}_{\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t}}\Big[f_1\left(\mathbf{x}^*_1,\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}\right) - \\ & f_1\left(\mathbf{x}_{1,t},\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}\right)|\mathcal{H}_{t-1}\Big] \\ \leq & \mathbb{E}_{\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t}}\Big[\alpha_{1,t}\left(\mathbf{x}^*_1,\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}\right) - \\ & f_1\left(\mathbf{x}^1_{2,t},\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}\right)|\mathcal{H}_{t-1}\Big] \\ \leq & \mathbb{E}_{\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t}}\Big[\alpha_{1,t}\left(\mathbf{x}^2_{1,t},\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}\right) - \\ & f_1\left(\mathbf{x}^1_{2,t},\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}\right)|\mathcal{H}_{t-1}\Big] \\ \leq & \mathbb{E}_{\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t}}\left[2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^2_{1,t},\mathbf{x}^0_{2,t},\ldots,\mathbf{x}^0_{M_0,t},\mathbf{x}^1_{M_0+1,t},\ldots,\mathbf{x}^1_{M,t}|\mathcal{H}_{t-1}\right]\end{aligned} \tag{20}$$

The proof given in (20) is analogous to (13). The key difference from (13) is that in this case, the expectation here is taken over the level-0 strategies of those agents reasoning at level 0, i.e., $\mathcal{A}_2,\ldots,\mathcal{A}_{M_0}$. In contrast, in (13), the expectation is only taken over the level-0 strategy of the single opponent reasoning at level 0.

Note that if none of the other agents reason at level 0, the expectation operator in (20) can be dropped. As a result, (16) can be directly used to show that the resulting upper bound on the regret is the same as that given in Theorem 3. On the other hand, if there exists at least 1 level-0 agents, the expectation operator remains. Therefore, the subsequent proof follows from (14) and the resulting regret upper bound becomes the same as that shown in Theorem 2, except that the expectation of the regret is taken over the history of input-output pairs of all level-0 agents.

F. More Experimental Details and Results

All experiments are run on computers with 16 cores of Intel Xeon processor, 5 NVIDIA GTX1080 Ti GPUs, and a RAM of 256G.

F.1. Synthetic Games

F.1.1. 2-Agent Synthetic Games

(a) Detailed Experimental Setting

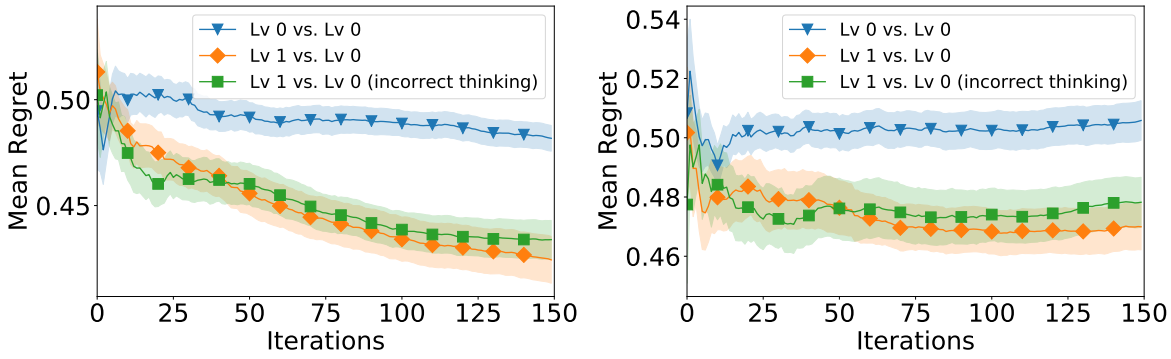
The payoff functions used in the synthetic games are sampled from GPs with the Squared Exponential kernel with length scale 0.1. All payoff functions are defined on a 2-dimensional grid of equally spaced points in $[0, 1]^2$ with size $|\mathcal{X}_1| \times |\mathcal{X}_2| = 100 \times 100$. Therefore, the action spaces of agent 1 and agent 2 both consist of $|\mathcal{X}_1| = |\mathcal{X}_2| = 100$ points. For common-payoff games, we randomly sample a function f_1 from a GP on the domain $\mathcal{X}_1 \times \mathcal{X}_2$ and set $f_2(\mathbf{x}_1, \mathbf{x}_2) = f_1(\mathbf{x}_1, \mathbf{x}_2)$ for all $\mathbf{x}_1 \in \mathcal{X}_1$ and $\mathbf{x}_2 \in \mathcal{X}_2$; regarding general-sum games, we randomly and independently sample two functions, f_1 and f_2, from the same GP; as for constant-sum games, we draw a function f_1 from the GP, and set $f_2(\mathbf{x}_1, \mathbf{x}_2) = 1 - f_1(\mathbf{x}_1, \mathbf{x}_2)$ for all $\mathbf{x}_1 \in \mathcal{X}_1$ and $\mathbf{x}_2 \in \mathcal{X}_2$. All payoff functions are scaled into the range $[0, 1]$. Note that since the domain size is not excessively large, the level-1 action can be selected by solving (3) exactly instead of approximately. The true GP hyperparameters, with which the synthetic payoff functions are sampled, are used as the GP hyperparameters.

