



Analytics, Computational Intelligence and Information Management

Opponent aware reinforcement learning

V́ctor Gallego^{a,b,*}, Roi Naveiro^c, David Ŕos Insua^a, David Ǵomez-Ullate^d^a Institute of Mathematical Sciences (ICMAT), Madrid, Spain^b Komorebi AI Technologies, Madrid, Spain^c CUNEF Universidad, Madrid, Spain^d School of Science and Technology, IE University, Madrid, Spain

ARTICLE INFO

Keywords:

Artificial intelligence

Multi-agent reinforcement learning

Game theory

Level-k thinking

ABSTRACT

In certain reinforcement learning (RL) scenarios there are adversaries trying to interfere with the underlying reward process for their own benefit. We introduce *Threatened Markov Decision Processes* (TMDPs) as a framework to support an agent against potential opponents in an RL context as well as schemes resulting in novel learning approaches to deal with TMDPs. After introducing our framework and deriving theoretical results, empirical evidence is given via extensive experiments, showing the importance for an RL agent of acknowledging adversarial awareness.

1. Introduction

Over the last decade, an increasing number of processes are being automated through machine learning (ML) algorithms as illustrated in e.g. Goodfellow et al. (2016) or Murphy (2023). It is therefore essential that these algorithms are robust and reliable so that operations based on their output can be trusted. State-of-the-art ML algorithms perform extraordinarily well on standard data. However, they have been repeatedly shown to be vulnerable to, e.g., adversarial examples (Goodfellow et al., 2014) purposefully designed to fool ML algorithms.

The pioneering work on adversarial classification (AC) by Dalvi et al. (2004) presented the confrontation between a learning system and adversaries through the lens of game theory (Menache & Ozdaglar, 2011) largely framing the field of adversarial machine learning (AML), see reviews in Biggio and Roli (2018), Insua et al. (2023), and Zhou et al. (2018). This entails well-known common knowledge assumptions, critically analyzed in e.g. Hargreaves-Heap and Varoufakis (1995). From a fundamental point of view, such assumptions are not sustainable in security settings where adversaries tend to conceal information. In recent work (Gallego et al., 2024; Naveiro et al., 2019), we presented novel frameworks for AC based on Adversarial Risk Analysis (ARA) (Roponen et al., 2020). ARA is a decision-theoretic approach to games (Hausken, 2024) providing one-sided prescriptive support to an expected utility maximizing agent making operational the Bayesian approach to games in Kadane and Larkey (1982) and Raiffa (1982), by facilitating procedures to predict adversarial decisions.

The AML literature has predominantly focused on the supervised learning setup. Our focus will be in reinforcement learning (RL) (Powell, 2022; Sutton & Barto, 2018). In it, an agent takes actions sequentially to maximize an expected cumulative reward, learning from interactions with the environment (Kaelbling et al., 1996). With the advent of deep learning (Gallego & Ŕos Insua, 2022), deep RL has experienced an incredible growth (Mnih et al., 2015; Silver et al., 2017) with important applications in domains such as automated driving systems or autonomous robots, as well as core operational research problems such as inventory control (Boute et al., 2022). However, RL systems may be also the target of adversarial attacks (Huang et al., 2017; Lin et al., 2017) and robust learning methods are thus in need. A related field of interest is multi-agent RL (MARL) (Lee & Lee, 2021), where multiple agents try to learn to compete or cooperate. Importantly, single-agent RL methods tend to fail in these settings since they do not take into account the non-stationarity due to the actions and learning of the other agents present in the RL scenario, as Busoni et al. (2008, 2010) argue.

Based on the note by Gallego et al. (2019), our major contribution here is to demonstrate how ARA facilitates dealing with secure RL, developing an approach to model and predict adversarial actions aimed at interfering with the reward generating process. Unlike earlier MARL approaches, which typically focus on specific aspects of learning, such as worst-case scenarios or stationary opponent models, we present a global framework and provide extensive empirical evidence of its efficiency, flexibility, and robustness. We deliver strategies to predict opponents' policies, depending on the information and computational

* Corresponding author at: Institute of Mathematical Sciences (ICMAT), Madrid, Spain.

E-mail address: victor.gallego@komorebi.ai (V. Gallego).

resources available: an approach inspired by fictitious play, a scheme based on level- k thinking, and a model averaging procedure combining several adversarial models. Importantly, we extend the approach to deep RL scenarios. A variety of security experiments illustrate the generality of our proposal, covering issues like the advantages of incorporating accurate adversarial prediction models, the robustness in the presence of misspecified models, and how to handle multiple opponents.

To facilitate reproducibility, all the code and experimental setup details are released at <https://github.com/vicgalle/ARAMARL>.

2. Background and related work

Our focus is on RL, a well-studied computational approach addressing *Markov decision processes* (MDPs) (Howard, 1960): a single agent (the decision maker, or DM) makes decisions while interacting with an environment over discrete time steps. An MDP is represented by the tuple $(S, \mathcal{A}, \mathcal{T}, R)$, described as follows.¹ At each time step, the agent makes decisions a_t from a feasible set $\mathcal{A}$ within an environment S whose states are s . Such decision leads to a new random state s_{t+1} characterized by a function $\mathcal{T} : S \times \mathcal{A} \rightarrow \Delta(S)$ modeling the transition distribution associated with a state s and action a . Besides, it also provides the agent with a random $R : S \times \mathcal{A} \rightarrow \Delta(\mathbb{R})$, where $R(s_t, a_t)$ is the distribution over rewards perceived from state s_t and action a_t , with realizations $r(s_t, a_t)$. The DM chooses her actions based on a policy $\pi : S \rightarrow \Delta(\mathcal{A})$, indicating for each state s the distribution from which an action a is drawn. Her aim is to maximize her long-term discounted expected reward, that is, $\mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$, where $\gamma \in (0, 1)$ is a discount factor and $\tau = (s_0, a_0, s_1, a_1, \dots)$ is a trajectory of states and actions.

Our discussion will mainly center around Q -learning (Sutton & Barto, 2018) an efficient approach in numerous RL settings, but see our final conclusions for extensions. In it, the DM maintains a function $Q : S \times \mathcal{A} \rightarrow \mathbb{R}$ that estimates her expected cumulative reward, updated according to

$$Q(s, a) := (1 - \alpha) Q(s, a) + \alpha \left(r(s, a) + \gamma \max_{a'} Q(s', a') \right), \quad (1)$$

where α is a learning rate hyperparameter and s' designates the state after having chosen action a from state s . We focus on Q -learning for four reasons: (i) its off-policy nature lets the DM bootstrap from observed transitions, naturally absorbing the opponent's actions and rewards; (ii) the max in (1) admits a clean extension by averaging the bootstrapped value over the DM's beliefs about the opponent's move, the technical core of this paper expressed through rule (2)–(3); (iii) that rule preserves Bellman contractive structure, so its convergence analysis Appendix A carries over with minor modifications; and, (iv) the machinery is agnostic to tabular vs. parametric/deep representations of Q (Section 3.4). Stochastic policy-based methods such as policy gradients (Baxter & Bartlett, 2000), trusted region policy optimization (TRPO) (Schulman et al., 2015), proximal policy optimization (PPO) (Schulman et al., 2017) and actor-critic (Konda & Tsitsiklis, 1999) are equally applicable, and may be preferable when the equilibrium is mixed (e.g., matching-pennies-like settings); the policy-gradient surrogate is then taken in expectation over $p_A(b | s)$, and the level- k and type-based schemes of Sections 3.2–3.3 can wrap any base learner that maintains an estimate of p_A . We elaborate this in Sections 4.5 and 5.

Our major interest is in security settings where another agent interferes with the DM's reward process for their own benefit. As an example, in an air combat scenario, the rewards of a supported pilot are affected by the performance of an enemy pilot. This renders the environment non-stationary, in particular making Q -learning suboptimal

(Busoniu et al., 2010). Thus, to support the agent, we must be able to reason about and forecast the adversary's behavior. Since our concern is on the ability of our agent to predict her adversary's actions, we focus on methods with this goal, encompassing three of the broad opponent modeling principles proposed in the literature, see e.g. by Albrecht and Stone (2018). First, *policy reconstruction* methods aim to fully reconstruct the adversary's decision making problem, typically assuming a parametric model fitted after observing the adversary's behavior. A representative approach stems from fictitious play (FP, Brown, 1951) modeling the other agents by computing their frequencies of choosing various actions. Next, *type-based reasoning* assumes that the modeled agent belongs to one of several fully specified types and learns a distribution over such types, without explicitly including the ability of other agents to reason about their opponents' decision making. Finally, *recursive learning* methods stem from explicit representations of the other agents' beliefs about their opponents leading to an infinite hierarchy of decision making problems, illustrated by Rios and Insua (2012) in a simpler class of problems.

These opponent modeling principles have been developed mostly within the decision analysis arena (Banks et al., 2022), and their integration with Q -learning in multi-agent RL settings remains, to our knowledge, largely unexplored. To articulate the gap that our work fills, we organize the related literature into four threads.

Multi-agent Q-learning via Markov games. Extensions of Q -learning to multi-agent settings have focused on modeling the whole system through Markov games. Three well-known solutions are *minimax-Q learning* (Littman, 1994), where at each iteration a minimax problem is solved; *Nash-Q learning* (Hu & Wellman, 2003), generalizing the previous algorithm to non-zero sum scenarios; and *friend-or-foe-Q learning* (Littman, 2001), in which the DM knows in advance whether her opponent is an adversary or a collaborator. Within the bandit literature, Auer et al. (1995) introduced a non-stationary setting where the reward process is affected by an adversary. All of these methods inherit standard game-theoretic common-knowledge assumptions (Hargreaves-Heap & Varoufakis, 1995), unrealistic in the security domains of interest to us. Our approach departs fundamentally from this work in that we consider the problem from a single DM's perspective and explicitly model the opponent for which we propose several strategies leading to important operational advantages.

Opponent modeling in deep RL. A second thread incorporates explicit opponent models into deep RL. Our work relates to Lanctot et al. (2017), who propose a deep cognitive hierarchy as an approximation to a response oracle to obtain policies that can exploit adversaries. Ours rather draws on the level- k thinking tradition (Stahl & Wilson, 1994), building opponent models that support the DM in predicting their behavior. In addition, we provide a solution to the important issue of choosing between different opponent cognition levels. Closely related is neural fictitious self-play (Heinrich & Silver, 2016) and the broader self-play family that Lanctot et al. (2017) encompass. These methods extend FP to deep RL by approximating best responses to the empirical average policy of past play, with the explicit goal of converging to a Nash equilibrium of the underlying game. Our framework departs from this equilibrium-seeking stance: TMDPs provide Bayesian prescriptive support to a single DM, treating the opponents' behavior as sources of uncertainty whose policy is predicted from its posterior rather than averaged into a self-play target. This makes the framework adaptive under non-stationarity and model misspecification, since the DM continually updates her beliefs about the opponent, instead of relying on the convergence of a joint best-response dynamic. He et al. (2016) also addressed opponent modeling in deep RL scenarios. However, the authors rely on using a specific neural network (NN) architecture over the deep Q -network model. Instead, we tackle the problem without assuming a NN architecture for the agent policies, though our proposed scheme can be adapted to that setting, as Section 3.4 shows. Foerster et al. (2018) adopt a similar experimental setting in their LOLA framework, but

¹ In its definition, $\Delta(X)$ is used to designate the set of all probability distributions over X , endowed with an appropriate probabilistic structure.

their methods apply only when both players have exact knowledge of their opponents' policy parameters or a maximum-likelihood estimator, still a form of common knowledge. Our work instead builds upon estimating the opponents' Q -function, not requiring direct knowledge of the opponent's parameters. Wen et al. (2019) use level- k thinking in deep RL, yet they only go up to level 2 and their framework does not extend easily to higher levels, unlike ours which is generic. Even at the same cognitive level, their framework is more expensive computationally, as they rely on variational inference to estimate several latent variables. More recent developments cluster around two core ideas. *Opponent-shaping* methods directly succeeding LOLA steer the opponent's learning process: POLA (Zhao et al., 2022) makes LOLA updates proximal but still requires the opponent's policy parameters; M-FOS (Lu et al., 2022) instead meta-learns a shaping policy over many opponent lifetimes; LOQA (Aghajohari et al., 2024) shapes opponents modeled as softmax Q -learners; and advantage alignment (Duque et al., 2025) unifies this family. *Opponent-modeling* methods include MBOM (Yu et al., 2022), a deep counterpart of the level- k hierarchy with type-based averaging developed below, LIAM (Papoudakis et al., 2021), and GSCU (Fu et al., 2022). Section 4.5 discusses their informational assumptions and we compare our approach against LOQA, POLA, and MBOM in Section 4.4.

