R2-B2: Recursive Reasoning-Based Bayesian Optimization for No-Regret Learning in Games

Zhongxiang Dai [1] **Yizhou Chen** [1] **Bryan Kian Hsiang Low** [1] **Patrick Jaillet** [2] **Teck-Hua Ho** [3]

Abstract

This paper presents a recursive reasoning formalism of *Bayesian optimization* (BO) to model the reasoning process in the interactions between boundedly rational, self-interested agents with unknown, complex, and costly-to-evaluate payoff functions in repeated games, which we call *Recursive Reasoning-Based BO* (R2-B2). Our R2-B2 algorithm is general in that it does not constrain the relationship among the payoff functions of different agents and can thus be applied to various types of games such as constant-sum, general-sum, and common-payoff games. We prove that by reasoning at level 2 or more and at one level higher than the other agents, our R2-B2 agent can achieve faster asymptotic convergence to no regret than that without utilizing recursive reasoning. We also propose a computationally cheaper variant of R2-B2 called R2-B2-Lite at the expense of a weaker convergence guarantee. The performance and generality of our R2-B2 algorithm are empirically demonstrated using synthetic games, adversarial machine learning, and multi-agent reinforcement learning.

1. Introduction

Several fundamental machine learning tasks in the real world involve intricate interactions between boundedly rational[1], self-interested agents that can be modeled as a form of repeated games with unknown, complex, and costly-to-evaluate payoff functions for the agents. For example, in adversarial *machine learning* (ML), the interactions between the *defender* $\mathcal{D}$ and the *attacker* $\mathcal{A}$ of an ML model can be modeled as a repeated game in which the payoffs to $\mathcal{D}$ and $\mathcal{A}$ are the performance of the ML model (e.g., validation accuracy) and its negation, respectively. Specifically, given a fully trained image classification model (say, provided as an online service), $\mathcal{A}$ attempts to fool the ML model into misclassification through repeated queries of the model using perturbed input images. On the other hand, for each queried image that is perturbed by $\mathcal{A}$, $\mathcal{D}$ tries to ensure the correctness of its classification by transforming the perturbed image before feeding it into the ML model. As another example, *multi-agent reinforcement learning* (MARL) in an episodic environment can also be modeled as a repeated game in which the payoff to each agent is its return from the execution of all the agents' selected policies.

Solving such a form of repeated games in a cost-efficient manner is challenging since the payoff functions of the agents are unknown, complex (e.g., possibly noisy, nonconvex, and/or with no closed-form expression/derivative), and costly to evaluate. Fortunately, the payoffs corresponding to different actions of each agent tend to be correlated. For example, in adversarial ML, the correlated perturbations performed by the attacker $\mathcal{A}$ (and correlated transformations executed by the defender $\mathcal{D}$) are likely to induce similar effects on the performance of the ML model. Such a correlation can be leveraged to *predict* the payoff associated with any action of an agent using a *surrogate* model such as the rich class of Bayesian nonparametric *Gaussian process* (GP) models (Rasmussen & Williams, 2006) which is expressive enough to represent a predictive belief of the unknown, complex payoff function over the action space of the agent. Then, in each iteration, the agent can select an action for evaluating its unknown payoff function that trades off between sampling at or near to a likely maximum payoff based on the current GP belief (exploitation) vs. improving the GP belief (exploration) until its cost/sampling budget is expended. To do this, the agent can use a sequential black-box optimizer such as the celebrated *Bayesian optimization* (BO) algorithm (Shahriari et al., 2016) based on the *GP-upper confidence bound* (GP-UCB) acquisition function (Srinivas et al., 2010), which guarantees asymptotic no-regret performance and is sample-efficient in practice. How then can

[1]Department of Computer Science, National University of Singapore, Republic of Singapore [2]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA [3]NUS Business School, National University of Singapore, Republic of Singapore. Correspondence to: Bryan Kian Hsiang Low <lowkh@comp.nus.edu.sg>.

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

[1]Boundedly rational agents are subject to limited cognition and time in making decisions (Gigerenzer & Selten, 2002).

we design a BO algorithm to account for its interactions with boundedly rational[1], self-interested agents and still guarantee the trademark asymptotic no-regret performance?

Inspired by the cognitive hierarchy model of games (Camerer et al., 2004), we adopt a recursive reasoning formalism (i.e., typical among humans) to model the reasoning process in the interactions between boundedly rational[1], self-interested agents. It comprises k levels of reasoning which represents the cognitive limit of the agent. At level $k = 0$ of reasoning, the agent randomizes its choice of actions. At a higher level $k \geq 1$ of reasoning, the agent selects its best response to the actions of the other agents who are reasoning at lower levels $0, 1, \ldots, k-1$.

This paper presents the first recursive reasoning formalism of BO to model the reasoning process in the interactions between boundedly rational[1], self-interested agents with unknown, complex, and costly-to-evaluate payoff functions in repeated games, which we call *R̲ecursive R̲easoning-B̲ased B̲O* (R2-B2) (Section 3). R2-B2 provides these agents with principled strategies for performing effectively in this type of game. In this paper, we consider repeated games with simultaneous moves and perfect monitoring[2]. Our R2-B2 algorithm is general in that it does not constrain the relationship among the payoff functions of different agents and can thus be applied to various types of games such as constant-sum games (e.g., adversarial ML in which the attacker $\mathcal{A}$ and defender $\mathcal{D}$ have opposing objectives), general-sum games (e.g., MARL where all agents have possibly different yet not necessarily conflicting goals), and common-payoff games (i.e., all agents have identical payoff functions). We prove that by reasoning at level $k \geq 2$ and one level higher than the other agents, our R2-B2 agent can achieve faster asymptotic convergence to no regret than that without utilizing recursive reasoning (Section 3.1.3). We also propose a computationally cheaper variant of R2-B2 called R2-B2-Lite at the expense of a weaker convergence guarantee (Section 3.2). The performance and generality of R2-B2 are demonstrated through extensive experiments using synthetic games, adversarial ML, and MARL (Section 4). Interestingly, we empirically show that by reasoning at a higher level, our R2-B2 defender is able to effectively defend against the attacks from the state-of-the-art black-box adversarial attackers (Section 4.2.2), which can be of independent interest to the adversarial ML community.

[2]In each iteration of a repeated game with (a) simultaneous moves and (b) perfect monitoring, every agent, respectively, (a) chooses its action simultaneously without knowing the other agents' selected actions, and (b) has access to the entire history of game plays, which includes all actions selected and payoffs observed by every agent in the previous iterations.

2. Background and Problem Formulation

For simplicity, we will mostly focus on repeated games between two agents, but have extended our R2-B2 algorithm to games involving *more than two* agents, as detailed in Appendix B. To ease exposition, throughout this paper, we will use adversarial ML as the running example and thus refer to the two agents as the *attacker* $\mathcal{A}$ and the *defender* $\mathcal{D}$. For example, the input action space $\mathcal{X}_1 \subset \mathbb{R}^{d_1}$ of $\mathcal{A}$ can be a set of allowed perturbations of a test image while the input action space $\mathcal{X}_2 \subset \mathbb{R}^{d_2}$ of $\mathcal{D}$ can represent a set of feasible transformations of the perturbed test image. We consider both input domains $\mathcal{X}_1$ and $\mathcal{X}_2$ to be discrete for simplicity; generalization of our theoretical results in Section 3 to continuous, compact domains can be easily achieved through a suitable discretization of the domains (Srinivas et al., 2010). When the ML model is an image classification model, the payoff function $f_1 : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ of $\mathcal{A}$, which takes in its perturbation $\mathbf{x}_1 \in \mathcal{X}_1$ and $\mathcal{D}$'s transformation $\mathbf{x}_2 \in \mathcal{X}_2$ as inputs, can be the maximum predictive probability among all incorrect classes for a test image since $\mathcal{A}$ intends to cause misclassification. Since $\mathcal{A}$ and $\mathcal{D}$ have opposing objectives (i.e., $\mathcal{D}$ intends to prevent misclassification), the payoff function $f_2 : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ of $\mathcal{D}$ can be the negation of that of $\mathcal{A}$, thus resulting in a constant-sum game between $\mathcal{A}$ and $\mathcal{D}$.

