SCIENCE CHINA
Information Sciences

• REVIEW •

July 2024, Vol. 67, Iss. 7, 171101:1–171101:27
https://doi.org/10.1007/s11432-023-3955-x

Learning in games: a systematic review

Rong-Jun QIN[1,2] & Yang YU[1,2*]

[1]*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023*, China;*
[2]*Polixir Technologies, Nanjing* 210023*, China*

Received 27 April 2023/Revised 20 September 2023/Accepted 1 December 2023/Published online 28 June 2024

Abstract Game theory studies the mathematical models for self-interested individuals. Nash equilibrium is arguably the most central solution in game theory. While finding the Nash equilibrium in general is known as polynomial parity arguments on directed graphs (PPAD)-complete, learning in games provides an alternative to approximate Nash equilibrium, which iteratively updates the player's strategy through interactions with other players. Rules and models have been developed for learning in games, such as fictitious play and no-regret learning. Particularly, with recent advances in online learning and deep reinforcement learning, techniques from these fields greatly boost the breakthroughs in learning in games from theory to application. As a result, we have witnessed many superhuman game AI systems. The techniques used in these systems evolve from conventional search and learning to purely reinforcement learning (RL)-style learning methods, gradually getting rid of the domain knowledge. In this article, we systematically review the above techniques, discuss the trend of basic learning rules towards a unified framework, and recap applications in large games. Finally, we discuss some future directions and make the prospect of future game AI systems. We hope this article will give some insights into designing novel approaches.

Keywords non-cooperative games, learning in games, no-regret learning, reinforcement learning, superhuman AI

1 Introduction

Games involve strategic interactions between multiple players where each rational player should pursue the maximum payoff. A game is quite different from the single-agent decision-making problem, which aims to find an optimal strategy that maximizes the agent's expected payoff in a given environment. In games, the payoff of each player heavily depends on the choices of other players, so the player may not unilaterally achieve the maximum payoff without impairing other players' interests. Games have been the playgrounds of novel artificial intelligence techniques for ages. Game theory is the study of interactions among independent, self-interested players. In this review, we mainly concentrate on non-cooperative game theory[1)], which has become the most popular branch of game theory. Game theory also studies the strategy of each individual. If the player has already known others' strategies, a game will reduce to a single-agent problem that seeks an optimal strategy, which is not the general case of games. As a result, the notion of the optimal strategy is no longer meaningful in games, and game theorists deal with this problem by identifying certain subsets of outcomes, called solution concepts.

One of the most influential solution concepts in game theory is the Nash equilibrium (NE), where nobody has incentives to deviate from his current strategy unilaterally. In this case, the joint strategy of all the players forms the NE. In the 1950s, John Nash proved that if we admit a mixed strategy, NE is guaranteed to exist in games. However, finding NE is believed not easy. It does not correspond to NP problems that are decision problems, since the answer to whether NE exists is always yes. Polynomial parity arguments on directed graphs (PPAD) is a complexity class for these problems. The computational complexity of finding NE is known as PPAD-complete [1] for multi-player general-sum games [2]. Even

* Corresponding author (email: yuy@nju.edu.cn)

1) Note that non-cooperative games do not necessarily prohibit cooperation. The term "non-cooperative" refers to the fact that the basic modeling unit in this type of game is the individual, rather than a group.

for two-player general-sum games, finding NE is PPAD-complete [3]. However, finding the second NE or NE that satisfies some property is NP-complete in general.

Instead of finding the exact equilibrium, an approximate equilibrium is often desired, which can be obtained by learning approaches that learn from repeatedly playing the game. In broad strokes, the approximation process iteratively evaluates the current strategy and updates it based on the feedback, as if the player is learning to play the game. Fictitious play (FP) [4] is a well-established learning framework where the player models the opponent's (empirical) strategy by tracking the historical plays of the opponent, and then chooses the strategy that achieves the highest payoff against the opponent's empirical strategy in each round. As a player would simulate the play in mind and update their future play based on this simulation, hence the name fictitious play. Fictitious play is one of the earliest learning rules, and it is actually proposed as an iterative method for computing NE in the 1950s. In modern literature, the terminology learning in games refers to the mathematical models to describe the behavior of players that learn to play the game, as well as the iterative solving framework. The theory of learning in games [5] formally discusses learning models to study the long-run outcome of players and their behaviors in a repeated playing game. A later work [6] raises the seminal question "If learning is the answer, what is the question?" and frames the agendas of learning in games. These studies greatly boost the formalization of learning in games.