Adversary-aware planning frameworks beyond RL. A third thread tackles adversarial decision making outside the standard RL setting. These include links with iPOMDPs as in Gmytrasiewicz and Doshi (2005) and, more broadly, with Partially Observable Stochastic Games (POSGs) (Hansen et al., 2004). These authors refer only to planning, not addressing the typical RL scenario where the DM has to learn as well from the environment. Moreover, both formalisms are notoriously intractable: POSG planning is NEXP-complete, whereas i-POMDPs require maintaining beliefs over an, in principle, infinite hierarchy of nested beliefs, only computationally feasible after aggressive truncation. Our level- k bounded-rationality scheme can be viewed as a deliberate, finitely truncated counterpart of the i-POMDP nested-belief regress, designed to preserve computations linear in k , while still supporting Bayesian updating about the opponent. In addition, we introduce a more simplified algorithmic apparatus (in terms of both implementation and complexity) that performs well in examples and covers the case of mixtures of opponent types. Caballero, Lunday et al. (2021) ground their approach on level- k thinking to compute optimal strategies in normal form games, relying on solving a stochastic optimization program exactly. As our framework also tackles MDPs, their approach would be infeasible in this case due to the large number of states. Therefore, we resort to our upgraded Q -learning scheme.

Robustness and adversarial attacks on RL. A fourth thread targets the robustness of RL agents to specific attack vectors rather than strategic opponent modeling. Pinto et al. (2017) propose a method for deep RL based on policy gradients which works only for zero-sum games. In addition, their experiments focus on single-agent RL settings not accounting for an adversary. Banihashem et al. (2021) focus on poisoning attacks over the reward process of an offline learning agent, another paradigm to deal with RL. Caballero, Jenkins et al. (2021) also deal with poisoning attacks, but their work is restricted to MDPs with known transition distributions, there being no learning of the environment nor the adversaries. There is also a fair amount of work focusing on adversarial perturbations of the (visual) inputs of the learning agent, without strategically modeling the adversary who could also be learning, as in our framework. Some references are Behzadan and Munir (2017), Chen et al. (2019), Tretschk et al. (2018), and Zhang et al. (2020).

In summary, across these four threads, the literature either assumes common knowledge among agents, restricts opponent reasoning to specific NN parameterizations or fixed cognitive depths, addresses planning rather than learning, or targets robustness against specific attack

vectors instead of strategic opponent modeling. The gap we fill is the integration of the three opponent modeling principles mentioned earlier (policy reconstruction, type-based reasoning, and recursive learning) into (mainly) Q -learning, through the single-DM perspective advocated by ARA (Insua et al., 2009).

To this end, consider the problem of prescribing decisions to a single agent versus her opponents, augmenting the MDP to account for potential adversaries conveniently re-defining the Q -learning rule. This enables us to apply some of the previously mentioned modeling techniques to the Q -learning setting, explicitly accounting for the possible lack of information about the modeled opponent. Our focus will be on the case of a DM (agent A , she) facing a single opponent (B , he), although Section 3.5 provides an extension to a setting with multiple adversaries illustrated in Section 4.2.

3. Threatened Markov decision processes

Our concept of *Threatened Markov Decision Process* (TMDP) addresses the presence of an adversary who modifies state and reward dynamics resulting in a non-stationary environment. This MDP augmentation considers modified transition and reward processes and incorporates the agent's beliefs about the adversary's actions. We thus adopt a Bayesian approach to games, in the Kadane and Larkey (1982) and Raiffa (1982) sense, and making it operational in multi-stage contexts.

Definition 1. A TMDP is a tuple $(S, \mathcal{A}, \mathcal{B}, \mathcal{T}, R, p_A)$ in which S is the space of states s ; $\mathcal{A}$ is the set of actions a available to the supported agent A ; $\mathcal{B}$ is the set of threat actions b available to the adversary B ; $\mathcal{T} : S \times \mathcal{A} \times \mathcal{B} \rightarrow \Delta(S)$ is the (possibly unknown) transition distribution; $R : S \times \mathcal{A} \times \mathcal{B} \rightarrow \Delta(\mathbb{R})$ is the distribution over the reward that agent A perceives from a given state s when the pair of actions (a, b) is implemented; and $p_A(b|s)$ represents the distribution modeling A 's beliefs about B 's moves, when the current state is $s \in S$.

To ground Definition 1 in a realistic setting, consider beyond-visual-range (BVR) air-combat decision support, in the spirit of Camacho et al. (2024) and Virtanen et al. (2006). The supported agent A is a blue pilot. Her adversary B , a red one. The state s aggregates the kinematic variables of both aircraft (position, velocity, heading, pitch and roll), the remaining missile inventories, and indicators such as whether a missile is currently in flight or the target is within radar. The DM's action set $\mathcal{A}$ specifies whether to launch a missile and which waypoint (heading) to pursue, while the adversary's action set $\mathcal{B}$ comprises canonical tactical maneuvers like *search*, *launch-and-support*, *pump*, *recommit*, or *withdraw*. The transition $\mathcal{T}$ is induced by a flight-dynamics simulator together with a stochastic weapon-engagement server. Reward R combines a missile-launch term peaked at the optimal firing range, weapon-engagement-zone shaping, and a penalty for entering B 's minimum-abort range. The forecast $p_A(b|s)$ is then naturally a posterior over the adversary's latent risk profile, updated from the post-engagement observability of his maneuvering that is routine in after-action reports and tactical replays. The machinery developed below would let the blue pilot anticipate that her opponent may himself be reasoning about her launches and waypoints. This is precisely the prescriptive lift TMDPs provide over an opponent-unaware controller.

Note 1. This definition of a TMDP is relative to the supported agent A . In particular, notation p_A remarks that A 's beliefs about B 's actions are being modeled. When supporting the other agent, we would have to model her corresponding TMDP with her rewards and beliefs. Similarly, the concept and notation extends when there is more than one adversary, as Sections 3.5 and 4.2 illustrate. Δ

Note 2. Common conventions in the RL and stochastic optimization communities recommend including all necessary information in the system's state variables to compute the policy, transitions, or cumulative

rewards (see, e.g., Powell, 2019, 2022). In our case, this would mean incorporating the opponent's actions into the state. However, while the literature in these fields typically assumes a stationary environment, our setting is different: other agents are also learning, causing their policies, and, consequently, A 's estimate of $p_A(b|s)$, to evolve over time, making the environment non-stationary. In TMDPs, the non-stationary component (A 's beliefs about B 's actions) is treated separately from the stationary component. This simplifies the notation and allows us to explicitly describe how non-stationary elements are handled. Δ

TMDPs will be viewed as a relevant extension to MDPs with discounted rewards. To deal with TMDPs, we modify the standard Q -learning update rule (1) by forecasting B 's actions and averaging over them. This way the DM anticipates potential threats within her decision making process, enhancing the robustness of her decision-making policy. Formally, replace the update rule (1) by

$$Q(s, a, b) := (1 - \alpha)Q(s, a, b) + \alpha \left(r(s, a, b) + \gamma \max_{a'} \mathbb{E}_{p_A(b|s')} [Q(s', a', b)] \right), \quad (2)$$

where s' is the state reached after the DM and her adversary adopt, respectively, actions a and b at state s . Then, compute the expectation of $Q(s, a, b)$ taking into account the uncertainty about the opponent's action

$$Q(s, a) := \mathbb{E}_{p_A(b|s)} [Q(s, a, b)]. \quad (3)$$

Appendix A proves the convergence of (2)–(3).

This result only applies when the adversary's policy is stationary, i.e., its distribution does not change over time. However, as mentioned, in realistic scenarios the adversary may also learn from our actions, so that his policy will change over time. To deal with this, the next subsections will propose several approaches to model adversarial behavior. Furthermore, in most of our experiments we use $Q(s, a)$ to compute an ϵ -greedy policy for the DM when the system is at state s : with probability $(1 - \epsilon)$ choose $a^* = \arg \max_a [Q(s, a)]$; with probability ϵ , choose a random action uniformly. Yet other sampling methods such as softmax can be straightforwardly used as Section 4.1.1 illustrates.

Since common knowledge is not imposed, the core difficulty lies in modeling agent A 's uncertainty regarding adversary B 's policy through $p_A(b|s)$. For this, we make the standard multi-agent RL assumption that both agents observe their opponent's actions and rewards after they have committed to them. Based on it, three approaches are proposed to forecast the opponent's policy $p_A(b|s)$, depending on the information and computational resources available. In the first one, the adversary is assumed to be non-strategic, acting without awareness of the DM, and an FP inspired forecast is presented. Secondly, more sophisticated adversaries are modeled with a scheme based on level- k thinking. Finally, a mixture method combining different opponent models is outlined, allowing us to deal with mixed behavior. Fig. 1 presents a vignette with these components. We complete the discussion suggesting how to deal with multiple opponents, an extension to deep RL scenarios, and present computational complexity issues.

3.1. Non-strategic (level-0) opponents

Stateless setting. To fix ideas, consider first a stateless setting with discrete action spaces $\mathcal{A}$ and $\mathcal{B}$. The Q -function (2) is displayed as a table $Q(a_i, b_j)$, with $a_i \in \mathcal{A}$ the action chosen by the DM and $b_j \in \mathcal{B}$ the action chosen by the adversary. Then, the DM computes the expected reward of action a_i using the stateless version of (3), $\psi(a_i) = \mathbb{E}_{p_A(b)} [Q(a_i, b)] = \sum_{b_j \in \mathcal{B}} Q(a_i, b_j) p_A(b_j)$, where $p_A(b_j)$ reflects A 's beliefs about her opponent's actions. Finally, she chooses the action $a_i \in \mathcal{A}$ maximizing $\psi(a_i)$.

Our first option models the adversary's decisions inspired by FP. We estimate the required probabilities with the empirical frequencies of the opponent past plays with $Q(a_i, b_j)$ updated according to the

stateless version of (2). We refer to it as FPQ-learning,² re-framing this in a Bayesian manner to favor its convergence should we have prior information about the adversary's decision making. Let $p_j = p_A(b_j)$ be the probability with which the opponent chooses action b_j . If we adopt a Dirichlet prior $(p_1, \dots, p_n) \sim \mathcal{D}(\alpha_1, \dots, \alpha_n)$, where n is the number of actions available to the opponent, the posterior will be $\mathcal{D}(\alpha_1 + h_1, \dots, \alpha_n + h_n)$, with h_i being the count for action b_i , $i = 1, \dots, n$. If we denote the posterior density as $f(p|h)$, the DM would choose action a_i maximizing her expected reward, adopting the form

$$\begin{aligned} \psi(a_i) &= \int_{p \in \Delta^{n-1}} \left[\sum_{b_j \in \mathcal{B}} Q(a_i, b_j) p_j \right] f(p|h) dp \\ &= \sum_{b_j \in \mathcal{B}} Q(a_i, b_j) \mathbb{E}_{p|h} [p_j] \propto \sum_{b_j \in \mathcal{B}} Q(a_i, b_j) (\alpha_j + h_j). \end{aligned}$$

where Δ^{n-1} denotes the standard $(n - 1)$ simplex.