(b) More Results on the Impact of Incorrect Thinking about the Other Agent

We further investigate how the performance of an agent is affected by incorrect thinking about the other agent. Fig. 5 plots the performance of agent 1 when agent 1 and agent 2 reason at levels 1 and 0 respectively, while agent 1's thinking about agent 2's level-0 strategy is incorrect. The figures demonstrate that in the presence of an incorrect thinking about the other agent's level-0 strategy, the performance of agent 1 only suffers from a marginal drop, although the theoretical guarantee offered by Theorem 2 no longer holds. Fig. 6 illustrates the impacts of an incorrect thinking about the other agent's reasoning level. As shown in the figure, when agent 2's reasoning level is fixed at level 0, agent 1 obtains the best performance when reasoning at level 1, which agrees with our theoretical analysis since by reasoning at level 1, agent 1's performance is theoretically guaranteed (Theorem 2). Meanwhile, when agent 1 reasons at a higher level (e.g., level 2 or level 3), the performance becomes worse (compared with reasoning at level 1) yet is still better than reasoning at level 0 (the blue curve); this might be attributed to the fact that when agent 1 reasons at level 2 or 3, even though agent 1's GP-UCB value is highly likely to be maximized with respect to the wrong action in every iteration (6), this could still help agent 1 to eliminate some potentially "dominated actions", i.e., those actions which yield small GP-UCB values regardless of the action of agent 2. This ability to discard those dominated actions gives agent 1 a preference to avoid selecting actions with small GP-UCB values, and thus might help agent 1 obtain a better performance compared with reasoning at level 0.

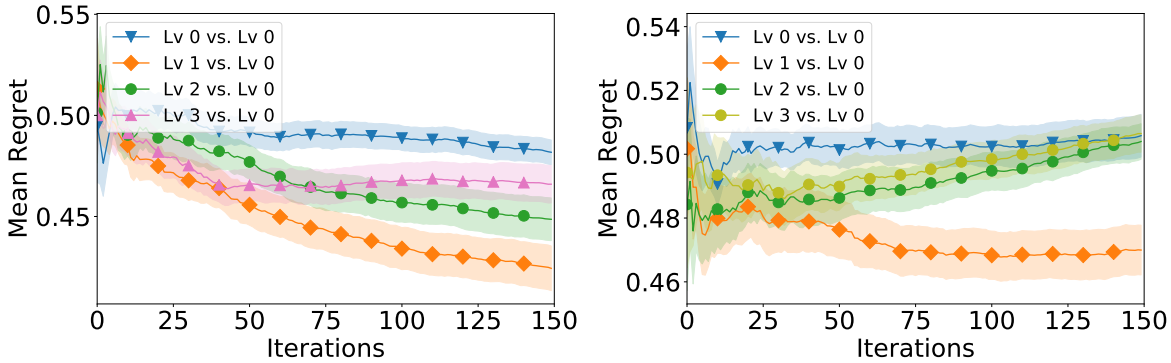
(c) Results Using Other Level-0 Strategies

In addition to the results presented in the main text which use GP-MW as the level-0 strategy (Fig. 2a to c), the entire set of experiments are repeated for the random search and EXP3 level-0 strategies, whose corresponding results are presented in Figs. 7 and 8. These results yield the same observations and interpretations as Figs. 2a to c, and demonstrate the robustness of our R2-B2 algorithm with respect to the choice of the level-0 strategy. Another interesting observation regarding different level-0 strategies is that in common-payoff and general-sum games, when both agents reason at level 0, running a no-regret level-0 strategy (e.g., GP-MW or EXP3), instead of random search, leads to decreasing mean regret. Specifically, when both agents reason at level 0, the mean regret in common-payoff and general-sum games is decreasing if either GP-MW (Fig. 2a and b) or EXP3 (Fig. 8a and b) is used as the level-0 strategy (with the decreasing trend more discernible in common-payoff games), while the random search level-0 strategy results in a non-decreasing mean regret (Fig. 7a and b). This observation demonstrates the benefit of adopting a better/more strategic level-0 strategy (instead of a non-strategic level-0 strategy such as random search) when reasoning at level 0.



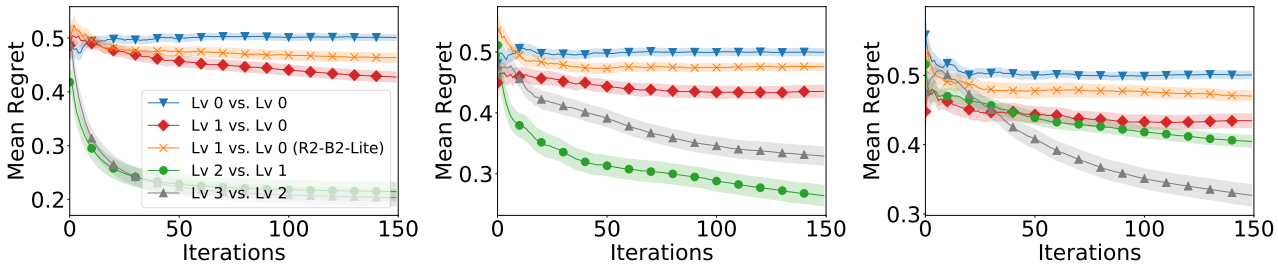
(a) General-sum games. (b) Constant-sum games.