In each iteration $t = 1, \ldots, T$ of the repeated game with simultaneous moves and perfect monitoring[2,3], $\mathcal{A}$ and $\mathcal{D}$ select their respective input actions $\mathbf{x}_{1,t}$ and $\mathbf{x}_{2,t}$ simultaneously using our R2-B2 algorithm (Section 3) for evaluating their payoff functions f_1 and f_2. Then, $\mathcal{A}$ and $\mathcal{D}$ receive the respective noisy observed payoffs $y_{1,t} \triangleq f_1(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}) + \epsilon_1$ and $y_{2,t} \triangleq f_2(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}) + \epsilon_2$ with i.i.d. Gaussian noises $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ and noise variances σ_i^2 for $i = 1, 2$.

A common practice in game theory is to measure the performance of $\mathcal{A}$ via its *(external) regret* (Nisan et al., 2007):

$$R_{1,T} \triangleq \sum_{t=1}^{T} [f_1(\mathbf{x}_1^*, \mathbf{x}_{2,t}) - f_1(\mathbf{x}_{1,t}, \mathbf{x}_{2,t})] \quad (1)$$

where $\mathbf{x}_1^* \triangleq \arg\max_{\mathbf{x}_1 \in \mathcal{X}_1} \sum_{t=1}^{T} f_1(\mathbf{x}_1, \mathbf{x}_{2,t})$. The external regret $R_{2,T}$ of $\mathcal{D}$ is defined in a similar manner. An algorithm is said to achieve asymptotic *no regret* if $R_{1,T}$ grows sub-linearly in T, i.e., $\lim_{T \to \infty} R_{1,T}/T = 0$. Intuitively, by following a no-regret algorithm, $\mathcal{A}$ is guaranteed to eventually find its optimal input action $\mathbf{x}_1^*$ in hindsight, regardless of $\mathcal{D}$'s sequence of input actions.

To guarantee no regret (Section 3), $\mathcal{A}$ represents a predictive belief of its unknown, complex payoff function f_1 using the rich class of *Gaussian process* (GP) models by modeling f_1 as a sample of a GP (Rasmussen & Williams, 2006). $\mathcal{D}$ does likewise with its unknown f_2. Interested readers are

[3]Note that in some tasks such as adversarial ML, the requirement of perfect monitoring can be relaxed considerably. Refer to Section 4.2.2 for more details.

referred to Appendix A.1 for a detailed background on GP. In particular, $\mathcal{A}$ uses the GP predictive/posterior belief of f_1 to compute a probabilistic upper bound of f_1 called the *GP-upper confidence bound* (GP-UCB) (Srinivas et al., 2010) at any joint input actions $(\mathbf{x}_1, \mathbf{x}_2)$, which will be exploited by our R2-B2 algorithm (Section 3):

$$\alpha_{1,t}(\mathbf{x}_1, \mathbf{x}_2) \triangleq \mu_{t-1}(\mathbf{x}_1, \mathbf{x}_2) + \beta_t^{1/2} \sigma_{t-1}(\mathbf{x}_1, \mathbf{x}_2) \quad (2)$$

for iteration t where $\mu_{t-1}(\mathbf{x}_1, \mathbf{x}_2)$ and $\sigma^2_{t-1}(\mathbf{x}_1, \mathbf{x}_2)$ denote, respectively, the GP posterior mean and variance at $(\mathbf{x}_1, \mathbf{x}_2)$ (Appendix A.1) conditioned on the history of game plays up till iteration $t-1$ that includes $\mathcal{A}$'s observed payoffs and the actions selected by both agents in iterations $1, \ldots, t-1$. The GP-UCB acquisition function $\alpha_{2,t}$ for $\mathcal{D}$ is defined likewise. Supposing $\mathcal{A}$ knows the input action $\mathbf{x}_{2,t}$ selected by $\mathcal{D}$ and chooses an input action $\mathbf{x}_1$ to maximize the GP-UCB acquisition function $\alpha_{1,t}$ (2), its choice involves trading off between sampling close to an expected maximum payoff (i.e., with large GP posterior mean) given the current GP belief of f_1 (exploitation) vs. that of high predictive uncertainty (i.e., with large GP posterior variance) to improve the GP belief of f_1 (exploration) where the parameter β_t is set to trade off between exploitation vs. exploration for bounding its external regret (1), as specified later in Theorem 1.

3. Recursive Reasoning-Based Bayesian Optimization (R2-B2)

Algorithm 1 describes the R2-B2 algorithm from the perspective of *attacker* $\mathcal{A}$ which we will adopt in this section. Our R2-B2 algorithm for *defender* $\mathcal{D}$ can be derived analogously. We will now discuss the recursive reasoning formalism of BO for $\mathcal{A}$'s action selection in step 2 of Algorithm 1.

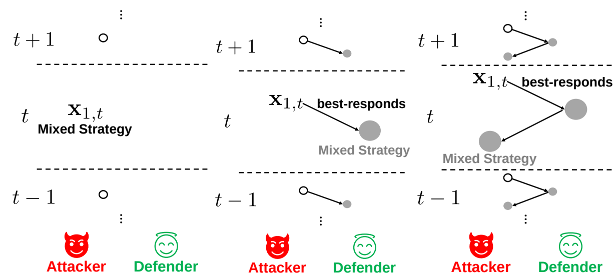
3.1. Recursive Reasoning Formalism of BO

Our recursive reasoning formalism of BO follows a similar principle as the cognitive hierarchy model (Camerer et al., 2004): At level $k = 0$ of reasoning, $\mathcal{A}$ adopts some randomized/mixed strategy of selecting its action. At level $k \geq 1$ of reasoning, $\mathcal{A}$ best-responds to the strategy of $\mathcal{D}$ who is reasoning at a lower level. Let $\mathbf{x}^k_{1,t}$ denote the input action $\mathbf{x}_{1,t}$ selected by $\mathcal{A}$'s strategy from reasoning at level k in iteration t. Depending on the (a) degree of knowledge about $\mathcal{D}$ and (b) available computational resource, $\mathcal{A}$ can choose one of the following three types of strategies of selecting its action with varying levels of reasoning, as shown in Fig. 1:

Level-$k = 0$ Strategy. Without knowledge of $\mathcal{D}$'s level of reasoning nor its level-0 strategy, $\mathcal{A}$ by default can reason at level 0 and play a mixed strategy $\mathcal{P}^0_{1,t}$ of selecting its action by sampling $\mathbf{x}^0_{1,t}$ from the probability distribution $\mathcal{P}^0_{1,t}$ over its input action space $\mathcal{X}_1$, as discussed in Section 3.1.1.

Algorithm 1 R2-B2 for attacker $\mathcal{A}$'s level-k reasoning

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Select input action $\mathbf{x}_{1,t}$ using its level-k strategy (while defender $\mathcal{D}$ selects input action $\mathbf{x}_{2,t}$)
3: Observe noisy payoff $y_{1,t} = f_1(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}) + \epsilon_1$
4: Update GP posterior belief using $\langle(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}), y_{1,t}\rangle$



(a) Level 0 (b) Level 1 (c) Level 2

Figure 1. Illustration of attacker $\mathcal{A}$'s strategies of selecting its input action from reasoning at levels $k = 0$, 1, and 2.

Level-$k = 1$ Strategy. If $\mathcal{A}$ thinks that $\mathcal{D}$ reasons at level 0 and has knowledge of $\mathcal{D}$'s level-0 mixed strategy $\mathcal{P}^0_{2,t}$, then $\mathcal{A}$ can reason at level 1 and play a pure strategy that best-responds to the level-0 strategy of $\mathcal{D}$, as explained in Section 3.1.2. Such a level-1 reasoning of $\mathcal{A}$ is general since it caters to *any* level-0 strategy of $\mathcal{D}$ and hence does not require $\mathcal{D}$ to perform recursive reasoning.