In addition to mathematical models that describe the behaviors of learning players, a more practical question is whether and when a learning rule will result in the desired outcome. Since the goal of each player is to improve his payoff, learning in games is essentially an optimization problem. From the optimization perspective, researchers have devised numerous learning algorithms for games via no-regret/online learning [7] and variational inequality for generalized Nash equilibrium problems (GNEP) [8]. These learning algorithms are mainly tailored to stateless games, where players in each game interact once, rather than sequentially, and thus solving for the best response to an opponent is tractable. However, real-world large-scale games like Go and Poker proceed sequentially, and finding an optimal strategy against a fixed opponent can be immensely challenging, which is equivalent to solving a single-agent sequential decision-making problem with a large state-action space. Consequently, traditional reasoning and search methodologies suffer from the curse of dimensionality. Reinforcement learning (RL) [9] offers a viable solution to large-scale single-agent decision-making problems, which interacts with the environment and learns an optimal policy (strategy). Since RL perceives the environment as a black box and does not require much domain knowledge, it is a general oracle for solving the optimal policy and has also been widely employed in large-scale games.

In recent years, we have witnessed many superhuman AI systems that incorporate learning algorithms for more complex scenarios than ever, from single-agent problem [10,11] by RL to games [12–18] by search, no-regret learning, RL, and meta-games. These advances that surpassed top humans in large-scale non-cooperative games (especially two-player zero-sum games) adopt two seemingly different categories of techniques, i.e., no-regret learning and reinforcement learning with population-based self-play training. These two techniques have achieved unprecedented success in complex large-scale two-player zero-sum games (Go, no-limit Texas Hold'em, StarCraft II) consecutively. Moreover, both techniques achieved these feats earlier than the timeline humans predicted for AI to "defeat" humans. Tracing back the development of these two approaches, it can be found that no-regret learning focuses more on the theoretical aspects, and its implementation more closely aligns with the theory. On the other hand, although reinforcement learning with population-based training has broader applications (board games, card games, real-time strategy games, and multi-player online battles), there exist many heuristic designs in practice, such as win-rate-based selections of the opponent for training, or a preference for the most recent ego policy, and the final policy to deploy may not follow the theory. Thus, the theoretical understanding of these practical techniques is still missing.

Additionally, the previous focuses of these two communities are quite different. No-regret learning leaned more towards optimization techniques, proposing specific learning methods for different types of games and providing proof of convergence. On the other hand, reinforcement learning in games has two levels to study: the meta-game level and the original game level. At the level of the meta-game (also called empirical game), beating a given opponent is assumed to be "easy" and there is usually an increasing policy pool (also population), and then in order to reach an equilibrium, relevant studies include which opponent to choose from the pool, how to evaluate the current policy pool, and how to improve the effectiveness of the pool. In the original game level, the focus is on how RL algorithms should be conducted in the multi-player game setting, with related studies including RL under the game-

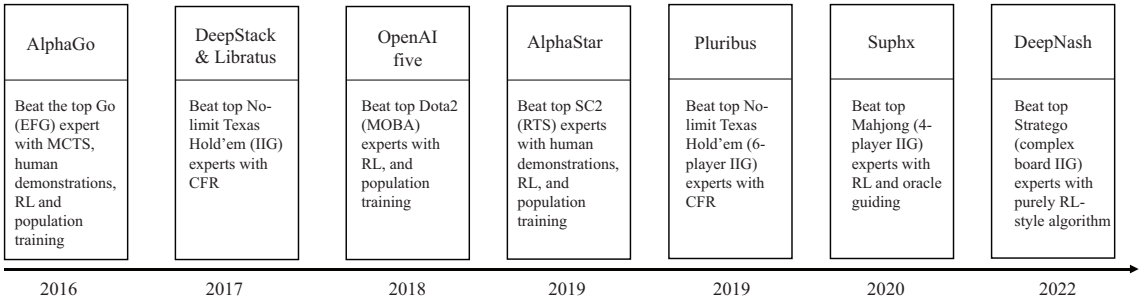

Figure 1 Milestones of learning in large-scale games that defeat top humans for the first time in chronological order.

theoretic framework, methods that focus on learning with opponent, theoretical properties, and the sample complexity of multi-agent reinforcement learning.