Figure 5. Agent 1's performance of level-1 reasoning (agent 2 reasons at level 0) when agent 1's thinking about agent 2's level-0 strategy is incorrect. I.e., agent 2 uses GP-MW as the level-0 strategy, while agent 1 thinks that agent 2 uses the random search level-0 strategy.



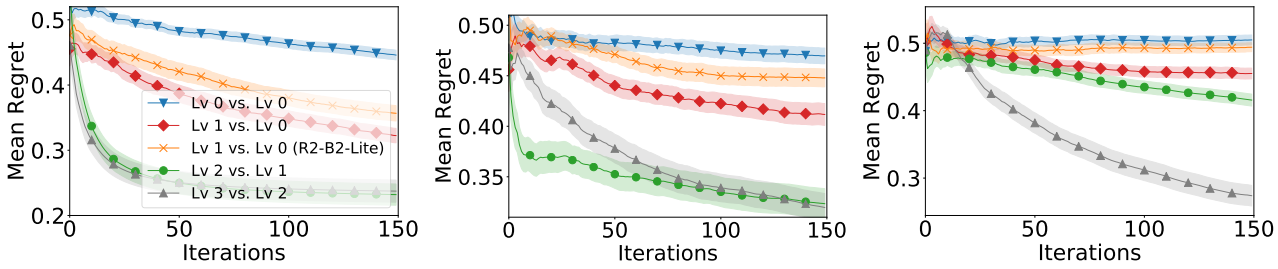
(a) General-sum games. (b) Constant-sum games.

Figure 6. Agent 1's performance when its thinking about agent 2's reasoning level is incorrect. That is, agent 2 reasons at level 0, while agent 1 reasons at levels 1, 2 and 3, where the last two settings result from agent 1's incorrect thinking about agent 2's reasoning level.



(a) Common-payoff games. (b) General-sum games. (c) Constant-sum games.

Figure 7. Mean regret of agent 1 in different types of synthetic games, with agent 2 taking the random search level-0 strategy.



(a) Common-payoff games. (b) General-sum games. (c) Constant-sum games.

Figure 8. Mean regret of agent 1 in different types of synthetic games, with agent 2 taking the EXP-3 level-0 strategy.

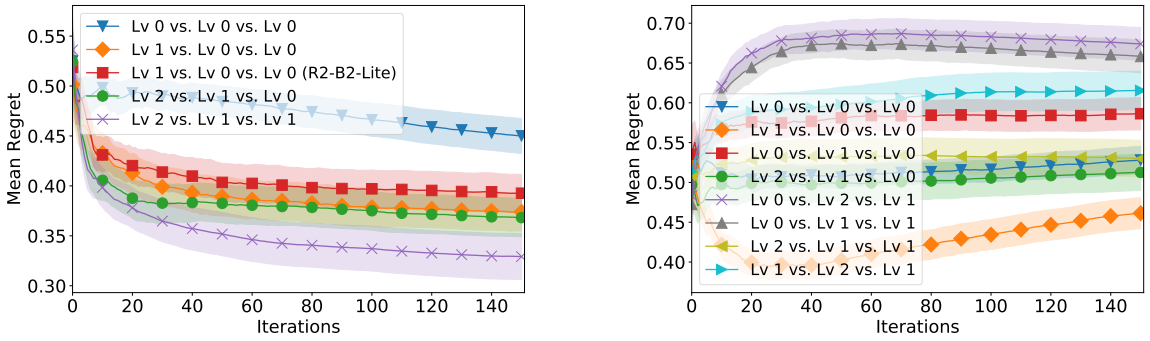
For the EXP3 level-0 strategy, we follow the practice of the work of Rahimi & Recht (2007). That is, we firstly draw $d_1' = 5$ samples of $[\omega_i]_{i=1,\dots,d_1'}$ from the spectral density of the GP kernel (i.e., the Squared Exponential kernel with length scale 0.1), and d_1' samples of $[b_i]_{i=1,\dots,d_1'}$ from the uniform distribution over $[0, 2\pi]$; then, for every input $\mathbf{x}_1 \in \mathcal{X}_1$ in the domain, we use $[\sqrt{2/d_1'}\cos(\omega_i \mathbf{x}_1 + b_i)]_{i=1,\dots,d_1'}$ as the d_1'-dimensional feature representing $\mathbf{x}_1$. Subsequently, the GP surrogate can be replaced with a linear surrogate model with the resulting features as inputs, and thus the EXP3 algorithm for adversarial linear bandit can be applied.

F.1.2. Synthetic Games with $M > 2$ Agents

We also use synthetic games with $M > 2$ agents to evaluate the effectiveness of our R2-B2 algorithm when more than two agents are involved. We consider two types of synthetic games involving three agents. In the first type of games, the payoff functions of the three agents are independently sampled from a GP. The second type of games includes one adversary and two (cooperating) agents, the payoff function for the adversary, $f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, is a function sampled from a GP (and scaled to the range $[0, 1]$), whereas the payoff functions for the two agents are identical and defined as $1 - f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$. We use GP-MW as the level-0 strategy.