Level-$k \geq 2$ Strategy. If $\mathcal{A}$ thinks that $\mathcal{D}$ reasons at level $k-1$, then $\mathcal{A}$ can reason at level k and play a pure strategy that best-responds to $\mathcal{D}$'s level-$(k-1)$ action, as detailed in Section 3.1.3. Different from the level-1 reasoning of $\mathcal{A}$, its level-k reasoning assumes that $\mathcal{D}$'s level-$(k-1)$ action is derived using the same recursive reasoning process.

3.1.1. Level-$k = 0$ Strategy

Level 0 is a conservative, default choice for $\mathcal{A}$ since it does not require *any* knowledge about $\mathcal{D}$'s strategy of selecting its input action and is computationally lightweight. At level 0, $\mathcal{A}$ plays a mixed strategy $\mathcal{P}^0_{1,t}$ by sampling $\mathbf{x}^0_{1,t}$ from the probability distribution $\mathcal{P}^0_{1,t}$ over its input action space $\mathcal{X}_1$: $\mathbf{x}^0_{1,t} \sim \mathcal{P}^0_{1,t}$. A mixed/randomized strategy (instead of a pure/deterministic strategy) is considered because without knowledge of $\mathcal{D}$'s strategy, $\mathcal{A}$ has to treat $\mathcal{D}$ as a black-box adversary. This setting corresponds to that of an *adversarial bandit* problem in which any deterministic strategy suffers from linear worst-case regret (Lattimore & Szepesvári, 2020) and *randomization* alleviates this issue. Such a randomized design of our level-0 strategies is consistent with that of the cognitive hierarchy model in which a level-0 thinker does not make any assumption about the other agent and selects its action via a probability distribution without using strategic thinking (Camerer et al., 2004). We will now present a few reasonable choices of level-0 mixed strategies. However, in both theory (Theorems 2, 3 and 4) and practice, *any* strategy of action selection (including existing methods (Section 4.2.2)) can be considered as a level-0 strategy.

In the simplest setting where $\mathcal{A}$ has no knowledge of $\mathcal{D}$'s strategy, a natural choice for its level-0 mixed strategy is *random search*. That is, $\mathcal{A}$ samples its action from a uniform distribution over $\mathcal{X}_1$. An alternative choice is to use the *EXP3 algorithm* for the adversarial linear bandit problem, which requires the GP to be transformed via a random features approximation (Rahimi & Recht, 2007) into linear regression with random features as inputs. Since the regret of EXP3 algorithm is bounded from above by $\mathcal{O}(\sqrt{d'_1 T \log|\mathcal{X}_1|})$ (Lattimore & Szepesvári, 2020) where d'_1 denotes the number of random features, it incurs sublinear regret and can thus achieve asymptotic no regret.

In a more relaxed setting where $\mathcal{A}$ has access to the history of actions selected by $\mathcal{D}$, $\mathcal{A}$ can use the *GP-MW algorithm* (Sessa et al., 2019) to derive its level-0 mixed strategy; for completeness, GP-MW is briefly described in Appendix A.2. The result below bounds the regret of $\mathcal{A}$ when using GP-MW for level-0 reasoning and its proof is slightly modified from that of Sessa et al. (2019) to account for its payoff function f_1 being sampled from a GP (Section 2):

Theorem 1. *Let $\delta \in (0,1)$, $\beta_t \triangleq 2\log(|\mathcal{X}_1|t^2\pi^2/(3\delta))$, and γ_T denotes the maximum information gain about payoff function f_1 from any history of actions selected by both agents and corresponding noisy payoffs observed by $\mathcal{A}$ up till iteration T. Suppose that $\mathcal{A}$ uses GP-MW to derive its level-0 strategy. Then, with probability of at least $1-\delta$,*

$$R_{1,T} = \mathcal{O}(\sqrt{T\log|\mathcal{X}_1|} + \sqrt{T\log(2/\delta)} + \sqrt{T\beta_T\gamma_T}) \ .$$

From Theorem 1, $R_{1,T}$ is sub-linear in T.[4] So, $\mathcal{A}$ using GP-MW for level-0 reasoning achieves asymptotic no regret.

3.1.2. Level-$k = 1$ Strategy

If $\mathcal{A}$ thinks that $\mathcal{D}$ reasons at level 0 and has knowledge of $\mathcal{D}$'s level-0 strategy $\mathcal{P}^0_{2,t}$, then $\mathcal{A}$ can reason at level 1. Specifically, $\mathcal{A}$ selects its level-1 action $\mathbf{x}^1_{1,t}$ that maximizes the expected value of GP-UCB (2) w.r.t. $\mathcal{D}$'s level-0 strategy:

$$\mathbf{x}^1_{1,t} \triangleq \arg\max_{\mathbf{x}_1\in\mathcal{X}_1} \mathbb{E}_{\mathbf{x}^0_{2,t}\sim\mathcal{P}^0_{2,t}}[\alpha_{1,t}(\mathbf{x}_1,\mathbf{x}^0_{2,t})] \ . \quad (3)$$

If input action space $\mathcal{X}_2$ of $\mathcal{D}$ is discrete and not too large, then (3) can be solved exactly. Otherwise, (3) can be solved approximately via sampling from $\mathcal{P}^0_{2,t}$. Such a level-1 reasoning of $\mathcal{A}$ to solve (3) only requires access to the history of actions selected by $\mathcal{D}$ but not its observed payoffs, which is the same as that needed by GP-MW. Our first main result (see its proof in Appendix C) bounds the expected regret of $\mathcal{A}$ when using R2-B2 for level-1 reasoning:

Theorem 2. *Let $\delta \in (0,1)$ and $C_1 \triangleq 8/\log(1+\sigma_1^{-2})$. Suppose that $\mathcal{A}$ uses R2-B2 (Algorithm 1) for level-1 reasoning and $\mathcal{D}$ uses mixed strategy $\mathcal{P}^0_{2,t}$ for level-0 reasoning. Then, with probability of at least $1-\delta$, $\mathbb{E}[R_{1,T}] \le \sqrt{C_1 T\beta_T\gamma_T}$ where the expectation is with respect to the history of actions selected and payoffs observed by $\mathcal{D}$.*

It follows from Theorem 2 that $\mathbb{E}[R_{1,T}]$ is sublinear in T.[4] So, $\mathcal{A}$ using R2-B2 for level-1 reasoning achieves asymptotic no expected regret, which holds for *any* level-0 strategy of $\mathcal{D}$ regardless of whether $\mathcal{D}$ performs recursive reasoning.