Including states. Generalizing this to account for states is conceptually straightforward. The Q -function adopts the form $Q(s, a_i, b_j)$. The DM needs to assess the probabilities $p_A(b_j|s)$, being natural to expect that her opponent behaves differently depending on the state. However, as S may be huge, even continuous, keeping track of $p_A(b_j|s)$ incurs in potentially prohibitive memory costs. Bayes rule $p_A(b_j|s) \propto p(s|b_j)p(b_j)$ mitigates this problem, since the supported DM will choose her action at state s by maximizing

$$\psi_s(a_i) = \sum_{b_j \in \mathcal{B}} Q(s, a_i, b_j) p_A(b_j|s) \propto \sum_{b_j \in \mathcal{B}} Q(s, a_i, b_j) p_A(s|b_j) p(b_j).$$

Inspired by Tang et al. (2017), we propose keeping track of n bloom filters, one per $p(s|b_j)$ distribution, to tractably predict the opponent's intentions in the TMDP setting, with each filter maintaining a count of the number of times that the agent visits each state. This is transparently integrated within the Bayesian paradigm, as we only need to store an additional array with the Dirichlet prior parameters α_i , $i = 1, \dots, n$, for the $p(b_j)$ part.

As a final remark, we could straightforwardly extend the above scheme if we assume that the opponent has memory of the previous stage actions. However, to mitigate memory requirements we use the concept of Markov chain mixtures (Raftery, 1985). Thus, we avoid the exponential growth in the number of required parameters and control model complexity linearly. For example, in case the opponent belief model is $p_A(b_i|a_{t-1}, b_{t-1}, s_t)$, so that the adversary recalls the previous actions a_{t-1} and b_{t-1} , we would factor it through a mixture

$$p_A(b_i|a_{t-1}, b_{t-1}, s_t) = w_1 p_A(b_i|a_{t-1}) + w_2 p_A(b_i|b_{t-1}) + w_3 p_A(b_i|s_t),$$

with $\sum_i w_i = 1$, $w_i \geq 0$, $i = 1, 2, 3$.

3.2. Level- k opponents

When the opponent is more strategic, he may model our supported DM as a level-0 thinker, thus making him a level-1 thinker in the Stahl and Wilson (1994) sense. This chain can go up as needed, so we will have to deal with modeling the opponent as a level- k thinker, with k bounded by the DM's computational or cognitive resources. For this, a hierarchy of TMDPs is introduced, with $TMDP_i^k$ referring to the TMDP that agent i , the DM or the adversary, needs to optimize, while considering its rival as a level- $(k - 1)$ thinker. Therefore,

- If the supported DM is a level-1 thinker, she optimizes $TMDP_A^1$, modeling B as a level-0 thinker (e.g., as in Section 3.1).

² Observe that only the DM uses such procedure to model her opponent. In the standard FP algorithm (Brown, 1951) all players are assumed to adopt such scheme which, under certain conditions, converges to a Nash equilibria; see, e.g., Menache and Ozdaglar (2011).

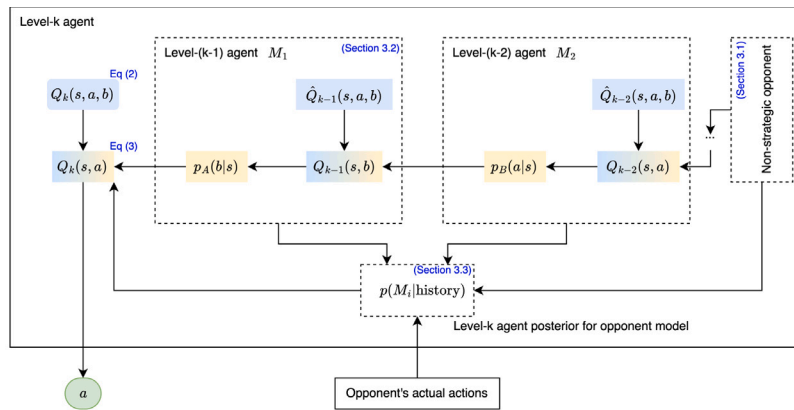


Fig. 1. Framework diagram. A level- k agent internally models her opponent as a level- $(k - 1)$ agent until arriving at a non-strategic opponent. The level- k agent has to estimate the Q -value tables (blue), and extract the corresponding policy for each of the adversarial models (yellow). Optionally, the level- k agent may place a prior over the opponent model distribution and update it. All Q -value tables are updated within each environment update, as the agent perceives her opponent(s) decisions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1 Level-2 thinking update rule

Require: $Q_2, \hat{Q}_1, \alpha_2, \alpha_1$ (DM and opponent Q -functions, learning rates).

Observe transition (s, a, b, r_A, r_B, s')

$$\hat{Q}_1(s, b, a) := (1 - \alpha_1)\hat{Q}_1(s, b, a) + \alpha_1(r_B + \gamma \max_{b'} \mathbb{E}_{p_B(a'|s')} [\hat{Q}_1(s', b', a')])$$

Compute B 's estimated ϵ -greedy policy $p_A(b|s')$ from $\hat{Q}_1(s, b, a)$

$$Q_2(s, a, b) := (1 - \alpha_2)Q_2(s, a, b) + \alpha_2(r_A + \gamma \max_{a'} \mathbb{E}_{p_A(b'|s')} [Q_2(s', a', b')])$$

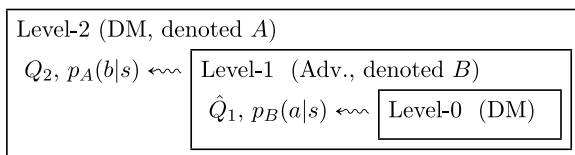


Fig. 2. Level- k thinking scheme, with $k = 2$.

- If she is a level-2 thinker, the DM optimizes $TMDP_A^2$ modeling B as a level-1 thinker: this “modeled” B optimizes $TMDP_B^1$, who, to do so, models the DM as level-0.
- In general, we shall have a chain of TMDPs

$$TMDP_A^k \rightarrow TMDP_B^{k-1} \rightarrow TMDP_A^{k-2} \rightarrow \dots$$

Exploiting the fact that TMDPs correspond to repeated interaction settings (and, by assumption, both agents observe all past decisions and rewards), each agent may estimate their counterpart’s Q -function: if the DM is optimizing $TMDP_A^k$, she will maintain her own Q -function for this, call it Q_k , as well as an estimate $\hat{Q}_{k-1}$ of her opponent’s Q -function. This estimate may be computed optimizing $TMDP_B^{k-1}$, and so on, until $k = 1$. The top level DM’s policy is

$$\arg \max_{a_{i_k}} Q_k(s, a_{i_k}, b_{j_{k-1}}),$$

where $b_{j_{k-1}}$ is given by $\arg \max_{b_{j_{k-1}}} \hat{Q}_{k-1}(s, a_{i_{k-2}}, b_{j_{k-1}})$, and so on, until the induction basis (level-1) is reached in which the opponent may be modeled using the FPQ-learning approach from Section 3.1. Fig. 2 provides a schematic view of the case in which $k = 2$, with the need to account for Q_2 , the DM’s Q -function, and $\hat{Q}_1$, that of her opponent (who will be level-1).

Algorithm 1 specifies the approach for a level-2 DM in detail, with immediate extension to higher level DMs.

Note that, in the previous hierarchy, decisions are obtained in a greedy manner, by maximizing the lower level $\hat{Q}$ estimate. We may gain insight in a Bayesian fashion by adding uncertainty to the policy at

each level. For instance, when considering ϵ -greedy policies, we could impose distributions $p_k(\epsilon)$ at each level k of the hierarchy, with the mean of $p_k(\epsilon)$ being an increasing function with respect to the level k to account for the fact that uncertainty is higher at the higher thinking levels.

3.3. Combining opponent models

In most situations the DM will not actually know which type of particular opponent she is actually facing. To deal with this, she may place a prior $p(M_i)$ denoting her beliefs that her opponent is using a model M_i , for $i = 1, \dots, m$, the range of models that might describe her adversary’s behavior, with $\sum_{i=0}^m p(M_i) = 1$ and $p(M_i) > 0$, $i = 1, \dots, m$. As an example, she might place a Dirichlet prior on the levels of the k -level hierarchy. Then, after having observed her opponent’s action at each iteration, she may update her belief $p(M_i)$ by increasing the count n_i of model M_i which caused that action, as in the standard Dirichlet-Categorical Bayesian update rule, see Algorithm 2. This is possible since the DM maintains an estimate $p_{M_i}(b|s)$ of the opponent’s policy for each opponent model M_i . Should none of these have predicted the observed b_t , then we either do not perform an update, as in Algorithm 2 and done in our experiments, or increase the count for all models.

This model averaging scheme subsumes the cognitive hierarchies framework (Camerer et al., 2004), although the distribution is placed over the different hierarchy levels. Our scheme is more flexible, though, since further kinds of opponents could be taken into account, say a minimax agent as in Insua et al. (2016).

3.4. Approximate Q-learning with function approximation

Tabular Q -learning (Algorithm 1) does not scale with the state or action spaces sizes. To address this issue, we extend the framework to settings in which Q -functions are represented using a function estimator, typically a linear regression model or a deep Q -network (Mnih et al., 2015). Algorithm 3 reflects the corresponding details, where ϕ_A and ϕ_B designate the parameters of the corresponding functions approximating the Q -values. It resembles Algorithm 1, but uses gradient

Algorithm 2 Opponent average updating

Require: $p(M|H) \propto (n_1, n_2, \dots, n_m)$ (counts for each model), with H the sequence $(b_0, b_1, \dots, b_{t-1})$ of past opponent actions.

At iteration t , observe transition $(s_t, a_t, b_t, r_{A,t}, r_{B,t}, s_{t+1})$.

For each opponent model M_i , set b^i to be the predicted action by model M_i .

If $b^i = b_t$, then update posterior

$$p(M|(H||b_t)) \propto (n_1, \dots, n_i + 1, \dots, n_m)$$

Algorithm 3 Level-2 thinking update rule using function estimators.

Require: $Q_{\phi_A}, \hat{Q}_{\phi_B}, \alpha_1, \alpha_2$ (DM's Q -function, estimate of opponent's Q -function, learning rates).

Observe transition (s, a, b, r_A, r_B, s') .

$$\phi_B := \phi_B - \alpha_1 \frac{\partial Q_{\phi_B}}{\partial \phi_B}(s, b, a) [\hat{Q}_{\phi_B}(s, b, a) - (r_B + \gamma \max_{b'} \mathbb{E}_{p_B(a'|s')} \hat{Q}_{\phi_B}(s', b', a'))]$$

Compute B 's estimated ϵ -greedy (or softmax) policy $p_A(b|s')$ from $\hat{Q}_{\phi_B}(s, b, a)$.

$$\phi_A := \phi_A - \alpha_2 \frac{\partial Q_{\phi_A}}{\partial \phi_A}(s, a, b) [Q_{\phi_A}(s, a, b) - (r_A + \gamma \max_{a'} \mathbb{E}_{p_A(b'|s')} Q_{\phi_A}(s', a', b'))]$$

descent for updating purposes, as the parameters of the Q -networks are now continuous. Observe that this framework would be compatible with continuous action spaces.

3.5. Facing multiple opponents

As announced, TMDPs may be extended to cases in which the DM faces more than one adversary. If so, the DM would have uncertainty about all of her opponents and she would need to average her Q -function over their likely actions. Let $p_A(b^1, \dots, b^M | s)$ represent the DM's beliefs about her M adversaries' actions. The extension of the TMDP framework to multiple adversaries will require to account for all possible opponents in the DM's Q function which adopts the form $Q(s, a, b^1, \dots, b^M)$. Finally, the DM would average this Q over $b^1, \dots, b^M$ in (2), proceeding as in (3).