Fig. 9a displays the mean regret of agent 1 in the first type of games, i.e., games with independent payoff functions. The figure shows that in games with more than two agents, agent 1 gains benefit by following the R2-B2 algorithm presented in Appendix B. Specifically, the orange and red curves demonstrate the advantage of level-1 reasoning using R2-B2 (10) and R2-B2-Lite (11) respectively, and the green and purple curves illustrate the benefit of level-k $k > 2$ reasoning (12).

Fig. 9b shows the mean regret of the adversary in the second type of games involving one adversary and two agents. Note that the mean regret of the two agents can be directly read from the figure since it is equal to $1-$ the mean regret of the adversary. A number of interesting insights can be drawn from Fig. 9. Comparing the orange and blue curves (similarly the green and red curves, and the yellow and gray curves) shows that the adversary obtains smaller regret by reasoning at a higher level than both agents; similarly, comparison of the blue and red curves (as well as the blue vs the purple, gray, and cyan curves) demonstrates that both agents enjoy a smaller regret when at least one of them reasons at a higher level than the adversary; comparing the gray and red curves reveals that when both agents reason at a higher level (in contrast to when one of them reasons at a higher level), the agents benefit more in terms of regret; comparison of the cyan and purple



(a) Mean regret of agent 1 in the three-agent game with independent payoff functions. The reasoning levels are in the form of agent 1 vs agent 2 vs agent 3.

(b) Mean regret of the adversary in the three-agent game with 1 adversary and 2 agents. The reasoning levels are in the form of adversary vs agent 1 vs agent 2.

Figure 9. Mean regret in three-agent games.

curves shows that given that the two agents reason at levels 2 and 1 respectively, the adversary reduces its deficit in regret by reasoning at level 1 instead of level 0.

F.2. Adversarial ML

F.2.1. R2-B2 for Adversarial ML

(a) Detailed Experimental Setting

We focus on the standard black-box setting, i.e., both $\mathcal{A}$ (the attacker) and $\mathcal{D}$ (the defender) can only access the target ML model by querying the model and observing the corresponding predictive probabilities for different classes (Tu et al., 2019). Query efficiency is of critical importance for a black-box attacker since each query of the target ML model can be costly and an excessive number of queries might lead to the risk of being detected. Similarly, when defending against an attacker who adopts a query-efficient algorithm, it is also reasonable for the defender to defend in a query-efficient manner. This justifies the use of BO-based methods for both adversarial attack and defense methods, since BO has been repeatedly demonstrated to be sample-efficient (Shahriari et al., 2016) and has been successfully applied to black-box adversarial attacks (Ru et al., 2020). The GP hyperparameters are optimized by maximizing the marginal likelihood after every 10 iterations.

Both the MNIST and CIFAR-10 datasets can be downloaded using the Keras package in Python[13]. All pixel values of all images are normalized into the range $[0, 1]$. For the MNIST dataset, we use a convolutional neural network (CNN) model[14] with 99.25% validation accuracy (trained on $60,000$ samples and validated using $10,000$ samples) as the target ML model, and for CIFAR-10, we use a ResNet model[15] with 92.32% validation accuracy (trained using $50,000$ samples and validated on $10,000$ samples, data augmentation is used). All test images used in the experiments for attack/defense are randomly selected among those correctly classified images from the validation set. To improve the query efficiency of black-box adversarial attacks, different dimensionality reduction techniques such as autoencoder have been adopted to reduce the dimensionality of image data (Tu et al., 2019). In this work, we let both $\mathcal{A}$ and $\mathcal{D}$ use Variational Autoencoders (VAEs) (Kingma & Welling, 2014) for dimensionality reduction in a realistic setting: In every iteration of the repeated game, $\mathcal{A}$ encodes the test image into a low-dimensional latent vector (i.e., the mean vector of the encoded latent distribution) using a VAE, perturbs the vector, and then decodes the perturbed vector to obtained the resulting image with perturbations; next, $\mathcal{D}$ receives the perturbed image, uses a VAE to encode the perturbed image to obtain a low-dimensional latent vector (i.e., the mean vector of the encoded latent distribution), adds transformations (perturbations) to the latent vector, and finally decodes the vector into the final image to be passed as input to the target ML model. In the experiments, the same VAE is used by both $\mathcal{A}$ and $\mathcal{D}$, but the use of different VAEs can be easily achieved. The latent dimension (LD) is $d_1 = d_2 = 2$ for MNIST and $d_1 = d_2 = 8$ for CIFAR-10; the action space for both $\mathcal{A}$ and $\mathcal{D}$ (i.e., the space of allowed perturbations to the

[13] https://keras.io/

[14] https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py

[15] https://github.com/keras-team/keras/blob/master/examples/cifar10_resnet.py

latent vectors) is $[-2, 2]^2$ for MNIST, and $[-2, 2]^8$ for CIFAR-10. For MNIST, the VAE[16] is a multi-layer perceptron (MLP) with ReLU activation, in which the input image is flattened into a 28×28-dimensional vector and both the encoder and decoder consist of a 512-dimensional hidden layer. Regarding CIFAR-10, the encoder of the VAE uses 3 convolutional layers followed by a fully connected layer, whereas the decoder uses 2 fully connected layers followed by 3 de-convolutional layers[17].