3.1.3. Level-$k \ge 2$ Strategy

If $\mathcal{A}$ thinks that $\mathcal{D}$ reasons at level 1, then $\mathcal{A}$ can reason at level 2 and select its level-2 action $\mathbf{x}^2_{1,t}$ (4) to best-respond to level-1 action $\mathbf{x}^1_{2,t}$ (5) selected by $\mathcal{D}$, the latter of which can be computed/simulated by $\mathcal{A}$ in a similar manner as (3):

$$\mathbf{x}^2_{1,t} \triangleq \arg\max_{\mathbf{x}_1\in\mathcal{X}_1} \alpha_{1,t}(\mathbf{x}_1,\mathbf{x}^1_{2,t}) \ , \quad (4)$$

$$\mathbf{x}^1_{2,t} \triangleq \arg\max_{\mathbf{x}_2\in\mathcal{X}_2} \mathbb{E}_{\mathbf{x}^0_{1,t}\sim\mathcal{P}^0_{1,t}}[\alpha_{2,t}(\mathbf{x}^0_{1,t},\mathbf{x}_2)] \ . \quad (5)$$

In the general case, if $\mathcal{A}$ thinks that $\mathcal{D}$ reasons at level $k-1 \ge 2$, then $\mathcal{A}$ can reason at level $k \ge 3$ and select its level-k action $\mathbf{x}^k_{1,t}$ (6) that best-responds to level-$(k-1)$ action $\mathbf{x}^{k-1}_{2,t}$ (7) selected by $\mathcal{D}$:

$$\mathbf{x}^k_{1,t} \triangleq \arg\max_{\mathbf{x}_1\in\mathcal{X}_1} \alpha_{1,t}(\mathbf{x}_1,\mathbf{x}^{k-1}_{2,t}) \ , \quad (6)$$

$$\mathbf{x}^{k-1}_{2,t} \triangleq \arg\max_{\mathbf{x}_2\in\mathcal{X}_2} \alpha_{2,t}(\mathbf{x}^{k-2}_{1,t},\mathbf{x}_2) \ . \quad (7)$$

Since $\mathcal{A}$ thinks that $\mathcal{D}$'s level-$(k-1)$ action $\mathbf{x}^{k-1}_{2,t}$ (7) is derived using the same recursive reasoning process, $\mathbf{x}^{k-1}_{2,t}$ best-responds to level-$(k-2)$ action $\mathbf{x}^{k-2}_{1,t}$ selected by $\mathcal{A}$, the latter of which in turn best-responds to level-$(k-3)$ action $\mathbf{x}^{k-3}_{2,t}$ selected by $\mathcal{D}$ and can be computed in the same way as (6). This recursive reasoning process continues until it reaches the base case of the level-1 action selected by either (a) $\mathcal{A}$ (3) if k is odd (in this case, recall from Section 3.1.2 that $\mathcal{A}$ requires knowledge of $\mathcal{D}$'s level-0 strategy $\mathcal{P}^0_{2,t}$ to compute (3)), or (b) $\mathcal{D}$ (5) if k is even. Note that $\mathcal{A}$ has to perform the computations made by $\mathcal{D}$ to derive $\mathbf{x}^{k-1}_{2,t}$ (7) as well as the computations to best-respond to $\mathbf{x}^{k-1}_{2,t}$ via (6). Our next main result (see its proof in Appendix C) bounds the regret of $\mathcal{A}$ when using R2-B2 for level-$k \ge 2$ reasoning:

Theorem 3. *Let $\delta \in (0,1)$. Suppose that $\mathcal{A}$ and $\mathcal{D}$ use R2-B2 (Algorithm 1) for level-$k \ge 2$ and level-$(k-1)$ reasoning, respectively. Then, with probability of at least $1-\delta$, $R_{1,T} \le \sqrt{C_1 T\beta_T\gamma_T}$.*

[4] The asymptotic growth of γ_T has been analyzed for some commonly used kernels: $\gamma_T = \mathcal{O}((\log T)^{d_1+1})$ for squared exponential kernel and $\gamma_T = \mathcal{O}(T^{d_1(d_1+1)/(2\nu+d_1(d_1+1))}\log T)$ for Matérn kernel with parameter $\nu > 1$. For both kernels, the last term in the regret bound in Theorem 1 grows sub-linearly in T.

Theorem 3 reveals that $R_{1,T}$ grows sublinearly in T.[4] So, $\mathcal{A}$ using R2-B2 for level-$k \geq 2$ reasoning achieves asymptotic no regret regardless of $\mathcal{D}$'s level-0 strategy $\mathcal{P}^0_{2,t}$. By comparing Theorems 1 and 3, we can observe that if $\mathcal{A}$ uses GP-MW as its level-0 strategy, then it can achieve faster asymptotic convergence to no regret by using R2-B2 to reason at level $k \geq 2$ and one level higher than $\mathcal{D}$. However, when $\mathcal{A}$ reasons at a higher level k, its computational cost grows due to an additional optimization of the GP-UCB acquisition function per increase in level of reasoning. So, $\mathcal{A}$ is expected to favor reasoning at a lower level, which agrees with the observation in the work of Camerer et al. (2004) on the cognitive hierarchy model that humans usually reason at a level no higher than 2.

3.2. R2-B2-Lite

We also propose a computationally cheaper variant of R2-B2 for level-1 reasoning called R2-B2-Lite at the expense of a weaker convergence guarantee. When using R2-B2-Lite for level-1 reasoning, instead of following (3), $\mathcal{A}$ selects its level-1 action $\mathbf{x}^1_{1,t}$ by sampling $\widetilde{\mathbf{x}}^0_{2,t}$ from level-0 strategy $\mathcal{P}^0_{2,t}$ of $\mathcal{D}$ and best-responding to this sampled action:

$$\mathbf{x}^1_{1,t} \triangleq \arg\max_{\mathbf{x}_1 \in \mathcal{X}_1} \alpha_{1,t}(\mathbf{x}_1, \widetilde{\mathbf{x}}^0_{2,t}) \,. \tag{8}$$

Our final main result (its proof is in Appendix D) bounds the expected regret of $\mathcal{A}$ using R2-B2-Lite for level-1 reasoning:

Theorem 4. *Let $\delta \in (0,1)$. Suppose that $\mathcal{A}$ uses R2-B2-Lite for level-1 reasoning and $\mathcal{D}$ uses mixed strategy $\mathcal{P}^0_{2,t}$ for level-0 reasoning. If the trace of the covariance matrix of $\mathbf{x}^0_{2,t} \sim \mathcal{P}^0_{2,t}$ is not more than ω_t for $t = 1, \ldots, T$, then with probability of at least $1-\delta$, $\mathbb{E}[R_{1,T}] = \mathcal{O}(\sum_{t=1}^{T} \sqrt{\omega_t} + \sqrt{T\beta_T\gamma_T})$ where the expectation is with respect to the history of actions selected and payoffs observed by $\mathcal{D}$ as well as $\widetilde{\mathbf{x}}^0_{2,t}$ for $t = 1, \ldots, T$.*

From Theorem 4, the expected regret bound tightens if $\mathcal{D}$'s level-0 mixed strategy $\mathcal{P}^0_{2,t}$ has a smaller variance for each dimension of input action $\mathbf{x}^0_{2,t}$. As a result, the level-0 action $\widetilde{\mathbf{x}}^0_{2,t}$ of $\mathcal{D}$ that is sampled by $\mathcal{A}$ tends to be closer to the true level-0 action $\mathbf{x}^0_{2,t}$ selected by $\mathcal{D}$. Then, $\mathcal{A}$ can select level-1 action $\mathbf{x}^1_{1,t}$ that best-responds to a more precise estimate $\widetilde{\mathbf{x}}^0_{2,t}$ of the level-0 action $\mathbf{x}^0_{2,t}$ selected by $\mathcal{D}$, hence improving its expected payoff. Theorem 4 also reveals that $\mathcal{A}$ using R2-B2-Lite for level-1 reasoning achieves asymptotic no expected regret if the sequence $(\omega_t)_{t\in\mathbb{Z}^+}$ uniformly decreases to 0 (i.e., $\omega_{t+1} < \omega_t$ for $t \in \mathbb{Z}^+$ and $\lim_{T\to\infty} \omega_T = 0$). Interestingly, such a sufficient condition for achieving asymptotic no expected regret has a natural and elegant interpretation in terms of the exploration-exploitation trade-off: This condition is satisfied if $\mathcal{D}$ uses a level-0 mixed strategy $\mathcal{P}^0_{2,t}$ with a decreasing variance for each dimension of input action $\mathbf{x}^0_{2,t}$, which corresponds to transitioning from exploration (i.e., a large variance results in a diffused $\mathcal{P}^0_{2,t}$ and hence many actions being sampled) to exploitation (i.e., a small variance results in a peaked $\mathcal{P}^0_{2,t}$ and hence fewer actions being sampled).