When the DM faces non-strategic opponents, she could learn $p_A(b^1, \dots, b^M | s)$ in a Bayesian way, as in Section 3.1. This would entail placing a Dirichlet prior on the n^M dimensional vector of joint actions of all adversaries. However, keeping track of those probabilities may be unfeasible as the dimension scales exponentially with the number of opponents. When A believes her adversaries act conditionally independent given the state, the analysis is simpler, as, in this case, $p_A(b^1, \dots, b^M | s) = p_A(b^1 | s) \dots p_A(b^M | s)$. Then, we could learn each $p_A(b^i | s)$ for $i = 1, \dots, M$ separately as above, then combining the forecasts multiplicatively.

3.6. Computational complexity

Let us finish with a brief discussion on computational complexity drawing on Section 3.2 approach: a level- k Q -learner has to estimate the Q function of a level- $(k-1)$ Q -learner, which has to learn the Q function of a level- $(k-2)$ Q learner, and so on. Assuming that the original Q -learning update rule has time complexity $\mathcal{O}(T(|\mathcal{A}|))$, with T a factor that depends on the number of DM's actions, Algorithm 1 update rule has time complexity $\mathcal{O}(kT(\max\{|\mathcal{A}|, |\mathcal{B}|\}))$, i.e., it is linear in the level of the hierarchy.

Regarding space complexity, the overhead is also linear in such level k since the DM only needs to store k Q -functions, leading to a memory complexity $\mathcal{O}(kM(|\mathcal{S}|, |\mathcal{A}| \cdot |\mathcal{B}|))$, with $M(|\mathcal{S}|, |\mathcal{A}| \cdot |\mathcal{B}|)$ accounting for the memory needed to store the Q -function in tabular form. This quantity depends on the number of states and pairs of actions for the DM and her opponent.

Appendix B provides further computational verification. Timings in Table B.3 corroborate the linear-in- k bound: per-experiment run-time grows roughly linearly across levels 0–3 in the games in Section 4.1.1, Level2Q incurs only a $\sim 1.6\times$ overhead over IndQ in the Blotto experiment, and RegressionLevel2Q is about twice as costly as RegressionIndQ in the gridworld of Section 4.3. The multi-opponent extension scales provided the conditional-independence factorization of Section 3.5 is exploited.

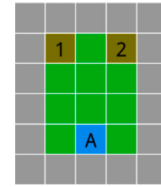


Fig. 3. Friend or foe environment from AI safety gridworlds (Leike et al., 2017). Blue cell represents DM's initial state, gray cells represent room walls. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Experiments

Let us illustrate the main modeling and computational concepts of the proposed TMDP reasoning framework through four sets of experiments. First, an adversarial security environment proposed by Leike et al. (2017) used to illustrate robustness issues. Next, a Blotto game security resource allocation problem to show how to handle multiple opponents. Third, a gridworld game to showcase the compatibility with Q -values estimated with parametric functions, as in deep RL. Finally, computational comparisons with other approaches are highlighted.

4.1. Advantages of modeling adversaries

Let us demonstrate first that by explicitly modeling the behavior of adversaries, our framework improve upon standard Q -learning methods. For this, we consider a suite of RL safety benchmarks in Leike et al. (2017), focusing on the safety friend or foe environment: the supported DM needs to travel through a room and choose between two identical boxes, respectively hiding a positive (+50) and a negative (−50) reward, under control by an adaptive opponent. Fig. 3 shows the initial set up. Cells 1 and 2 depict the adversary's targets, who decides which one will hide the positive reward.

This may be viewed as a spatial Stackelberg game in which the adversary plans to attack one of two targets. The defender obtains a positive reward if she travels to the chosen target. Otherwise, she misses the attacker and incurs in a loss. As Leike et al. (2017) show, a deep Q -network fails to achieve optimal results since the reward process is controlled by the adversary.

4.1.1. Stateless variant

Consider first a simplified setting with a single state and two actions. As in Leike et al. (2017), an adaptive opponent estimates the DM's actions using an exponential smoother. Let $p = (p_1, p_2)$ be the probabilities with which the DM respectively chooses targets 1 or 2 as estimated by B . Initial estimates are $p = (0.5, 0.5)$; after each iteration, the update is $p := \beta p + (1 - \beta)a$, with $0 < \beta < 1$ a learning rate unknown

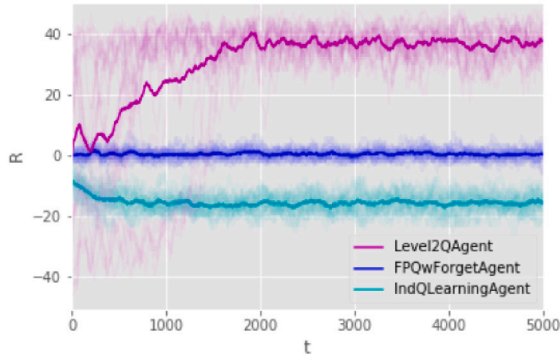


Fig. 4. Rewards for DM against adversarial opponent. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

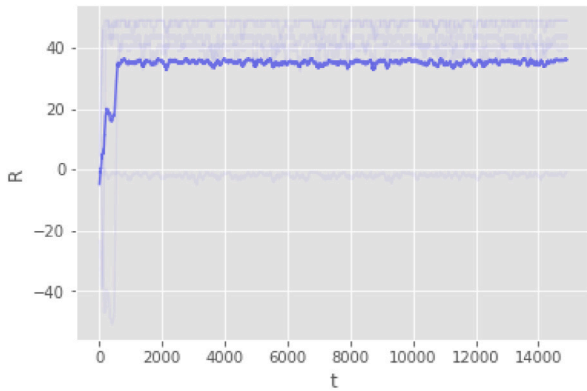


Fig. 5. Rewards for DM against adversarial opponent, using softmax policy.

by the DM, and $a \in \{(1, 0), (0, 1)\}$ is a one-hot encoded vector indicating whether the DM chose target 1 or 2. Assume an adversary which places the positive reward in target $t = \arg \min_i (p)_i$.

Since the DM has to deal with a strategic adversary, consider a variant of FPQ-learning (Section 3.1) giving additional relevance to more recent actions. Algorithm 4 provides a modified update using the fact that the Dirichlet distribution is conjugate of the Categorical one. Instead of weighting all observations equally, we essentially account for just the last $\frac{1}{1-\lambda}$ opponent actions, where $0 < \lambda < 1$ is viewed as a forget factor.

Algorithm 4 Dirichlet updating with forget factor

```

Initialize pseudocounts  $\alpha^0 = (\alpha_1^0, \dots, \alpha_n^0)$ 
for  $t = 1, \dots, T$  do
   $\alpha^t = \lambda \alpha^{t-1}$  ▷ Reweight with factor  $0 < \lambda < 1$ 
  Observe opponent action  $b_i^t, i \in \{b_1, \dots, b_n\}$ 
   $\alpha_i^t = \alpha_i^{t-1} + 1$  ▷ Update posterior
   $\alpha_{-i}^t = \alpha_{-i}^{t-1}$ 
end for

```

For a level-2 DM, as we do not know the actual adversary rewards (a level-1 learner), we model it as in a zero-sum scenario ($r_B = -r_A$), making this case similar to the Matching Pennies game. The discount factor is $\gamma = 0.8$. There are 5000 episodes. The initial exploration parameter is $\epsilon = 0.1$; the learning rate, $\alpha = 0.1$. The forget factor is $\lambda = 0.8$. Fig. 4 displays results for three defenders: an opponent unaware Q -learner (green); a level-1 DM with forget factor using Algorithm 4 (blue); and, finally, a level-2 agent (purple). The first

defender is clearly exploited by the adversary achieving suboptimal rewards (close to -20). In contrast, the level-1 DM with forget factor learns a stationary optimal policy (reward 0). Finally, the level-2 agent exploits the adaptive adversary achieving positive rewards (around 40, close to the upper bound of 50 due to the value of the positive reward).

Note that the actual adversary behaves differently from how the DM models him, as he is not a level-1 Q -learner. Even so, modeling him as such gives the DM sufficient advantage in this case. Such robustness against opponent misspecification emerges as an advantage of our framework.

Next we replace the ϵ -greedy policy by a softmax one: actions at state s are taken with probabilities proportional to the $Q(s, a)$ values. Fig. 5 provides several simulation runs of a level-2 Q -learner versus the adversary: changing the policy sampling scheme does not worsen the DM with respect to the ϵ -greedy alternative. Lastly, we checked with other models for the opponent's rewards r_B . Instead of the initial zero-sum setting, we tried two additional reward scalings $r_B \in \{-1, 1\}$ and $r_B \in \{0, 1\}$. Fig. 6 displays the results, portraying similar qualitative behavior to that in Fig. 4.

4.1.2. Facing more powerful adversaries

We study now cases when the DM faces adversaries more powerful than exponential smoothers.

First, let us parameterize our opponent as a level-2 Q -learner. For simplicity we specify the rewards he receives as $r_B = -r_A$ as in a zero-sum case, although the framework allows for general-sum problems. Fig. 7(a) depicts the rewards for both the DM (blue) and her adversary (red). We have computed the frequency of choosing each of the actions, and both players select either action with probability 0.5 ± 0.002 based on 10 different random seeds, thus achieving a Nash equilibrium (choosing between both actions with equal probability), leading to an expected cumulative reward of 0.

Increasing the DM sophistication to make her level-3 allows her to exploit a level-2 adversary, Fig. 7(b). However, this DM fails to exploit a level-1 opponent, Fig. 7(c). The explanation to this apparent paradox is that the DM is modeling her opponent as a more powerful agent than he actually is: this inaccuracy leads to poorer performance. However, this “failure” suggests a potential solution to the problem using type-based reasoning (Section 3.3). Fig. 7(d) depicts the rewards of a DM that keeps track of both level-1 and level-2 opponent models and learns, in a Bayesian manner, which one she is actually facing. The DM keeps estimates of the probabilities P_{L1Q} and P_{L2Q} that her opponent is acting as if he was a level-1 or a level-2 Q -learner, respectively. Fig. 7(e) tracks the evolution of P_{L1Q} , which places most of the probability in the correct opponent type.

4.1.3. Spatial variant

Compare now an independent Q -learner and a level-2 Q -learner against the adaptive exponential smoother opponent in the spatial gridworld domain from Fig. 3. The DM has four actions, one for each direction in which she is allowed to move one step. Target rewards are delayed until the DM arrives at one of the pertinent locations, obtaining ± 50 . Each step is penalized with -1 for the DM. Episodes end at a maximum of 50 steps or when the agent arrives first at targets 1 or 2. The discount factor is $\gamma = 0.8$ and 15 000 episodes are considered. For the level-2 agent, we consider different exploration hyperparameters, ϵ_A and ϵ_B , for the DM's policy and her corresponding model for her opponent; initially, $\epsilon_A = \epsilon_B = 0.99$ with decaying rules $\epsilon_A := 0.995\epsilon_A$ and $\epsilon_B := 0.9\epsilon_B$ every 10 episodes, and learning rates $\alpha_1 = \alpha_2 = 0.05$. For the independent Q -learner, we set similar initial hyperparameters.

Fig. 8(a) displays results. Again, an independent Q -learner (green) is exploited by the adversary, obtaining even more negative results than in Fig. 4 due to the penalty at each step. In contrast, a level-2 agent (purple) approximately estimates adversarial behavior, modeling him as a level-1 agent, obtaining positive rewards. Fig. 8(b) depicts the rewards of a DM that maintains opponent models for both level-1 and

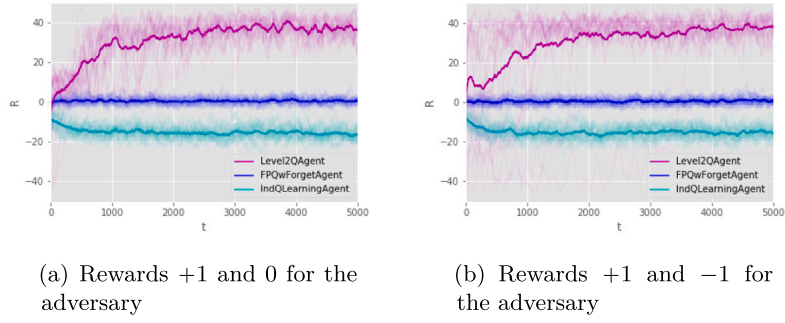


Fig. 6. Rewards against same adversary (exp. smoother), using different reward scaling.