For both $\mathcal{A}$ and $\mathcal{D}$, the image produced by the decoder of their VAE is clipped such that the requirement of bounded perturbations in terms of the infinity norm (as mentioned in Section 4.2.1 of the main text) is satisfied. We consider *untargeted attacks* in this work, i.e., the attacker's (defender's) goal is to cause (prevent) misclassification of the ML model. However, our framework can also deal with *targeted attacks* (i.e., the attacker aims at causing the target ML model to misclassify a test image into a particular class) through slight modifications to the payoff functions. The payoff function value for $\mathcal{A}$ ($f_1(\mathbf{x}_1, \mathbf{x}_2)$, referred to as the *attack score*) for a pair of perturbations selected by $\mathcal{A}$ ($\mathbf{x}_1$) and $\mathcal{D}$ ($\mathbf{x}_2$) is the maximum predictive probability (corresponding to the probability that test input belongs to a class) among all *incorrect classes*, which is bounded in $(0, 1)$. For example, in a 10-class classification model (i.e., for both MNIST and CIFAR-10), if the correct/ground-truth class for a test image is 0, the value of the payoff function for $\mathcal{A}$ is the maximum predictive probability among classes 1 to 9. The payoff function for $\mathcal{D}$ is $f_2(\mathbf{x}_1, \mathbf{x}_2) = 1 - f_1(\mathbf{x}_1, \mathbf{x}_2)$ since the defender attempts to make sure that the predictive probability of the correct class remains the largest by minimizing the maximum predictive probability among all incorrect classes.

As reported in the main text (Section 4.2.1), we use GP-MW and random search as the level-0 strategies for MNIST, and only use random search for CIFAR-10. The reason is that GP-MW requires a discrete input domain (or a discretized continuous input domain) since it needs to maintain and update a discrete distribution over the input domain. Therefore, it is difficult to apply GP-MW to a high-dimensional continuous input domain (e.g., the 8-dimensional domain in the CIFAR-10 experiment) since an accurate discretization of the high-dimensional domain would lead to an intractably large domain for the discrete distribution, making it intractable to update and sample from the distribution. Similarly, the application of the EXP3 algorithm is also limited to low-dimensional input domains for the same reason.

(b) Results Using Multiple Images

Note that different images may be associated with different degrees of difficulty to attack and to defend, i.e., some images are easier to attack (and thus harder to defend) and others may be easier to defend (and thus harder to attack). Therefore, for those images that are easier to attack than to defend, it is easier for the attacker to increase the attack score than for the defender to reduce the attack score; as a result, the advantage achieved by the defender (i.e., lower attack score) when the defender reasons at one level higher would be less discernible since the defender's task (i.e., to decrease the attack score) is more difficult. On the other hand, for those images that are easier to defend than to attack (e.g., the MNIST dataset as demonstrated below), the benefit obtained by the attacker (i.e., higher attack score) when it reasons at one level higher would be harder to delineate since the attacker's task of increasing the attack score is more difficult. The image from MNIST/CIFAR-10 that is used to produce the results reported in the main text (Fig. 2d to f) is selected to ensure that the difficulties of attack and defense are comparable such that the effects of both attack and defense can be clearly illustrated.

Figs. 10 and 11 show the attack scores on the MNIST and CIFAR-10 datasets averaged over multiple randomly selected images (30 images for MNIST and 9 images for CIFAR-10). These figures yield consistent observations with those presented in the main text, except that for MNIST (Fig. 10), the attack scores are generally lower (compared with the blue curve where both $\mathcal{A}$ and $\mathcal{D}$ reason at level 0), which could be explained by the fact that the images in the MNIST dataset are generally easier to defend than to attack (i.e., it is easier to make the attack score lower than to make it higher, as explained in the previous paragraph) because of the simplicity of the dataset and the high accuracy of the target ML model (i.e., a validation accuracy of 99.25%). As a result, when $\mathcal{A}$ reasons at level 2 and $\mathcal{D}$ reasons at level 1, the attack score is lower than when both agents reason at level 0 (compare the gray and blue curves in Fig. 10). In addition to the above-mentioned factor that the MNIST dataset is in general harder to attack (i.e., harder to make the attack score higher than to make it lower), this deviation from our theoretical result (Theorem 3) might also be attributed to the error in approximating the expectation operator in level-1 reasoning. However, the benefit of reasoning at one level higher can still be observed in this case, since when the reasoning level of $\mathcal{D}$ is fixed at 1, it is still beneficial for $\mathcal{A}$ to reason at level 2 (i.e., the gray curve) instead of level 0 (i.e., the green curves). The corresponding average number of successful attacks in 150 iterations for different reasoning levels yield the same observations and interpretations as Figs. 10 and 11: For MNIST (Fig. 10), the