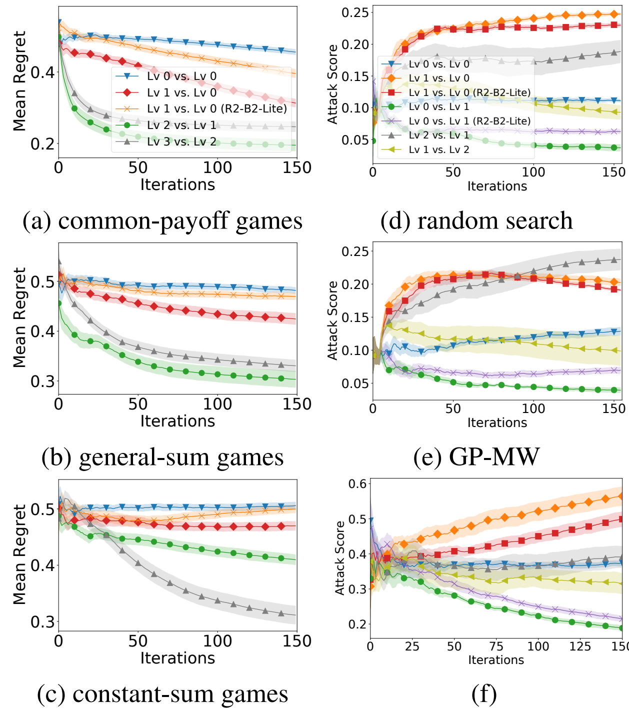


Figure 2. (a-c) Mean regret of agent 1 in synthetic games where the legend in (a) represents the levels of reasoning of agents 1 vs. 2. Attack score of $\mathcal{A}$ in adversarial ML for (d-e) MNIST and (f) CIFAR-10 datasets where the legend in (d) represents the levels of reasoning of $\mathcal{A}$ vs. $\mathcal{D}$.

4. Experiments and Discussion

This section empirically evaluates the performance of our R2-B2 algorithm and demonstrates its generality using synthetic games, adversarial ML, and MARL. Some of our experimental comparisons can be interpreted as comparisons with existing baselines used as level-0 strategies (Section 3.1.1). Specifically, we can compare the performance of our level-1 agent with that of a baseline method when they are against the same level-0 agent. Moreover, in constant-sum games, we can perform a more direct comparison by playing our level-1 agent against an opponent using a baseline method as a level-0 strategy (Section 4.2.2). Additional experimental details and results are reported in Appendix F due to lack of space. All error bars represent standard error.

4.1. Synthetic Games

Firstly, we empirically evaluate the performance of R2-B2 using synthetic games with two agents whose payoff functions are sampled from GP over a discrete input domain. Both agents use GP-MW and R2-B2/R2-B2-Lite for level-

0 and level-$k \geq 1$ reasoning, respectively. We consider 3 types of games: common-payoff, general-sum, and constant-sum games. Figs. 2a to 2c show results of the mean regret[5] of agent 1 averaged over 10 random samples of GP and 5 initializations of 1 randomly selected action with observed payoff per sample: In all types of games, when agent 1 reasons at one level higher than agent 2, it incurs a smaller mean regret than when reasoning at level 0 (blue curve), which demonstrates the performance advantage of recursive reasoning and corroborates our theoretical results (Theorems 2 and 3). The same can be observed for agent 1 using R2-B2-Lite for level-1 reasoning (orange curve) but it does not perform as well as that using R2-B2 (red curve), which again agrees with our theoretical result (Theorem 4). Moreover, comparing the red (orange) and blue curves shows that when against the same level-0 agent, our R2-B2 (R2-B2-Lite) level-1 agent outperforms the baseline method of GP-MW (as a level-0 strategy).

Figs. 2a and 2c also reveal the effect of incorrect thinking of the level of reasoning of the other agent on its performance: Since agent 2 uses recursive reasoning at level 1 or more, agent 2 thinks that it is reasoning at one level higher than agent 1. However, it is in fact reasoning at one level lower in these two figures. In common-payoff games, since agents 1 and 2 have identical payoff functions, the mean regret of agent 2 is the same as that of agent 1 in Fig. 2a. So, from agent 2's perspective, it benefits from such an incorrect thinking in common-payoff games. In constant-sum games, since the payoff function of agent 2 is negated from that of agent 1, the mean regret of agent 2 increases with a decreasing mean regret of agent 1 in Fig. 2c. So, from agent 2's viewpoint, it hurts from such an incorrect thinking in constant-sum games. Further experimental results on such incorrect thinking are reported in Appendix F.1.1b.

An intriguing observation from Figs. 2a to 2c is that when agent 1 reasons at level $k \geq 2$, it incurs a smaller mean regret than when reasoning at level 1. A possible explanation is that when agent 1 reasons at level $k \geq 2$, its selected level-k action (6) best-responds to the actual level-$(k-1)$ action (7) selected by agent 2. In contrast, when agent 1 reasons at level 1, its selected level-1 action (3) maximizes the *expected* value of GP-UCB w.r.t. agent 2's level-0 *mixed* strategy rather than the actual level-0 action selected by agent 2. However, as we shall see in the experiments on adversarial ML in Section 4.2.1, when the expectation in level-1 reasoning (3) needs to be approximated via sampling but insufficient samples are used, the performance of level-$k \geq 2$ reasoning can be potentially diminished due to propagation of the approximation error from level 1.

[5]The mean regret $T^{-1}\sum_{t=1}^{T}(\max_{\mathbf{x}_1\in\mathcal{X}_1,\mathbf{x}_2\in\mathcal{X}_2} f_1(\mathbf{x}_1,\mathbf{x}_2) - f_1(\mathbf{x}_{1,t},\mathbf{x}_{2,t}))$ of agent 1 pessimistically estimates (i.e., upper bounds) $R_{1,T}/T$ (1) and is thus not expected to converge to 0. Nevertheless, it serves as an appropriate performance metric here.

Moreover, Fig. 2c shows another interesting observation that is unique for constant-sum games: Agent 1 achieves a significantly better performance when reasoning at level 3 (i.e., agent 2 reasons at level 2) than at level 2 (i.e., agent 2 reasons at level 1). This can be explained by the fact that when agent 2 reasons at level 2, it best-responds to the level-1 action of agent 1, which is most likely different from the actual action selected by agent 1 since agent 1 is in fact reasoning at level 3. In contrast, when agent 2 reasons at level 1, instead of best-responding to a single (most likely wrong) action of agent 1, it best-responds to the expected behavior of agent 1 by attributing a distribution over all actions of agent 1. As a result, agent 2 suffers from a smaller performance deficit when reasoning at level 1 (i.e., agent 1 reasons at level 2) compared with reasoning at level 2 (i.e., agent 1 reasons at level 3) or higher. Therefore, agent 1 obtains a more dramatic performance advantage when reasoning at level 3 (gray curve) due to the constant-sum nature of the game. A deeper implication of this insight is that although level-1 reasoning may not yield a better performance than level-$k \geq 2$ reasoning as analyzed in the previous paragraph, it is more robust against incorrect estimates of the opponent's level of reasoning in constant-sum games.

Experimental results on the use of random search and EXP3 (Section 3.1.1) for level-0 reasoning (instead of GP-MW) are reported in Appendix F.1.1c; the resulting observations and insights are consistent with those presented here. This demonstrates the robustness of R2-B2 and corroborates the generality of our theoretical results (Theorems 2 and 3) which hold for any level-0 strategy of the other agent. We have also performed experiments using synthetic games involving *more than two* agents (Appendix F.1.2), which yield some interesting observations that are consistent with our theoretical analysis.