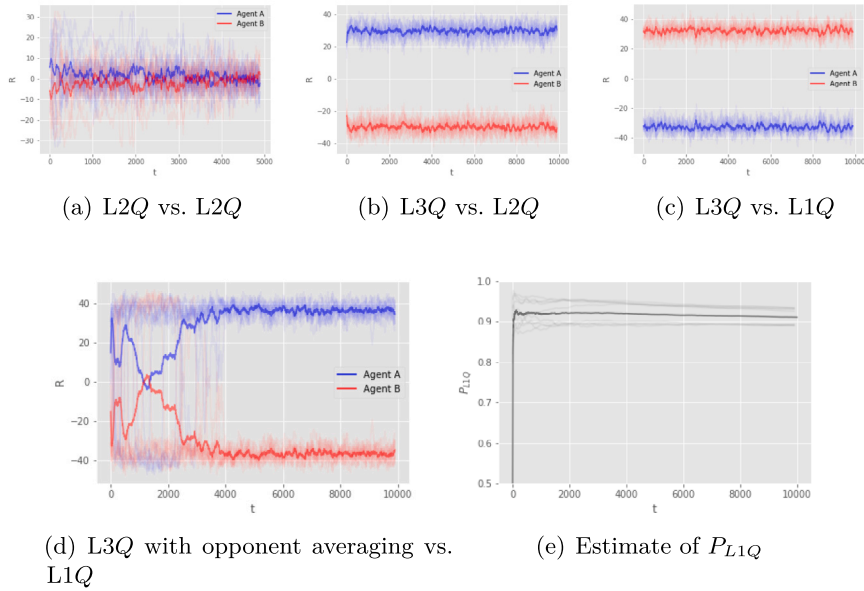


Fig. 7. Rewards (a–d) and posterior belief (e) against an exponential-smoother adversary. (a–c): DM is blue, opponent red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

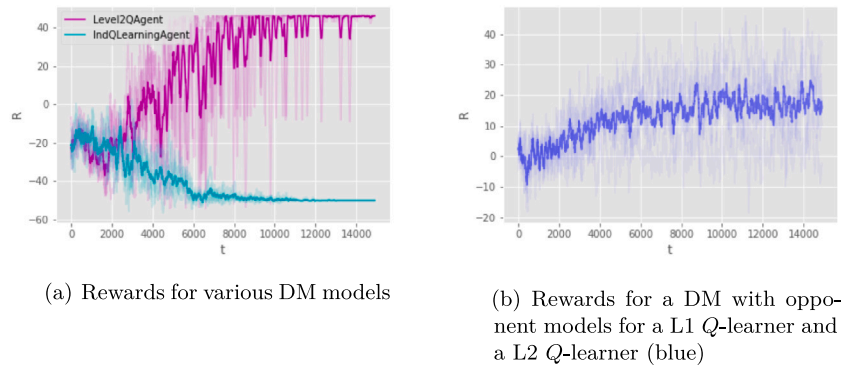


Fig. 8. Rewards against exponential smoother opponent in the spatial environment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

level-2 Q -learners. Although the adversary is of neither class, the DM achieves positive rewards, suggesting that the framework is capable of generalizing between opponent behaviors not exactly reflected by the DM's opponent model. This suggests that our level- k thinking scheme is sufficiently robust to opponent misspecification.

Lastly, we perform experiments with different hyperparameter values, to further highlight the framework robustness. Table 1 displays mean rewards and standard deviations for five different random seeds, over different hyperparameter choices within Algorithm 1. Except when the initial exploration rate ϵ_0 is set to a high value (0.5, which

Table 1

Hyperparameter robustness of Algorithm 1 in the spatial gridworld.

α_1	α_2	ϵ_0	Mean reward
0.005	0.01	0.5	15.46 $\pm$ 47.21
0.005	0.01	0.1	40.77 $\pm$ 27.48
0.005	0.01	0.01	46.32 $\pm$ 16.15
0.02	0.01	0.5	15.58 $\pm$ 47.17
0.02	0.01	0.1	43.05 $\pm$ 23.65
0.02	0.01	0.01	47.81 $\pm$ 10.83
0.05	0.1	0.5	15.30 $\pm$ 47.27
0.05	0.1	0.1	42.82 $\pm$ 24.08
0.05	0.1	0.01	48.34 $\pm$ 8.10
0.2	0.1	0.5	15.97 $\pm$ 47.03
0.2	0.1	0.1	43.05 $\pm$ 23.66
0.2	0.1	0.01	48.51 $\pm$ 6.96
0.25	0.5	0.5	15.95 $\pm$ 47.04
0.25	0.5	0.1	43.06 $\pm$ 23.64
0.25	0.5	0.01	48.41 $\pm$ 7.68
1.0	0.5	0.5	15.19 $\pm$ 47.31
1.0	0.5	0.1	42.98 $\pm$ 23.71
1.0	0.5	0.01	48.53 $\pm$ 6.82

makes the DM achieve a positive mean reward), the other settings showcase that the framework (for the level-2 case) is robust to different learning rates.

4.2. Facing multiple opponents

Let us illustrate the multiple opponent concepts from Section 3.5 introducing a novel suite of security resource allocation experiments. They are based on a modified Blotto game (Hart, 2008), a security resource-allocation setting also addressed, e.g., via counterfactual regret minimization in Keith and Ahner (2021). A DM needs to distribute limited resources over several positions susceptible of being attacked. Each of the attackers chooses different positions to deploy their attacks. Associated with each of the attacked positions there is a positive (resp. negative) reward with value 1 (−1). If the DM deploys more resources than the attackers in a given position, she wins the positive reward and the negative is equally split between the attackers choosing to attack such position. If the DM deploys less resources, she receives a negative reward and the positive one will be equally split between the corresponding attackers. In case of draw, no player receives any reward.

We compare the performance of an FPQ-learner versus a standard Q-learner, when facing two conditionally independent opponents, both using exponential smoothing to estimate the probability of the DM placing a resource at each position, and implementing the attack where this probability is the smallest (obviously, both opponents perform exactly the same attacks).

Consider defending three different positions. The DM needs to allocate two resources among such positions. For both the Q- and the FPQ-learner, the parameters will be $\gamma = 0.96$, $\epsilon = 0.1$, and $\alpha = 0.1$. As Fig. 9 shows, FPQ-learning (blue) is able to learn the opponents strategies and thus is less exploitable than standard Q-learning (red). This experiment suggests the suitability of our proposal to deal with multiple, independent adversaries, by extending the level- k thinking scheme as in Section 3.5.

4.3. Experiments with function approximation

We now run experiments to showcase that the framework is compatible with variants in which Q-values are approximated with a parametric function, in particular a deep learning model, recall Section 3.4. The game in Fig. 10 is played over an $N \times N$ grid ($N = 5$ in the Figure). Each turn, both players (blue and red) move in either of the four directions (unless they try to cross the boundary of the grid), with the objective of

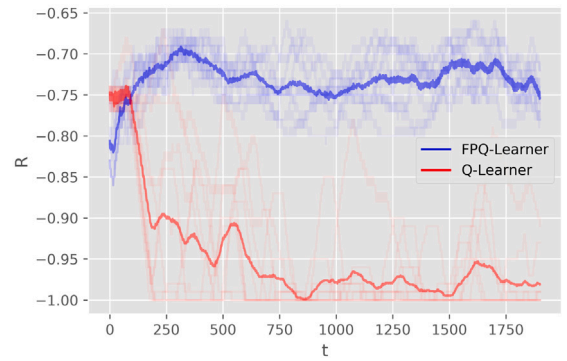


Fig. 9. Rewards for the DM against the opponents. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

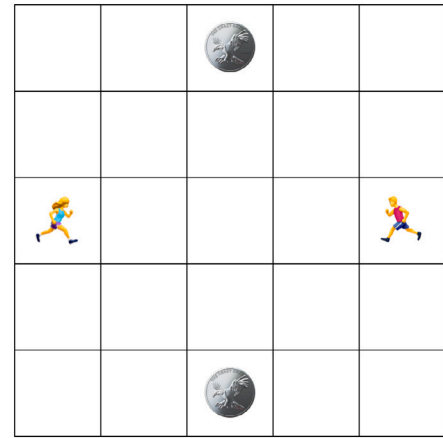


Fig. 10. An initial state for the Coin Game. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

capturing two different coins. The episode ends after 12 steps, so that each agent has ample time to pick both coins. For each coin an agent picks, he receives a reward of 1, unless the other agent arrives also at the same location, in which case the perceived reward is 0.5. Thus, this game can be cast as a coordination game.

The space of states is modeled as an $N \times N \times 4$ array. Each of its four last slices denotes the position of both players and both coins as a one-hot encoded $N \times N$ matrix. A straightforward application of Q-learning would require a $Q(s, a)$ table with $(N \times N)^4 \times 4$ entries. Instead, we parameterize the Q-values to reduce the number of parameters by flattening the state s from a tensor in $\mathbb{R}^{N \times N \times 4}$ to a vector in $\mathbb{R}^{4N^2}$, and projecting it into $\mathbb{R}^4$ using linear regression to obtain one Q-value for each of the four possible actions. Given any state $s \in \mathbb{R}^{4N^2}$, we obtain the corresponding Q-value via $Q(s, a) = w_a^T s$, with $w_a \in \mathbb{R}^{4N^2}$ being the weight vector for action a . Thus, we reduce the number of parameters to $N^2 \times 4 \times 4$. The extension to the Q-function estimator for the level-2 case is straightforward. Rather than projecting to $\mathbb{R}^4$, we do it to $\mathbb{R}^{4 \times 4}$ to consider each pair of actions (a, b) , to approximate the corresponding $Q(s, a, b)$ function.

Figs. 11(a) and 11(b) depict the results of two games. Light color depicts the smoothed rewards over 10 000 iterations with five random seeds. Darker colors reflect the three averages of the 3×5 previous runs. Even in this approximate regime, a DM exhibiting higher rationality than its adversary can behave advantageously. Thus, our opponent modeling framework is compatible with approximate Q-values, being capable of harnessing all the benefits from the approximate regime,

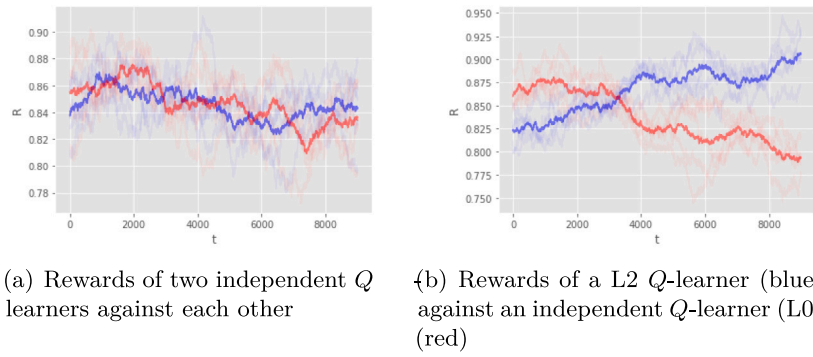


Fig. 11. Rewards in the Coin Game using function estimators. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2

Payoff bimatrix of the Chicken game used in this subsection. Rows correspond to DM's actions and columns to the opponent's. Each cell reports the (DM, opponent) rewards. Action *C* denotes *cooperate*; action *D*, *defect*.

	C	D
C	(0, 0)	(−2, 1)
D	(1, −2)	(−4, −4)

such as a reduced parameter dimensionality or enhanced generalization capabilities.