[16] https://github.com/keras-team/keras/blob/master/examples/variational_autoencoder.py

[17] https://github.com/chaitanya100100/VAE-for-Image-Generation

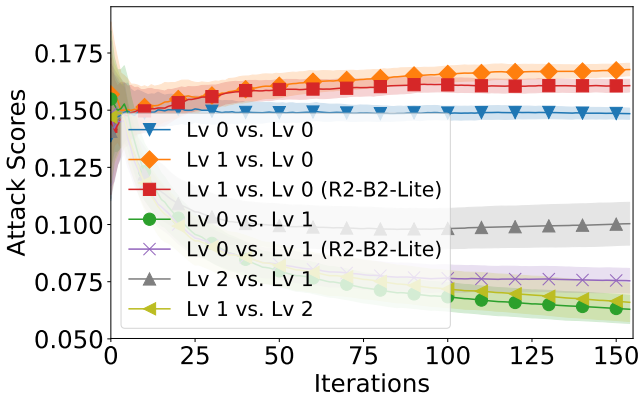


Figure 10. Attack scores averaged over 30 images from MNIST. Each image is again averaged over 5 initializations of 5 randomly selected actions.

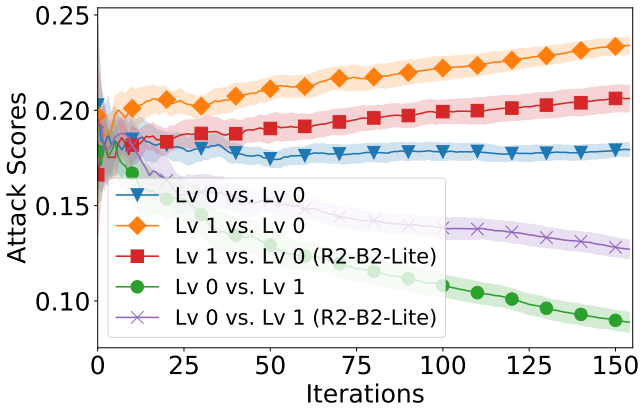


Figure 11. Attack scores averaged over 9 images from CIFAR-10. Each image is again averaged over 5 initializations of 5 randomly selected actions.

number of successful attacks are (in the order of the figure legend from top to bottom) $20.4, 23.0, 21.3, 9.7, 11.0, 12.4, 7.9$, for CIFAR-10 (Fig. 11), they are $32.9, 43.0, 38.8, 12.2, 21.0$.

(c) Impact of the Number of Samples Used for Approximating the Expectation in Level-1 Reasoning
For the results reported in the main text, the number of samples used to approximate the expectation in level-1 reasoning are 500 for MNIST (Fig. 2d and e) and $1,000$ for CIFAR-10 (Fig. 2f). Note that since the input dimension is higher for CIFAR-10, a larger number of samples is needed to accurately approximate the level-0 mixed strategy (over which the expectation in level-1 reasoning is taken). Here, we further investigate the impact of the number of samples used in the approximation of the expectation operator in level-1 reasoning (3). Fig. 12 shows the attack scores for the MNIST dataset when $\mathcal{A}$ and $\mathcal{D}$ reason at levels 2 and 1 respectively when different number of samples are used for the approximation. Random search is used as the level-0 mixed strategy. The figure, as well as the corresponding number of successful attacks, demonstrates that the attack becomes more effective as more samples are used for the approximation. The benefit offered by using more samples for the approximation results from the fact that with a better accuracy at estimating $\mathcal{D}$'s level-1 action (5) (i.e., the level-1 action of $\mathcal{D}$ simulated by $\mathcal{A}$ is more likely to be the same as the actual level-1 action selected by $\mathcal{D}$), the attacker is able to best-respond to $\mathcal{D}$'s action more accurately (4), thus leading to an improved performance.

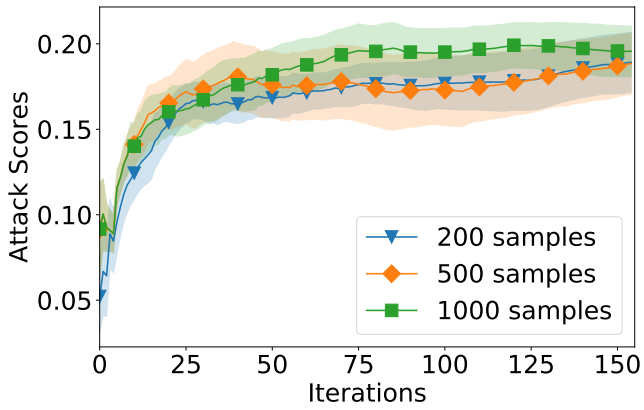


Figure 12. Attack scores for MNIST when $\mathcal{A}$ (the attacker) and $\mathcal{D}$ (the defender) reason at levels 2 and 1 respectively, with different number of samples used for approximating the expectation for level 1 reasoning. The corresponding number of successful attacks (for 200, 500 and 1000 samples) are 2.6, 3.0 and 3.3.