4.2. Adversarial Machine Learning (ML)

4.2.1. R2-B2 for Adversarial ML

We apply our R2-B2 algorithm to black-box adversarial ML for image classification problems with *deep neural networks* (DNNs) using the MNIST and CIFAR-10 image datasets. We consider *evasion attacks*: The attacker $\mathcal{A}$ perturbs a test image to fool a fully trained DNN (referred to as the *target ML model* hereafter) into misclassifying the image, while the defender $\mathcal{D}$ transforms the perturbed image with the goal of ensuring the correct prediction by the classifier. To improve query efficiency, dimensionality reduction techniques such as autoencoders have been commonly used for black-box adversarial attacks (Tu et al., 2019). In our experiments, *variational autoencoders* (VAE) (Kingma & Welling, 2014) are used by both $\mathcal{A}$ and $\mathcal{D}$ to project the images to a lower-dimensional space (i.e., 2D for MNIST and 8D for CIFAR-

10).[6] Following a common practice in adversarial ML, we focus on perturbations with bounded infinity norm as actions of $\mathcal{A}$ and $\mathcal{D}$: The maximum allowed perturbation to each pixel added by either $\mathcal{A}$ or $\mathcal{D}$ is no more than a pre-defined value ϵ where $\epsilon = 0.2$ for MNIST and $\epsilon = 0.05$ for CIFAR-10. We consider *untargeted attacks* whereby the goal of $\mathcal{A}$ ($\mathcal{D}$) is to cause (prevent) misclassification by the target ML model. So, the payoff function of $\mathcal{A}$ is the maximum predictive probability among all incorrect classes (referred to as *attack score* hereafter) and its negation is the payoff function of $\mathcal{D}$. As a result, the application of R2-B2 to black-box adversarial ML represents a *constant-sum game*. An attack is considered *successful* if the attack score is larger than the predictive probability of the correct class, hence resulting in misclassification of the test image. Both $\mathcal{A}$ and $\mathcal{D}$ use GP-MW/random search[7] and R2-B2/R2-B2-Lite for level-0 and level-$k \geq 1$ reasoning, respectively.

Figs. 2d to 2f show results of the attack score of $\mathcal{A}$ in adversarial ML for both image datasets while Table 1 shows results of the number of successful attacks by $\mathcal{A}$ over 150 iterations of the game; the results are averaged over 10 initializations of 5 randomly selected actions with observed payoffs.[8] It can be observed from Figs. 2d to 2f that when $\mathcal{A}$ reasons at one level higher than $\mathcal{D}$ (orange, red, and gray curves), its attack score is higher than when reasoning at level 0 (blue, green, and purple curves). Similarly, when $\mathcal{D}$ reasons at one level higher (green, purple, and yellow curves), the attack score of $\mathcal{A}$ is reduced. These observations demonstrate the performance advantage of using recursive reasoning in adversarial ML. Such an advantage of recursive reasoning can also be seen from Table 1: For MNIST, when random search is used for level-0 reasoning and $\mathcal{A}$ reasons at one level higher than $\mathcal{D}$, it achieves a larger number of successful attacks (12.8, 10.2, and 3.0) than when reasoning at level 0 (2.6, 0.8, and 1.8). Similarly, when $\mathcal{D}$ reasons at one level higher, it reduces the number of successful attacks by $\mathcal{A}$ (0.8, 1.8, and 0.9) than when reasoning at level 0 (2.6, 12.8, and 10.2). The observations are similar for MNIST with GP-MW for level-0 reasoning as well as for CIFAR-10 (Table 1).

The performance advantage of $\mathcal{A}$ reasoning at level 2 is observed to be smaller than that at level 1; this may be explained by the propagation of error of approximating the expectation in level-1 reasoning (3), as explained previously in Section 4.1. We investigate and report the effect of the number of samples for such an approximation in Appendix F.2.1c, which reveals that the performance improves with more samples, albeit with higher computational cost. Moreover, some insights can also be drawn regarding the consequence of an incorrect thinking about the opponent's level of reasoning in constant-sum games. For example, for the gray curves in Figs. 2d to 2f, $\mathcal{D}$ reasons at level 1 because it thinks that $\mathcal{A}$ reasons at level 0. However, $\mathcal{A}$ is in fact reasoning at level 2. As a result, in this constant-sum game, $\mathcal{D}$'s incorrect thinking about the opponent's level of reasoning negatively impacts $\mathcal{D}$'s performance since the attack scores are increased. This is consistent with the corresponding analysis in synthetic games regarding the effect of incorrect thinking about the level of reasoning of the other agent (Section 4.1).

Table 1. Average number of successful attacks by $\mathcal{A}$ over 150 iterations in adversarial ML for MNIST and CIFAR-10 datasets where the levels of reasoning are in the form of $\mathcal{A}$ vs. $\mathcal{D}$.

Levels of reasoning	MNIST (random)	MNIST (GP-MW)	CIFAR-10
0 vs. 0	2.6	4.3	70.1
1 vs. 0	12.8	6.0	113.1
1 vs. 0 (R2-B2-Lite)	10.2	6.8	99.7
0 vs. 1	0.8	0.4	25.2
0 vs. 1 (R2-B2-Lite)	1.8	1.0	29.7
2 vs. 1	3.0	5.2	70.9
1 vs. 2	0.9	0.4	54.0

4.2.2. Comparison with State-of-the-art Adversarial Attack Methods

It was mentioned in Section 3.1 that our theoretical results hold for *any* level-0 strategy of the other agent. So, any existing adversarial attack (defense) method can be used the level-0 strategy of $\mathcal{A}$ ($\mathcal{D}$). In this experiment, we perform a direct comparison of R2-B2 with the state-of-the-art black-box adversarial attack method called *Parsimonious* (Moon et al., 2019): We use Parsimonious as the level-0 strategy of $\mathcal{A}$ and let $\mathcal{D}$ use R2-B2 for level-1 reasoning. We consider a realistic setting where in each iteration, $\mathcal{D}$ only needs to receive the image perturbed by $\mathcal{A}$ and choose its action that best-responds to this perturbed image. In this manner, $\mathcal{D}$ naturally has access to the history of actions selected by $\mathcal{A}$ (as required by *perfect monitoring* in our repeated game) since it receives all images perturbed by $\mathcal{A}$. Additional details of the experimental setting are reported in Appendix F.2.2a.

We randomly select 70 images from the CIFAR-10 dataset that are successfully attacked by Parsimonious using $\epsilon = 0.05$ over 500 iterations without the defender $\mathcal{D}$.[9] Our level-1 R2-B2 defender manages to *completely prevent any successful attacks* for 53 of these images and requires Parsimonious to use *more than* 3.5 *times* more queries on average to

[6] We have detailed in Appendix F.2.1a how VAE can be realistically incorporated into our algorithm.

[7] For CIFAR-10 dataset, $\mathcal{A}$ uses only random search for level-0 reasoning due to high dimensions, as explained in Appendix F.2.1a.

[8] The results here use a test image from each dataset that can clearly illustrate the effects of both attack and defense. Refer to Appendix F.2.1b for more details and results using more test images; the observations are consistent with those presented here.

[9] Compared to the work of Moon et al. (2019), we use fewer iterations and a larger ϵ, which we think is more realistic as attacks with an excessively large no. of queries may be easily detected.

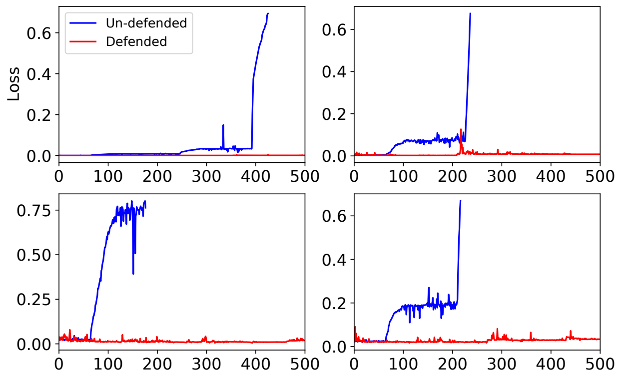


Figure 3. Loss incurred by Parsimonious with and without our level-1 R2-B2 defender on 4 randomly selected images that are successfully attacked by Parsimonious.

succeed for 10 other images.[10] Fig. 3 shows results of the loss incurred by Parsimonious (i.e., its original attack objective) with and without our level-1 R2-B2 defender for 4 of the successfully defended images; results for other images are shown in Appendix F.2.2a. This experiment not only demonstrates the generality of our R2-B2 algorithm, but can also be of significant independent interest to the adversarial ML community as a defense method against black-box adversarial attacks.