Figure 1 presents the milestones of learning in large-scale games in chronological order. We noticed the solving techniques for games have evolved from conventional search and learning [12] to purely RL-style learning methods [18] which originate from the no-regret perspective, gradually getting rid of the prior knowledge about the game structure and human demonstrations. Thus, these recent approaches have the potential to scale to other complex games without changing too much. No-regret learning is one of the earlier methods used for game learning, while reinforcement learning has broad applicability and is widely used in some complex, large-scale games, due to its versatility and the recent advances of deep neural networks as policy models. Although these two techniques may seem to belong to entirely different categories, many ideas and tools can be borrowed from one another, and they develop toward a unified learning objective. A successful example is DeepNash. Besides, we also add a concise overview in Table 1.

There exist many surveys concerning multi-agent learning (MAL) [19–21] mainly from the multi-agent RL (MARL) view and modern AI in games [22, 23] from the technical view; however, we notice that a systematic review of recent advances on learning in games from theory and algorithm to application is missing. We also notice that these techniques of learning in games and reinforcement learning share many commonalities and borrow tricks from each other. Learning in games has also reshaped the way of thinking in other communities. Thus, we hope this review of recent techniques can boost new algorithms for these fields.

The rest of this article is organized as follows. Section 2 gives the notations. In Section 3, we will present the development of no-regret learning in games and some theoretic results. Counterfactual regret minimization [24], the key technique in solving Poker in recent years, with its evolution is also reviewed in Section 3. Then learning in games with RL is introduced in Section 4, which covers value-based methods, policy gradient methods, RL with the game-theoretic learning framework, and opponent-aware learning. As self-play with RL and population-based training has become a very popular paradigm for many large-scale applications, population-based training methods, which originate from the empirical game, are included in Section 4. In Section 5, we compare the design philosophy of different algorithms. Then we retrospect the regret minimization and reward maximization, and discuss the unified learning framework for regret minimization and reward maximization from recent studies. In Section 6, we recap some applications in large-scale games and some platforms for learning in games. In the end, we briefly discuss possible future directions and make the prospect of a future game AI system.

2 Preliminary

2.1 Non-cooperative games

A (non-cooperative) game usually contains a finite set N of n players. Each player is indexed by i, and $-i$ denotes the players except i. The finite set of available actions for player i is A_i. $A = A_1 \times \cdots \times A_n$ is the joint action space, and each $\boldsymbol{a} = (a_1, \ldots, a_n) \in A$ is called an action profile. The utility (or payoff) function $u_i : A \mapsto \mathbb{R}$ for player i is a real-valued function. Note that u_i depends on the action profile rather than merely on player i's own action, which is different from the single-player decision-making problem. When $\sum_i u_i(\boldsymbol{a}) = 0$ or $\sum_i u_i(\boldsymbol{a}) = c$ for every $\boldsymbol{a}$, where c is s constant, then the game is zero-sum or constant-sum game. The two-player zero-sum game (2p0s) is widely studied in the field of game theory, as it reflects a fully-competitive scenario and has many nice properties. If for any $\boldsymbol{a}$ and for

Table 1 Concise overview of approaches to learning in games presented in this work

Category	Methodology	Pros and cons	Some relevant work
No-regret learning in games	No-regret learning	Intuitive with nice convergence property, while the implementation may be complicated.	FTRL [25], OFTRL [26, 27], OMWU [28]
	CFR family	The first scalable algorithm and easy to use, but tailored to extensive-form games.	CFR [24], MCCFR [29], linearCFR [30]
RL in games	Centralized training	Theoretically sound but too restrictive.	Minimax Q, Nash Q [31–33]
	RL with self-play	Generally applicable, but sometimes less efficient.	FSP [34], NFSP [35]
	Policy gradient in games	Able to combine powerful RL tools.	PG-based algorithms [36–39]
	Learning with opponent awareness	Leading to interesting outcomes but is not scalable now.	Opponent modeling [40], LOLA [41–43]
Open-ended learning	PSRO and meta-strategy solvers	Generally applicable, but sometimes less efficient; convergence is in the asymptotic sense.	PSRO variants [44–47]
	Population diversity	Important tools for finding diverse policies, but they are mainly heuristic methods.	[48–51]

any pair of players i, j, $u_i(\boldsymbol{a}) = u_j(\boldsymbol{a})$, then the game is common-payoff game. The general case, where the utility function does not have some special property, is subsumed into the general-sum game.