F.2.2. Defense against State-of-the-art Adversarial Attack Methods

(a) Against the Parsimonious Attacker[18]

Since the Parsimonious algorithm is deterministic (assuming that the random seed is fixed), it corresponds to a level-0 pure strategy, which is equivalent to a mixed strategy with all probability measure concentrated on a single action. Therefore, in our setting, when $\mathcal{D}$ (the defender) is selecting its level-1 strategy in iteration t using R2-B2, it knows exactly the action (perturbations) that $\mathcal{A}$ (the attacker) will select in the current iteration t. To make the setting more practical, we use the (encoded) image perturbed by $\mathcal{A}$ (instead of the encoded perturbations as in the experiments in Section 4.2.1) as the action of $\mathcal{A}$, $\mathbf{x}_1$. Specifically, every time $\mathcal{D}$ receives the perturbed image from $\mathcal{A}$, $\mathcal{D}$ encodes the image using its VAE, and use the encoded latent vector (i.e., the mean vector of the encoded latent distribution) as the input from $\mathcal{A}$ in the current iteration (i.e., $\mathbf{x}_{1,t}$). As a result, in every iteration, $\mathcal{D}$ naturally gains access to the action of $\mathcal{A}$ in the current iteration $\mathbf{x}_{1,t}$ and can thus reason at level 1 by best-responding to $\mathbf{x}_{1,t}$. Therefore, $\mathcal{D}$ has natural access to $\mathcal{A}$'s history of selected actions, which, combined with the fact that the game is constant-sum (which allows $\mathcal{D}$ to know $\mathcal{A}$'s payoff by observing $\mathcal{D}$'s own payoff), satisfies the requirement of perfect monitoring. Note that Parsimonious maximizes the loss (instead of the attack score as in the experiments in Section 4.2.1) of a test image as the objective of attack, so to be consistent with their algorithm, we use the negative loss as the payoff function of our level-1 R2-B2 defender. Refer to Fig. 13 for the loss values achieved by Parsimonious with and without our level-1 R2-B2 defender for some selected images. The losses for different images are reported individually since they are highly disparate across different images, thus making their average losses hard to visualize.

(b) Against the BO Attacker

In addition to evaluating the effectiveness of our level-1 R2-B2 defender using the state-of-the-art Parsimonious algorithm (Section 4.2.2), we also investigate whether our level-1 R2-B2 defender is able to defend against black-box adversarial attacks using BO, which has recently become popular as a sample-efficient black-box method for adversarial attacks (Ru et al., 2020). Specifically, as a gradient-free technique to optimize black-box functions, BO can be naturally used to maximize the attack score (i.e., the output) over the space of adversarial perturbations (i.e., the input). Note that in contrast to the attacker in Section 4.2.1, the BO attacker here is not aware of the existence of the defender and thus the input to its GP surrogate only consists of the (encoded) perturbations of the attacker. We adopt two commonly used acquisition functions for BO: (a) Thompson sampling (TS) which, as a randomized algorithm, corresponds to a level-0 mixed strategy, and (b) GP-UCB, which represents a level-0 pure strategy. For both types of adversarial attacks, we let our level-1 defender run the R2-B2-Lite algorithm. In particular, when the attacker uses the GP-UCB acquisition function, in each iteration, the defender calculates/simulates the action (perturbations) that would be selected by the attacker in the current iteration, and best-responds to it; when TS is adopted by the attacker as the acquisition function, the defender draws a sample using the attacker's randomized level-0 TS strategy in the current iteration, and best-responds to it. Fig. 14 shows the results of adversarial attacks using the TS and GP-UCB acquisition functions with and without our level-1 R2-B2-Lite defender. As