Finally, we also used a multi-layer NN to parameterize the Q -functions, with no significant improvement in the results. Therefore, we just report those attained with the linear model.

4.4. Comparisons against canonical and recent multi-agent methods

Sections 4.1–4.3 benchmarked our scheme against an independent (opponent-unaware) Q -learner, which is the natural ablation of our method: switching off the opponent-modeling component recovers exactly a standard Q -learner, so the comparison isolates the contribution of opponent modeling. To complement this ablation, we report head-to-head experiments in an iterated Chicken (IC) game, whose payoff bimatrix is recalled in Table 2: the actions *C* (cooperate) and *D* (defect) yield two pure Nash equilibria, (*C*, *D*) and (*D*, *C*), with mutual defection (*D*, *D*) giving the worst outcome (−4, −4). We face the DM against two external opponents: WoLF-PHC (Bowling & Veloso, 2001), a canonical multi-agent learner, and LOQA (Aghajohari et al., 2024), a state-of-the-art representative of the *opponent-shaping* family that has succeeded LOLA.

As a level-1 thinker (FPQ-learner) the DM is exploited by WoLF-PHC, who steers the joint dynamics toward its preferred (*C*, *D*) (Fig. 12(a)). Modeling the DM as level-2 reverses the outcome: she anticipates the opponent's adaptation and converges to her preferred (*D*, *C*), dominating WoLF-PHC (Fig. 12(b)).

LOQA, in turn, models its opponent as a Q -learner that adopts a softmax policy over its action values, estimates those values from the opponent's observed actions and rewards, and ascends the gradient of its expected return *through* the opponent's induced policy, thereby shaping the opponent's learning to its own benefit. Unlike the other recent methods later discussed in Section 4.5, its informational requirements coincide exactly with those of TMDPs, enabling a clean comparison: we employ a tabular instantiation with Monte Carlo estimators are replaced by exact expectations.³ Under the same protocol, LOQA does exactly what it was designed for: against level-0 and level-1 DMs it

commits to defection, steering our DM's Q -values until she concedes to its preferred (*C*, *D*) (Fig. 12(c)). A level-2 DM cannot be shaped: she too expects the opponent to concede, but LOQA never does, and both shapers lock into mutual defection (Fig. 12(d)), an instance of a shaper-versus-shaper conflict (Lu et al., 2022). Finally, the type-based DM with opponent averaging (Section 3.3) identifies a committed defector and concedes, securing the maximal −2.0 (Fig. 12(e)). POLA (Zhao et al., 2022), the proximal successor to LOLA run as an adversary estimating the DM's policy from play, yields the same pattern (trajectories in the companion code), so the advantage extends to the modern shaping family as a whole. Taken together, no fixed cognitive level dominates: the level-2 DM prevails over the adaptive WoLF-PHC yet clashes with the shaping LOQA, and the practical advantage lies in Bayesian, type-based selection among opponent models.

The comparisons above place the recent methods in the adversary's seat. We complement them by benchmarking, in the DM's seat, against MBOM (Yu et al., 2022), the recent method closest to our design: it builds a hierarchy of imagined opponent policies by recursive best response and mixes them through Bayesian similarity to observed behavior, a deep counterpart of the level- k recursion and type-based averaging of Sections 3.2 and 3.3. As this recursive imagination is informative only in competitive games (in Chicken it collapses onto a pure equilibrium), we adopt MBOM's own protocol: iterated Matching Pennies (the DM scores +1 when its action matches the opponent's, −1 otherwise; equilibrium value 0) against the three opponent classes it considers, fixed, naive, and reasoning learner. Our DM is the type-based scheme of Section 3.3, spanning the same model range as MBOM's imagined policies (a non-strategic branch tracking action frequencies and a level-1 branch), selected online with a Bayesian posterior over types. Fig. 13 shows it matches MBOM against the fixed policy (+0.45), slightly exceeds it against the naive learner (+0.42 vs. +0.37), and dominates against the reasoning learner (+0.86 vs. +0.14), while an opponent-unaware IndQ exploits none. MBOM attains uniform but modest adaptation; the type-based DM instead recovers the best response against every class of opponents.

4.5. Discussion

We close the experimental section distilling the main experimental insights and practical implications. Empirical evidence supporting TMDPs has been provided via extensive experiments with encouraging results. In security settings, we see that by explicitly modeling a finite

³ The adversary maintains running-mean estimates $\hat{r}_A(a, b)$, $\hat{r}_B(a, b)$ of both reward functions; models the DM as playing softmax over $\hat{Q}_A(a) =$

$\sum_b \pi_\theta(b) \hat{r}_A(a, b)$, exact up to an action-independent shift for a stateless Q -learner at its fixed point; and ascends the exact gradient of $V_B(\theta) = \sum_a \hat{\pi}_A(a; \theta) \sum_b \pi_\theta(b) \hat{r}_B(a, b)$, which includes LOQA's opponent-shaping path through $\hat{\pi}_A$. Results are robust to its learning rate, softmax temperature, and exploration rate.

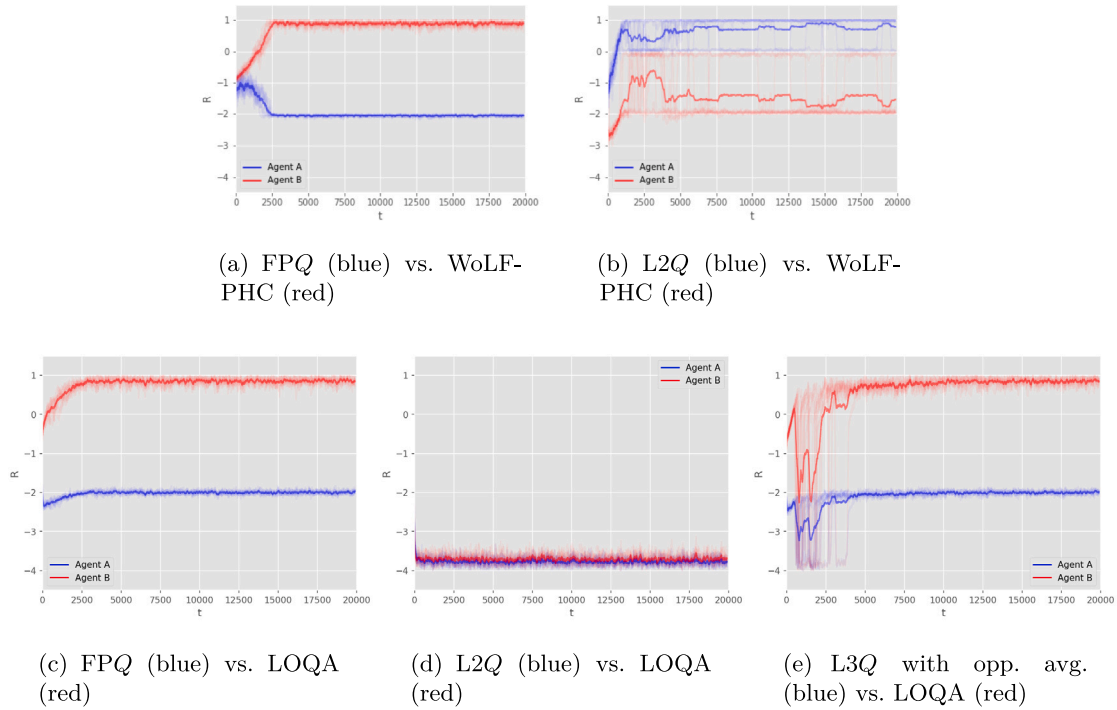


Fig. 12. Rewards in the iterated Chicken game against WoLF-PHC (top row) and LOQA (bottom row) adversaries. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

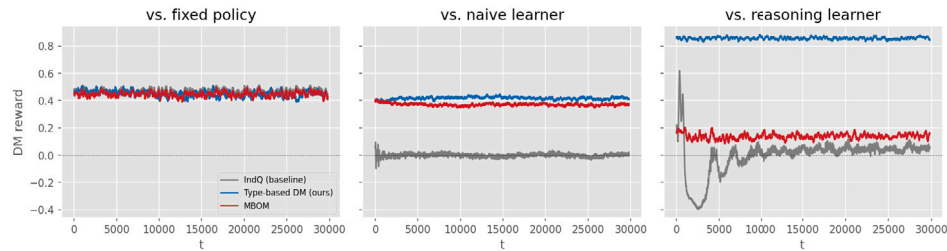


Fig. 13. Reward of supported DM (equilibrium value 0; higher is better) in iterated Matching Pennies against the three opponent classes in Yu et al. (2022). Our type-based DM (blue) matches or exceeds MBOM (red) throughout and dominates against the reasoning learner, while opponent-unaware IndQ (gray) exploits none. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

set of adversaries via the opponent averaging scheme, a supported DM can take advantage of her actual opponent, even when he is not explicitly modeled through one of the components in the finite mixture. This highlights the ability of our framework to generalize between different kinds of opponents. Interestingly, we find that a level-2 Q -learner may effectively deal with a wide class of adversaries. However, maintaining a mixture of different adversaries is necessary if we consider a level-3 DM. As a general heuristic rule, the supported DM may start at a low level in the hierarchy, and switch to a level-up temporarily, to check if the obtained rewards are higher. Otherwise, she would continue at the initial, lower level.

Two practical properties of the proposed scheme are also delivered through the experiments. First, it is robust, as it offers protection against adversarial behaviors that are not exactly as described by the DM's opponent model. Second, these benefits come at a reasonable cost: the increase in time and space complexity is linear compared to unprotected, baseline Q -learners.

Our experiments have compared against seven complementary baselines: (i) an independent (opponent-unaware) tabular Q -learner, which is the natural ablation, since our scheme is a drop-in modification of Q -learning; (ii) the DQN behavior reported by Leike et al. (2017) on

the Friend-or-Foe environment; (iii) adaptive exponential-smoother adversaries of varying sophistication; (iv) within-framework comparisons across cognitive levels, a non-trivial baseline since recursive-reasoning schemes such as Wen et al. (2019) only operate at level-2; (v) the WoLF-PHC (Bowling & Veloso, 2001) comparison in Section 4.4; (vi) the comparisons against LOQA (Aghajohari et al., 2024) and POLA (Zhao et al., 2022), two state-of-the-art opponent-shaping methods, in the same section; and (vii) MBOM (Yu et al., 2022), a recent model-based recursive-reasoning method, benchmarked as an alternative decision-maker there. We do not benchmark against the Markov-game family (minimax- Q Littman, 1994, Nash- Q Hu & Wellman, 2003, friend-or-foe- Q Littman, 2001) because they require common knowledge of payoffs, zero-sum, or known friend/foe status; LOLA (Foerster et al., 2018) requires access to the opponent's policy parameters or gradients; DPIQN/DRON (He et al., 2016) are tied to a specific NN architecture; PSRO (Lanctot et al., 2017) relies on a meta-game oracle; and Pinto et al. (2017) address only zero-sum perturbations. Analogous obstacles affect another recent generation of methods: M-FOS (Lu et al., 2022) dispenses with opponent gradients but requires meta-training over many opponent lifetimes in a resettable simulator; LIAM (Papoudakis et al., 2021) and GSCU (Fu et al., 2022) depend on pretrained pools of

opponent policies. None of these assumptions hold under the limited-information setting natural to ARA, in which a single lifetime of play against an unknown opponent is available. The exceptions, all compared against in Section 4.4, are LOQA (whose requirements coincide with ours), POLA (run as a shaping adversary), and MBOM (whose environment model is available in closed form for the tabular games studied, so it serves as an alternative decision-maker). The motivation for selecting Q -learning, and the transfer to policy-based RL, was given at the end of Section 2.