In addition, as another comparison, we use the same experimental setting with the CIFAR-10 dataset in Section 4.2.1 and play Parsimonious against a level-0 defender using random search. The results show that when against the same level-0 defender, Parsimonious achieves a significantly smaller average number of successful attacks (27.6) compared with our level-1 attacker (113.1, as shown in Table 1). In other words, our level-1 defender can defend effectively against Parsimonious, while our level-1 attacker can attack better than Parsimonious. Note that the unsatisfactory performances of Parsimonious in our experiments might be largely explained the fact that it does not consider the presence of a defender. Moreover, our level-1 R2-B2 defender can also defend against black-box adversarial attacks from standard BO algorithms (Appendix F.2.2b)[11], which have become popular recently (Ru et al., 2020).

4.3. Multi-Agent Reinforcement Learning (MARL)

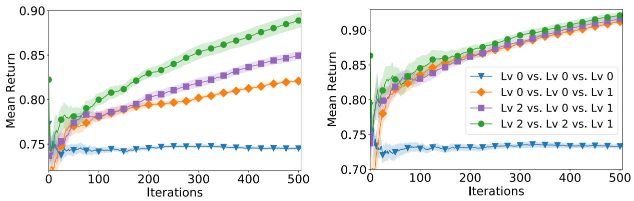
We apply R2-B2 to policy search for MARL with *more than two* agents. Each action of an agent represents a particular set of policy parameters controlling the behavior of the agent in an environment. The payoff to each agent corresponding to a selected set of its policy parameters (i.e., action) is its mean return (i.e., cumulative reward) from the execution of all the agents' selected policies across 5 independent episodes. Since the agents interact in the environment, the payoff function of each agent depends on the policies (actions) selected by all agents. We use the predator-prey game from the widely used multi-agent particle environment in (Lowe et al., 2017). This 3-agent game (see Fig. 15 in Appendix F.3) contains two predators who are trying to catch a prey. The prey is rewarded for being far from the predators and penalized for stepping outside the boundary. The two predators have identical payoff functions and are rewarded for being close to the prey (if the prey stays within the boundary). So, the predator-prey game represents a *general-sum game*. All agents use random search[12] and R2-B2 for level-0 and level-$k \geq 1$ reasoning, respectively.

Fig. 4 shows results of the (scaled) mean return of the agents averaged over 10 initializations of 5 randomly selected actions with observed payoffs. It can be observed from Fig. 4b that when the prey reasons at level 1 and both predators reason at level 0 (orange curve), its mean return is much higher than when reasoning at level 0 (blue curve); this results from the prey's ability to learn to stay within the boundary. Specifically, there exist some "dominated actions" in this game, namely, those causing the prey to step beyond the boundary. Regardless of the predators' policies, such dominated actions never give large returns to the prey and are thus likely to yield small values of GP-UCB for any actions (policies) selected by the predators. So, by reasoning at level 1 (i.e., by maximizing the expected value of GP-UCB), the prey is able to eliminate those dominated actions and thus learn to stay within the boundary. From Fig. 4a, the mean return of the predators is also improved (orange curve) because the prey's ability to stay within the boundary allows the predators to improve their rewards by being close to the prey despite using random search for level-0 reasoning. In contrast, when the prey reasons at level 0, the predators rarely get rewarded (blue curve) since the prey repeatedly steps beyond the boundary. On the other hand, when predator 1 reasons at level 2 (purple curve), the mean return of the predators is further increased since predator 1 is now able to learn to actively move close to the prey instead of moving around using random search for level-0 reasoning (orange curve). When both predators reason at level 2 (green curve), their mean return is improved even further. In both of these scenarios, the mean return of the prey stays close to that associated with the orange curve: Although the predators are able to actively approach the prey, this also further helps to prevent the prey from moving beyond the boundary, which compensates for the loss in its mean return due to the more strategic predators.

[10] The remaining 7 images are so easy to attack such that the attacks are already successful during the initial exploration phase of our level-1 R2-B2 defender.

[11] The BO attacker here only takes its perturbations as inputs and thus does not consider the defender.

[12] All agents use only random search for level-0 reasoning due to high dimensions, as explained in Appendix F.3.



(a) predators (b) prey

Figure 4. Mean return of predators and prey in predator-prey game where the legend in (b) represents the levels of reasoning of predator 1 vs. predator 2 vs. prey.

5. Related Work

The recent work of Sessa et al. (2019) combines online learning and GP-UCB to derive a no-regret learning algorithm called *GP-multiplicative weight* (GP-MW) for repeated games. As explained in Section 3.1.1, GP-MW can be used as a level-0 mixed strategy (i.e., no recursive reasoning) in our R2-B2 algorithm. Moreover, BO has also been recently applied in game theory to find the Nash equilibria (Picheny et al., 2019).

Humans possess the ability to reason about the mental states of others (Goldman, 2012). In particular, a person tends to reason recursively by analyzing the others' thinking about himself, which gives rise to recursive reasoning (Pynadath & Marsella, 2005). The recursive reasoning model of humans has inspired the development of the cognitive hierarchy model in behavioral game theory, which uses recursive reasoning to explain the behavior of players in games (Camerer et al., 2004). Moreover, the improved decision-making capability offered by recursive reasoning has motivated its application in ML and sequential decision-making problems such as interactive partially observable Markov decisionn processes (Gmytrasiewicz & Doshi, 2005; Hoang & Low, 2013), MARL (Wen et al., 2019), among others.

Deep neural networks (DNNs) have recently been found to be vulnerable to carefully crafted adversarial examples (Szegedy et al., 2014). Since then, a variety of adversarial attack methods have been developed to exploit this vulnerability of DNNs (Goodfellow et al., 2015). However, most of the existing attack methods are *white-box* attacks since they require access to the gradient of the ML model. In contrast, the more realistic *black-box attacks* (Tu et al., 2019; Moon et al., 2019), which we have adopted in our experiments, only require query access to the target ML model and have been attracting significant attention recently. Of note, BO has recently been used for black-box adversarial attacks (without considering defenses) and demonstrated promising query efficiency (Ru et al., 2020). On the other hand, many attempts have been made to design adversarial defense methods (Madry et al., 2017; Tramèr et al., 2018) to make ML models robust against adversarial attacks. In our experiments, we have adopted the input reconstruction/transformation technique (Meng & Chen, 2017; Samangouei et al., 2018) as the defense mechanism, in which the defender attempts to transform the perturbed input to ensure the correct prediction by the ML model. Refer to the detailed survey of adversarial ML in (Yuan et al., 2019).

6. Conclusion and Future Work

This paper describes the first BO algorithm called R2-B2 that is endowed with the capability of recursive reasoning to model the reasoning process in the interactions between boundedly rational[1], self-interested agents with unknown, complex, and expensive-to-evaluate payoff functions in repeated games. We prove that by reasoning at level $k \geq 2$ and one level higher than the other agents, our R2-B2 agent can achieve faster asymptotic convergence to no regret than that without utilizing recursive reasoning. We empirically demonstrate the competitive performance and generality of R2-B2 through extensive experiments using synthetic games, adversarial ML, and MARL. For our future work, we plan to investigate the connection of R2-B2 to other game-theoretic solution concepts such as Nash equilibrium. We will also explore the extension of R2-B2 to a more general setting where a level-k agent selects its best response to the action of the other agent who reasons according to a distribution (e.g., Poisson) over lower levels instead of only at level $k-1$, which is also captured by the cognitive hierarchy model (Camerer et al., 2004). We will consider generalizing R2-B2 to nonmyopic BO (Kharkovskii et al., 2020b; Ling et al., 2016), batch BO (Daxberger & Low, 2017), high-dimensional BO (Hoang et al., 2018), differentially private BO (Kharkovskii et al., 2020a), and multi-fidelity BO (Zhang et al., 2017; 2019) settings and incorporating early stopping (Dai et al., 2019). For applications with a huge budget of function evaluations, we like to couple R2-B2 with the use of distributed/decentralized (Chen et al., 2012; 2013a;b; 2015; Hoang et al., 2016; 2019b;a; Low et al., 2015; Ouyang & Low, 2018) or online/stochastic (Hoang et al., 2015; 2017; Low et al., 2014; Xu et al., 2014; Teng et al., 2020; Yu et al., 2019a;b) sparse GP models to represent the belief of the unknown objective function efficiently.