[18] https://github.com/snu-mllab/parsimonious-blackbox-attack

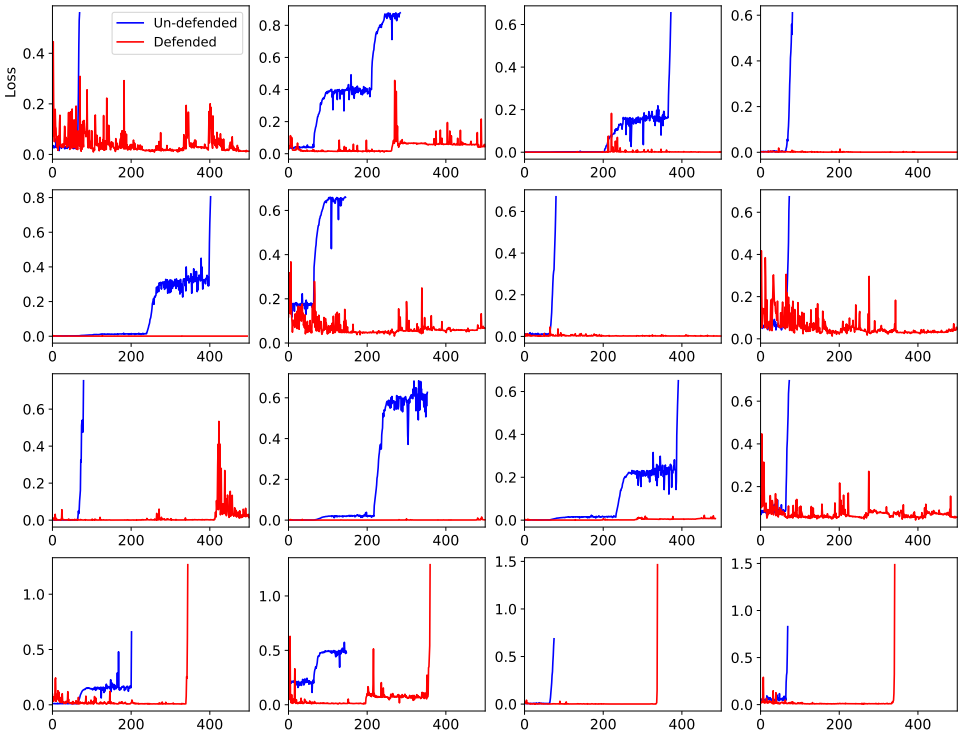


Figure 13. The loss of the Parsimonious algorithm with and without our level-1 R2-B2 defender on some selected images. For the images on the first three rows, Parsimonious fails to achieve any successful attack; for the images on the last row, our level-1 R2-B2 defender requires Parsimonious to use a significantly larger number of queries to obtain a successful attack.

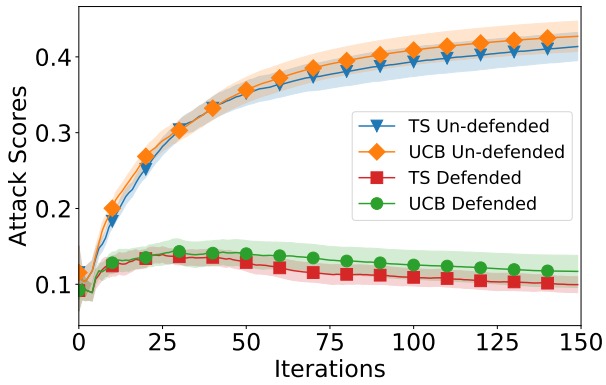


Figure 14. Attack scores achieved by the black-box attacker using BO with the GP-UCB and Thompson sampling acquisition functions, with and without our level-1 R2-B2-Lite defender. The corresponding number of successful attacks are 70.1, 67.0, 0.8 and 0.7 respectively (in the order of the figure legend from top to bottom).

demonstrated in the figure, our level-1 R2-B2-Lite defender is able to effectively defend against and almost eliminate the impact of both types of adversarial attacks (i.e., allow the attacker to succeed for less than once over 150 iterations).

F.3. Multi-Agent Reinforcement Learning

The multi-agent particle environment adopted in our experiment can be found at https://github.com/openai/multiagent-particle-envs. The state and action of the two predators (referred to as predator 1 and predator 2 for simplicity), are represented by a 14-dimensional vector and a 5-dimensional vector respectively, whereas the state and action of the prey are represented by a 12-dimensional vector and a 5-dimensional vector correspondingly. For simplicity, we perform direct policy search using a linear policy space. That is, the policy of each predator is represented by a 14×5 matrix, which maps a 14-dimensional state vector to a 5-dimensional action vector, thus producing the action to be taken by the predator according to the current policy when the predator is in a particular state. Similarly, the policy of the prey corresponds to a 12×5 matrix, which is able to map a 12-dimensional state vector to a 5-dimensional action vector. To further simplify the setting and reduce the dimensionality of the policy space, we use rank-1 approximations of the policy matrices. That is, the 14×5 policy matrix of each predator is obtained by the outer product of a 14-dimensional vector and a 5-dimensional vector, whereas the 12×5 policy matrix of the prey is attained by the outer product of a 12-dimensional vector and a 5-dimensional vector. As a result, the policy of each predator is represented by $14 + 5 = 19$ parameters, whereas the policy of the prey is characterized by $12 + 5 = 17$ parameters. Therefore, the dimension of the input to the GP surrogate models is $19 + 19 + 17 = 55$. For every one of the 55 input dimensions, the search space is $[-1, 1]$. In each iteration of the repeated game, after all agents have selected their policy parameters, the agents use their respective policies to interact in the environment for 50 steps and use their obtained returns (i.e., cumulative rewards) as the corresponding payoff; every iteration of the repeated game involves 5 independent runs in the environment (with different initializations) using the selected policy parameters, and the averaged return over the 5 independent runs is reported as the corresponding observed payoff. For ease of visualization, the returns are clipped and scaled into the range $[0, 1]$. All agents use random search as the level-0 strategy due to the high dimension of input action space; refer to Appendix F.2.1a for a detailed explanation about this choice. The GP hyperparameters are optimized via maximizing the marginal likelihood after every 10 iterations.

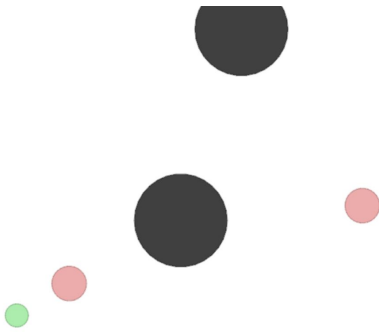


Figure 15. Illustration of the predator-prey game. Red: predators; green: prey; black: obstacles.