A natural concern is that TMDPs assume the DM eventually observes her opponent's actions and rewards, whereas in security and cybersecurity these signals can be partially observable. Two remarks. First, in the application domains we target, post-hoc observability is the rule rather than the exception: intrusion-detection logs, forensic traces, threat-intelligence feeds, and after-action reports routinely reveal attacker actions and the damage they caused, even when unavailable in real time. Second, the framework is modular: the opponent forecast $p_A(b | s)$ can be replaced by a posterior over latent opponent actions learned from noisy or delayed observations, without altering the level- k machinery. The full-information POSG and i-POMDP formulations that drop this assumption altogether are intractable beyond toy problems (Section 2), so the stated assumption is a deliberate tractability trade-off. A full partial-observability extension and head-to-head comparison against truncated i-POMDPs constitutes promising future work.

A second natural question is how TMDPs compare to policy-based RL (PG, TRPO, PPO) when scaling to continuous or high-dimensional action spaces. As anticipated at the end of Section 2, the construction lifts to those settings essentially unchanged. Here we discuss the resulting trade-offs. Q -learning is off-policy and sample-efficient, admits a contractive Bellman operator (Appendix A), and yields the linear-in- k overheads reported in Section 3.6. These properties fit the discrete, security-flavored decisions we target. Conversely, the $\max_a$ in Q -learning is awkward in continuous action spaces and a poor fit for mixed equilibria. In those regimes, stochastic policy-based methods handle continuous controls natively and benefit from the trust-region or clipped-surrogate machinery of TRPO and PPO for stable updates. The trade-off they buy is more acute in non-stationary multi-agent settings: policy gradients have higher variance, on-policy data demands, and weaker theory, and the opponent's drift further inflates the cost of fresh interactions.

5. Conclusions and further work

We have introduced TMDPs, a reformulation of MDPs to support decision-makers who confront opponents interfering with the reward generation process in RL settings. Potential applications abound. As examples, we mention developing schemes for pilot decision support in air combat modeling, see Virtanen et al. (2006) for motivation; for autonomous driving decision support, see Naveiro et al. (2022) for a static version of an example problem in lane-changing support; or for dynamic pricing, see a discussion in Rasines et al. (2024).

TMDPs aim at providing one-sided prescriptive support to an RL agent, maximizing her expected reward taking into account predictions about potential negative actions adopted by an adversary. The learning rule (2)–(3) is a contraction mapping (Appendix A) and we may use RL convergence results, while gaining advantage from opponent modeling within Q -learning. Indeed, we have proposed a scheme to model adversarial behavior based on level- k reasoning extended with type-based reasoning to account for uncertainty about the opponent's level. Finally, we have sketched how the framework could be extended to deep learning settings and multiple opponents. Empirical insights and practical properties (generality, robustness, and complexity) were discussed in Section 4.5.

Several lines of work are possible for further research. First of all, in the experiments, we have just considered DMs up to level-3, though the extension to higher-order adversaries is relevant. In addition, a

concrete next step is the policy-gradient instantiation of the framework for continuous-action and high-dimensional settings, which would also enable a model-agnostic extension to deep multi-agent RL scenarios such as those of Mnih et al. (2015). Finally, it would be interesting to explore similar expansions to semi-MDPs, in order to perform hierarchical RL or allow for time-dependent rewards and transitions between states.

CRedit authorship contribution statement

Víctor Gallego: Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Resources, Methodology, Investigation, Conceptualization. **Roi Naveiro:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Conceptualization. **David Ríos Insua:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Funding acquisition, Conceptualization. **David Gómez-Ullate:** Writing – original draft, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This manuscript was initially prepared while the authors were visiting SAMSI within the Games and Decisions in Risk and Reliability program, under National Science Foundation (NSF) Grant DMS-1638521 (Statistical and Applied Mathematical Sciences Institute). The authors also acknowledge support from the Severo Ochoa Excellence Programme CEX-2023-001347-S, Ministry of Science programs PID2021-124662OB-I00 and PID2025-172412OB-C21, the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101021797 (STARLIGHT), the FBBVA AMALFI project, and the SEDIA Excellent AI project EMOROB CARE. VG acknowledges support from grant FPU16-05034 and PTQ2021-011758, RN acknowledges support from grant FPU15-03636. RN acknowledges the 2026 Leonardo Grant for Scientific Research and Cultural Creation from the BBVA Foundation. The BBVA Foundation accepts no responsibility for the opinions, comments, and content included in the project and/or the results derived from it, which are the sole and absolute responsibility of their authors. DRI is grateful to the AXA-ICMAT Chair in Adversarial Risk Analysis.

Appendix A. Convergence proof of update rule for TMDPs

Given a TMDP, consider an augmented state space with transitions $(s, b) \xrightarrow{a} (s', b') \xrightarrow{a'} \dots$. The DM does not observe the full state, since she does not know the action b taken by her adversary, but it suffices that she knows his policy $p(b|s)$ or a good estimate of it. Assume the opponent's action b is observed. Using the standard Q -learning recursion (Sutton & Barto, 2018) and the conditional independence $p(s', b' | s, a, b) = p(b' | s') p(s' | s, a, b)$, the optimal Q -function satisfies

$$Q^*(s, a, b) = \sum_{s'} p(s' | s, a, b) \left[R_{ss'}^{ab} + \gamma \max_{a'} \mathbb{E}_{p(b' | s')} [Q^*(s', a', b')] \right],$$

with $R_{ss'}^{ab} = \mathbb{E} [r_{t+1} | s_{t+1} = s', s_t = s, a_t = a, b_t = b]$. Observe now that

Lemma 1. Given $q : S \times B \times A \rightarrow \mathbb{R}$, the operator $\mathcal{H}$

$$(\mathcal{H}q)(s, b, a) = \sum_{s'} p(s' | s, b, a) [r(s, b, a) + \gamma \max_{a'} \mathbb{E}_{p(b' | s')} q(s', b', a')]$$

is a contraction mapping under the supremum norm.

Table B.3

Experiment times. Sections 4.1.1–4.1.2: 20 seeds, 5000 steps. Sections 4.1.3 and 4.3: 5 seeds, 10 000 steps.

Sections 4.1.1–4.1.2	Section 4.1.3	Section 4.3
IndQ	10.9 s	IndQ 3 m 24 s
FPQwForget	18.6 s	Level2Q 5 m 18 s
Level2Q	25.7 s	RegressionIndQ 4 m 23 s
Level3Q	34.5 s	RegressionLevel2Q 8 m 29 s

Proof. Let us prove that $\|Hq_1 - Hq_2\|_\infty \leq \gamma \|q_1 - q_2\|_\infty$.

$$\begin{aligned}
 \|Hq_1 - Hq_2\|_\infty &= \gamma \max_{s,b,a} \left| \sum_{s'} p(s'|s,b,a) \left[\max_{a'} \mathbb{E}_{p(b'|s')q_1}(s',b',a') \right. \right. \\
 &\quad \left. \left. - \max_{a'} \mathbb{E}_{p(b'|s')q_2}(s',b',a') \right] \right| \\
 &\leq \gamma \max_{s,b,a} \sum_{s'} p(s'|s,b,a) \max_{a',z} \left| \mathbb{E}_{p(b'|z)q_1}(z,b',a') - \mathbb{E}_{p(b'|z)q_2}(z,b',a') \right| \\
 &\leq \gamma \max_{s,b,a} \sum_{s'} p(s'|s,b,a) \max_{a',z,b'} |q_1(z,b',a') - q_2(z,b',a')| = \gamma \|q_1 - q_2\|_\infty. \quad \square
 \end{aligned}$$

Then, using the proposed learning rule (2), we would converge to the optimal Q for each of the opponent actions. The result follows directly from the standard Q -learning convergence proof, see, e.g., Melo (2001), using Lemma 1.

However, at the time of making the decision we do not know what action he would take. Thus, we average over the possible opponent actions, weighting each of the actions by $p(b|s)$, as (3) suggests.

Appendix B. Experiment details

Experiments were run on a 16-core Intel i7-5960X server with 128 GB RAM; code is at <https://github.com/vicgalle/ARAMARL>. Table B.3 reports per-episode times, whose overhead grows linearly with the DM's cognitive level, consistent with Section 3.6.

References

- Aghajohari, M., Duque, J. A., Cooijmans, T., & Courville, A. (2024). LOQA: Learning with opponent Q-learning awareness. In *12th international conference on learning representations*. <https://dx.doi.org/10.48550/arXiv.2405.01035>.
- Albrecht, S. V., & Stone, P. (2018). Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258, 66–95. <https://dx.doi.org/10.1016/j.artint.2018.01.002>.
- Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proc. 36th symp. found. c.s.* (pp. 322–331). <https://dx.doi.org/10.1109/SFCS.1995.492488>.
- Banihashem, K., Singla, A., & Radanovic, G. (2021). Defense against reward poisoning attacks in reinforcement learning. <https://dx.doi.org/10.48550/arXiv.2102.05776>, arXiv preprint arXiv:2102.05776.
- Banks, D., Gallego, V., Naveiro, R., & Ríos Insua, D. (2022). Adversarial risk analysis: An overview. *Wiley Interdisciplinary Reviews: Computational Statistics*, 14(1), Article e1530. <https://dx.doi.org/10.1002/wics.1530>.
- Baxter, J., & Bartlett, P. L. (2000). Direct gradient-based reinforcement learning. vol. 3, In *2000 IEEE int. symp. circ. sys. proc.* (pp. 271–274). IEEE, <https://dx.doi.org/10.1109/ISCAS.2000.856049>.
- Behzadan, V., & Munir, A. (2017). Vulnerability of deep reinforcement learning to policy induction attacks. In *Mach. learn. data mining in patt. recog. 13th int. conf.* Springer, https://dx.doi.org/10.1007/978-3-319-62416-7_19.
- Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84, 317–331. <https://dx.doi.org/10.1016/j.patcog.2018.07.023>.
- Boute, R. N., Gijbrecchts, J., van Jaarsveld, W., & Vanvuchelen, N. (2022). Deep reinforcement learning for inventory control: A roadmap. *European Journal of Operational Research*, 298(2), 401–412. <https://dx.doi.org/10.1016/j.ejor.2021.07.016>.
- Bowling, M., & Veloso, M. (2001). Rational and convergent learning in stochastic games. In *Proc. 17th int. joint conf. AI* (pp. 1021–1026). Morgan Kaufmann, URL: <https://dl.acm.org/doi/10.5555/1642194.1642231>.
- Brown, G. W. (1951). Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 374–376.