Acknowledgements

This research/project is supported in part by the Singapore National Research Foundation through the Singapore-MIT Alliance for Research and Technology (SMART) Centre for Future Urban Mobility (FM) and in part by A*STAR under its RIE2020 Advanced Manufacturing and Engineering (AME) Industry Alignment Fund – Pre Positioning (IAF-PP) (Award A19E4a0101). Teck-Hua Ho acknowledges funding from the Singapore National Research Foundation's Returning Singaporean Scientists Scheme, grant NRFRSS2014-001.

References

Camerer, C. F., Ho, T.-H., and Chong, J.-K. A cognitive hierarchy model of games. *Quarterly J. Economics*, 119 (3):861–898, 2004.

Chen, J., Low, K. H., Tan, C. K.-Y., Oran, A., Jaillet, P., Dolan, J. M., and Sukhatme, G. S. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, pp. 163–173, 2012.

Chen, J., Cao, N., Low, K. H., Ouyang, R., Tan, C. K.-Y., and Jaillet, P. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, pp. 152–161, 2013a.

Chen, J., Low, K. H., and Tan, C. K.-Y. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*, 2013b.

Chen, J., Low, K. H., Jaillet, P., and Yao, Y. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Trans. Autom. Sci. Eng.*, 12: 901–921, 2015.

Dai, Z., Yu, H., Low, K. H., and Jaillet, P. Bayesian optimization meets Bayesian optimal stopping. In *Proc. ICML*, pp. 1496–1506, 2019.

Daxberger, E. A. and Low, K. H. Distributed batch Gaussian process optimization. In *Proc. ICML*, pp. 951–960, 2017.

Gigerenzer, G. and Selten, R. *Bounded Rationality*. MIT Press, 2002.

Gmytrasiewicz, P. J. and Doshi, P. A framework for sequential planning in multi-agent settings. *J. Artif. Intell. Res.*, 24:49–79, 2005.

Goldman, A. I. Theory of mind. In Margolis, E., Samuels, R., and Stich, S. P. (eds.), *The Oxford Handbook of Philosophy of Cognitive Science*. Oxford Univ. Press, 2012.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *Proc. ICLR*, 2015.

Hoang, Q. M., Hoang, T. N., and Low, K. H. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, pp. 2007–2014, 2017.

Hoang, Q. M., Hoang, T. N., Low, K. H., and Kingsford, C. Collective model fusion for multiple black-box experts. In *Proc. ICML*, pp. 2742–2750, 2019a.

Hoang, T. N. and Low, K. H. Interactive POMDP Lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. In *Proc. IJCAI*, 2013.

Hoang, T. N., Hoang, Q. M., and Low, K. H. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, pp. 569–578, 2015.

Hoang, T. N., Hoang, Q. M., and Low, K. H. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, pp. 382–391, 2016.

Hoang, T. N., Hoang, Q. M., and Low, K. H. Decentralized high-dimensional Bayesian optimization with factor graphs. In *Proc. AAAI*, pp. 3231–3238, 2018.

Hoang, T. N., Hoang, Q. M., Low, K. H., and How, J. P. Collective online learning of Gaussian processes in massive multi-agent systems. In *Proc. AAAI*, pp. 7850–7857, 2019b.

Kharkovskii, D., Dai, Z., and Low, K. H. Private outsourced Bayesian optimization. In *Proc. ICML*, 2020a.

Kharkovskii, D., Ling, C. K., and Low, K. H. Nonmyopic Gaussian process optimization with macro-actions. In *Proc. AISTATS*, pp. 4593–4604, 2020b.

Kim, J. and Choi, S. On local optimizers of acquisition functions in Bayesian optimization. arXiv:1901.08350, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proc. ICLR*, 2014.

Lattimore, T. and Szepesvári, C. *Bandit Algorithms*. 2020.

Ling, C. K., Low, K. H., and Jaillet, P. Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. AAAI*, pp. 1860–1866, 2016.

Low, K. H., Xu, N., Chen, J., Lim, K. K., and Özgül, E. B. Generalized online sparse Gaussian processes with application to persistent mobile robot localization. In *Proc. ECML/PKDD Nectar Track*, pp. 499–503, 2014.

Low, K. H., Yu, J., Chen, J., and Jaillet, P. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*, pp. 2821–2827, 2015.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proc. NeurIPS*, pp. 6379–6390, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *Proc. ICLR*, 2017.

Meng, D. and Chen, H. MagNet: A two-pronged defense against adversarial examples. In *Proc. CCS*, pp. 135–147, 2017.

Moon, S., An, G., and Song, H. O. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *Proc. ICML*, 2019.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. *Algorithmic Game Theory*. Cambridge Univ. Press, 2007.

Ouyang, R. and Low, K. H. Gaussian process decentralized data fusion meets transfer learning in large-scale distributed cooperative perception. In *Proc. AAAI*, pp. 3876–3883, 2018.

Picheny, V., Binois, M., and Habbal, A. A Bayesian optimization approach to find Nash equilibria. *Journal of Global Optimization*, 73(1):171–192, 2019.

Pynadath, D. V. and Marsella, S. C. PsychSim: Modeling theory of mind with decision-theoretic agents. In *Proc. IJCAI*, pp. 1181–1186, 2005.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Proc. NeurIPS*, pp. 1177–1184, 2007.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Ru, B., Cobb, A., Blaas, A., and Gal, Y. BayesOpt adversarial attack. In *Proc. ICLR*, 2020.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *Proc. ICLR*, 2018.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. No-regret learning in unknown games with correlated payoffs. In *Proc. NeurIPS*, 2019.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. of the IEEE*, 104(1): 148–175, 2016.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pp. 1015–1022, 2010.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *Proc. ICLR*, 2014.

Teng, T., Chen, J., Zhang, Y., and Low, K. H. Scalable variational bayesian kernel selection for sparse Gaussian process regression. In *Proc. AAAI*, pp. 5997–6004, 2020.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In *Proc. ICLR*, 2018.

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proc. AAAI*, pp. 742–749, 2019.

Wen, Y., Yang, Y., Luo, R., Wang, J., and Pan, W. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *Proc. ICLR*, 2019.

Xu, N., Low, K. H., Chen, J., Lim, K. K., and Özgül, E. B. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, pp. 2585–2592, 2014.

Yu, H., Chen, Y., Dai, Z., Low, K. H., and Jaillet, P. Implicit posterior variational inference for deep Gaussian processes. In *Proc. NeurIPS*, pp. 14475–14486, 2019a.

Yu, H., Hoang, T. N., Low, K. H., and Jaillet, P. Stochastic variational inference for Bayesian sparse Gaussian process regression. In *Proc. IJCNN*, 2019b.

Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learning Syst.*, 30(9):2805–2824, 2019.

Zhang, Y., Hoang, T. N., Low, K. H., and Kankanhalli, M. Information-based multi-fidelity Bayesian optimization. In *Proc. NIPS Workshop on Bayesian Optimization*, 2017.

Zhang, Y., Dai, Z., and Low, K. H. Bayesian optimization with binary auxiliary information. In *Proc. UAI*, 2019.