- Busoni, L., Babuska, R., & De Schutter, B. (2008). A comprehensive survey of multi-agent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2), 156–172. <https://dx.doi.org/10.1109/TSMCC.2007.913919>.
- Busoni, L., Babuska, R., & De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1* (pp. 183–221). Springer, https://dx.doi.org/10.1007/978-3-642-14435-6_7.
- Caballero, W. N., Jenkins, P. R., & Keith, A. J. (2021). Poisoning finite-horizon Markov decision processes at design time. *Computers & Operations Research*, 129, Article 105185. <https://dx.doi.org/10.1016/j.cor.2020.105185>.
- Caballero, W. N., Lunday, B. J., & Uber, R. P. (2021). Identifying behaviorally robust strategies for normal form games under varying forms of uncertainty. *European Journal of Operational Research*, 288(3), 971–982. <https://dx.doi.org/10.1016/j.ejor.2020.06.022>.
- Camacho, J. M., Naveiro, R., & Ríos Insua, D. (2024). Algorithmic decision analysis for multi-stage games with incomplete information. In *Int. conf. alg. dec. theory* (pp. 98–112). Springer, https://dx.doi.org/10.1007/978-3-031-73903-3_7.
- Camerer, C. F., Ho, T.-H., & Chong, J.-K. (2004). A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3), 861–898. <https://dx.doi.org/10.1162/0033553041502225>.
- Chen, T., Liu, J., Xiang, Y., Niu, W., Tong, E., & Han, Z. (2019). Adversarial attack and defense in reinforcement learning from AI security view. *Cybersecurity*, 2, 1–22. <https://dx.doi.org/10.1186/s42400-019-0027-x>.
- Dalvi, N., Domingos, P., Sanghvi, S., Verma, D., et al. (2004). Adversarial classification. In *Proc. 10th ACM SIGKDD* (pp. 99–108). ACM, <https://dx.doi.org/10.1145/1014052.1014066>.
- Duque, J. A., Aghajohari, M., Cooijmans, T., Ciucu, R., Zhang, T., Gidel, G., & Courville, A. (2025). Advantage alignment algorithms. In *13th international conference on learning representations*. <https://dx.doi.org/10.48550/arXiv.2406.14662>.
- Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., & Mordatch, I. (2018). Learning with opponent-learning awareness. In *Proc. 17th int. conf. aut. agents and multiAgent systems* (pp. 122–130). URL: <https://dl.acm.org/doi/10.5555/3237383.3237408>.
- Fu, H., Tian, Y., Yu, H., Liu, W., Wu, S., Xiong, J., Wen, Y., Li, K., Xing, J., Fu, Q., & Yang, W. (2022). Greedy when sure and conservative when uncertain about the opponents. In *Proceedings of the 39th international conference on machine learning* (pp. 6829–6848).
- Gallego, V., Naveiro, R., & Insua, D. R. (2019). Reinforcement learning under threats. vol. 33, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 9939–9940). <https://dx.doi.org/10.1609/aaai.v33i01.33019939>.
- Gallego, V., Naveiro, R., Redondo, A., Ríos Insua, D., & Ruggeri, F. (2024). Protecting classifiers from attacks. *Statistical Science*, 39(3), 449–468.
- Gallego, V., & Ríos Insua, D. (2022). Current advances in neural networks. *Annual Review of Statistics and its Application*, 9(1), 197–222. <https://dx.doi.org/10.1146/annurev-statistics-040220-112019>.
- Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24, 49–79. <https://dx.doi.org/10.1613/jair.1579>.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press, ISBN: 9780262035613.
- Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. <https://dx.doi.org/10.48550/arXiv.1412.6572>, arXiv preprint arXiv:1412.6572.
- Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *AAAI conference on artificial intelligence* (pp. 709–715). URL: <https://dl.acm.org/doi/10.5555/1597148.1597262>.
- Hargreaves-Heap, S., & Varoufakis, Y. (1995). *Game Theory: A Critical Introduction*. Routledge.
- Hart, S. (2008). Discrete colonel blotto and general lotto games. *International Journal of Game Theory*, 36(3–4), 441–460. <https://dx.doi.org/10.1007/s00182-007-0099-9>.
- Hausken, K. (2024). Fifty years of operations research in defense. *European Journal of Operational Research*, 318(2), 355–368.
- He, H., Boyd-Graber, J., Kwok, K., & Daume, H., III (2016). Opponent modeling in deep reinforcement learning. In *International conference on machine learning* (pp. 1804–1813). <https://dx.doi.org/10.48550/arXiv.1609.05559>.
- Heinrich, J., & Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. <https://dx.doi.org/10.48550/arXiv.1603.01121>, arXiv preprint arXiv:1603.01121.
- Howard, R. A. (1960). *Dynamic programming and Markov processes*. Cambridge, MA: MIT Press.
- Hu, J., & Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4(Nov), 1039–1069, URL: <https://dl.acm.org/doi/10.5555/945365.964288>.
- Huang, S., Papernot, N., Goodfellow, I., Duan, Y., & Abbeel, P. (2017). Adversarial attacks on neural network policies. <https://dx.doi.org/10.48550/arXiv.1702.02284>, arXiv preprint arXiv:1702.02284.
- Insua, D., Banks, D., & Ríos, J. (2016). Modeling opponents in adversarial risk analysis. *Risk Analysis*, 36(4), 742–755. <https://dx.doi.org/10.1111/risa.12439>.

- Insua, D. R., Naveiro, R., Gallego, V., & Poulos, J. (2023). Adversarial machine learning: Bayesian perspectives. *Journal of the American Statistical Association*, 1–12. <http://dx.doi.org/10.1080/01621459.2023.2183129>.
- Insua, D., Rios, J., & Banks, D. (2009). Adversarial risk analysis. *Journal of the American Statistical Association*, 104(486), 841–854. <http://dx.doi.org/10.1198/jasa.2009.0155>.
- Kadane, J. B., & Larkey, P. D. (1982). Subjective probability and the theory of games. *Management Science*, 28(2), 113–120. <http://dx.doi.org/10.1287/mnsc.28.2.113>.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237–285. <http://dx.doi.org/10.1613/jair.301>.
- Keith, A. J., & Ahner, D. K. (2021). Counterfactual regret minimization for integrated cyber and air defense resource allocation. *European Journal of Operational Research*, 292(1), 95–107. <http://dx.doi.org/10.1016/j.ejor.2020.10.015>.
- Konda, V., & Tsitsiklis, J. (1999). Actor-critic algorithms. 12 URL: <https://dl.acm.org/doi/10.5555/3009657.3009799>.
- Lancotot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Perolat, J., Silver, D., & Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in neural information processing systems* (pp. 4190–4203). <http://dx.doi.org/10.48550/arXiv.1711.00832>.
- Lee, H., & Lee, T. (2021). Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response. *European Journal of Operational Research*, 291(1), 296–308. <http://dx.doi.org/10.1016/j.ejor.2020.09.018>.
- Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., & Legg, S. (2017). AI safety gridworlds. <http://dx.doi.org/10.48550/arXiv.1711.09883>, arXiv preprint [arXiv:1711.09883](https://arxiv.org/abs/1711.09883).
- Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., & Sun, M. (2017). Tactics of adversarial attack on deep reinforcement learning agents. <http://dx.doi.org/10.24963/jcai.2017/525>, arXiv preprint [arXiv:1703.06748](https://arxiv.org/abs/1703.06748).
- Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994* (pp. 157–163). Elsevier, <http://dx.doi.org/10.1016/B978-1-55860-335-6.50027-1>.
- Littman, M. L. (2001). Friend-or-foe Q-learning in general-sum games. In *Proc. of the 18th ICML* (pp. 322–328). URL: <https://dl.acm.org/doi/10.5555/645530.655661>.
- Lu, C., Willi, T., Schroeder de Witt, C., & Foerster, J. (2022). Model-free opponent shaping. In *Proceedings of the 39th international conference on machine learning* (pp. 14398–14411).
- Melo, F. S. (2001). *Convergence of Q-learning: A simple proof*. Tech. rep., URL: <http://users.isr.ist.utl.pt/~mtjspsaan/readingGroup/ProofQlearning.pdf>.
- Menache, I., & Ozdaglar, A. (2011). Network games: Theory, models, and dynamics. *Synthesis Lectures on Communication Networks*, 4(1), 1–159. <http://dx.doi.org/10.2200/S00330ED1V01Y201101CNT009>.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529. <http://dx.doi.org/10.1038/nature14236>.
- Murphy, K. P. (2023). *Probabilistic machine learning: Advanced topics*. MIT Press, ISBN: 9780262048439.
- Naveiro, R., Caballero, W. N., & Ríos Insua, D. (2022). Adversarial risk analysis for heterogeneous traffic management. <http://dx.doi.org/10.2139/ssrn.4365987>, Available At SSRN 4365987.
- Naveiro, R., Redondo, A., Insua, D., & Ruggeri, F. (2019). Adversarial classification: An adversarial risk analysis approach. *International Journal of Approximate Reasoning*, <http://dx.doi.org/10.1016/j.ijar.2019.07.003>.
- Papoudakis, G., Christianos, F., & Albrecht, S. V. (2021). Agent modelling under partial observability for deep reinforcement learning. vol. 34, In *Advances in neural information processing systems*. <http://dx.doi.org/10.48550/arXiv.2006.09447>.
- Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017). Robust adversarial reinforcement learning. In *International conference on machine learning* (pp. 2817–2826). <http://dx.doi.org/10.48550/arXiv.1703.02702>.
- Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3), 795–821. <http://dx.doi.org/10.1016/j.ejor.2018.07.014>.
- Powell, W. B. (2022). *Reinforcement learning and stochastic optimization: A unified framework for stochastic optimization*. John Wiley & Sons, <http://dx.doi.org/10.1002/9781119815068>.
- Raftery, A. E. (1985). A model for high-order Markov chains. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 528–539. <http://dx.doi.org/10.1111/j.2517-6161.1985.tb01383.x>.
- Raiffa, H. (1982). *The art and science of negotiation*. Cambridge, MA: Belknap Press of Harvard University Press, ISBN: 9780674048133.
- Rasines, D. G., Naveiro, R., Insua, D. R., & Santana, S. R. (2024). Personalized pricing decisions through adversarial risk analysis. *International Transactions in Operational Research*, <http://dx.doi.org/10.1111/itor.13545>.
- Rios, J., & Insua, D. (2012). Adversarial risk analysis for counterterrorism modeling. *Risk Analysis: An International Journal*, 32(5), 894–915. <http://dx.doi.org/10.1111/j.1539-6924.2011.01713.x>.
- Roponen, J., Rios Insua, D., & Salo, A. (2020). Adversarial risk analysis under partial information. *European Journal of Operational Research*, 287(1), 306–316. <http://dx.doi.org/10.1016/j.ejor.2020.04.037>.
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In F. Bach, & D. Blei (Eds.), *Proceedings of machine learning research: vol. 37, Proceedings of the 32nd international conference on machine learning* (pp. 1889–1897). Lille, France: PMLR, <http://dx.doi.org/10.48550/arXiv.1502.05477>.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. <http://dx.doi.org/10.48550/arXiv.1707.06347>, arXiv preprint [arXiv:1707.06347](https://arxiv.org/abs/1707.06347).
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676), 354. <http://dx.doi.org/10.1038/nature24270>.
- Stahl, D. O., & Wilson, P. W. (1994). Experimental evidence on players' models of other players. *Journal of Economic Behavior and Organization*, 25(3), 309–327. [http://dx.doi.org/10.1016/0167-2681\(94\)90103-1](http://dx.doi.org/10.1016/0167-2681(94)90103-1).
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, ISBN: 9780262039246.
- Tang, H., Houthoofd, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., & Abbeel, P. (2017). # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems* (pp. 2750–2759). <http://dx.doi.org/10.48550/arXiv.1611.04717>.
- Tretschk, E., Oh, S. J., & Fritz, M. (2018). Sequential attacks on agents for long-term adversarial goals. <http://dx.doi.org/10.48550/arXiv.1805.12487>, arXiv preprint [arXiv:1805.12487](https://arxiv.org/abs/1805.12487).
- Virtanen, K., Karelaitis, J., & Raivio, T. (2006). Modeling air combat by a moving horizon influence diagram game. *Journal of Guidance, Control, and Dynamics*, 29(5), 1080–1091. <http://dx.doi.org/10.2514/1.17168>.
- Wen, Y., Yang, Y., Luo, R., Wang, J., & Pan, W. (2019). Probabilistic recursive reasoning for multi-agent reinforcement learning. vol. 7, In *7th international conference on learning representations*. <http://dx.doi.org/10.48550/arXiv.1901.09207>.
- Yu, X., Jiang, J., Zhang, W., Jiang, H., & Lu, Z. (2022). Model-based opponent modeling. vol. 35, In *Advances in neural information processing systems*. <http://dx.doi.org/10.48550/arXiv.2108.01843>.
- Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., & Hsieh, C.-J. (2020). Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33, 21024–21037. <http://dx.doi.org/10.48550/arXiv.2003.08938>.
- Zhao, S., Lu, C., Grosse, R. B., & Foerster, J. N. (2022). Proximal learning with opponent-learning awareness. vol. 35, In *Advances in neural information processing systems*. <http://dx.doi.org/10.48550/arXiv.2210.10125>.
- Zhou, Y., Kantarcioglu, M., & Xi, B. (2018). A survey of game theoretic approach for adversarial machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, <http://dx.doi.org/10.1002/widm.1259>.