In game theory, the strategy of player i prescribes how the player will play. The simplest way is to choose a single action for the player, resulting in a pure strategy. When all the players adopt a pure strategy, the action profile is equivalent to the (pure) strategy profile. The mixed strategy $\sigma_i : A_i \mapsto [0, 1]$ for player i is a probability distribution over the action set. Similarly, the mixed strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is the Cartesian product of each player's mixed strategy, which can be simplified as $\sigma = (\sigma_i, \sigma_{-i})$. Then the expected utility for player i is $u_i(\sigma) = \mathbb{E}_{\boldsymbol{a}\sim\sigma}[u_i(\boldsymbol{a})] = \sum_{\boldsymbol{a}} u_i(\boldsymbol{a}) \prod_j^n \sigma_j(a_j)$. The best response of player i against a given σ_{-i} satisfies $\mathrm{BR}_i(\sigma_{-i}) = \arg\max_{\sigma_i} u(\sigma_i, \sigma_{-i})$. However, the solution concept in the game turns to equilibrium, since each player cannot unilaterally maximize his utility. A (mixed) strategy profile $\sigma = (\sigma_i, \sigma_{-i})$ is an NE if and only if

$$\forall i, \forall \sigma_i',\ u_i(\sigma_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \leqslant 0, \tag{1}$$

or equivalently for each player i, the strategy satisfies $\sigma_i \in \mathrm{BR}_i(\sigma_{-i})$. The intuition behind an NE is that nobody can gain more by unilaterally deviating from the NE. If the right-hand side of (1) is replaced by a non-negative ϵ, then σ is a ϵ-NE. Nash convergence (NashConv) is a common approach to measure the distance of a given σ to an NE, which is defined as

$$\textsc{NashConv}(\sigma) = \sum_i^n \max_{\sigma_i'} u_i(\sigma_i', \sigma_{-i}) - u_i(\sigma). \tag{2}$$

Another important metric is the exploitability of the other players' strategy σ_{-i},

$$\texttt{expl}_i(\sigma) = u_i(\mathrm{BR}_i(\sigma_{-i}), \sigma_{-i}) - u_i(\sigma). \tag{3}$$

$\texttt{expl}_i(\sigma)$ tells how much σ_{-i} will be exploited by player i when player i switches to a best response. And the exploitability[2)] of a strategy profile σ is

$$\texttt{expl}(\sigma) = \sum_i^n \texttt{expl}_i(\sigma). \tag{4}$$

In two-player zero-sum games, for the same σ, $\textsc{NashConv}(\sigma) = \texttt{expl}(\sigma)$, and both values are 0 when σ is an NE; i.e., an NE is unexploitable.

The basic assumption behind Nash equilibrium is that every player acts independently, and the probability of the joint action $\boldsymbol{a}$ is $\sigma(\boldsymbol{a}) = \prod_{i=1}^n \sigma_i(a_i)$. However, players can act jointly, and thus their

2) In some studies, the exploitability is averaged by the number of players.

strategies are correlated. Let the joint action $\boldsymbol{a}$ sample from the strategy profile σ (a coordinator), then σ is a correlated equilibrium (CE) if it satisfies

$$\forall i, a_i',\ \mathbb{E}_{\boldsymbol{a}\sim\sigma}[u_i(\boldsymbol{a})] \geqslant \mathbb{E}_{(a_i',a_{-i})\sim\sigma}[u_i(a_i', a_{-i})|a_i]. \tag{5}$$

That is, for player i, the utility will not increase if he replaces the action a_i with another action a_i' every time the coordinator recommends a_i. If the condition a_i on the right hand side (RHS) of (5) is removed, i.e.,

$$\forall i, \sigma_i',\ \mathbb{E}_{\boldsymbol{a}\sim\sigma}[u_i(\boldsymbol{a})] \geqslant \mathbb{E}_{a_{-i}\sim\sigma_{-i}}[u_i(\sigma_i', a_{-i})], \tag{6}$$

then σ is a coarse correlated equilibrium (CCE). It is easy to see that every Nash equilibrium is a correlated equilibrium by letting $\sigma(\boldsymbol{a}) = \prod_i^n \sigma_i(a_i)$.

One more thing that deserves note is that real-world games are often extensive-form games (see Section 5 in [52]) or Markov games (also known as stochastic games, see Definition 6.2.1 in [52]), while only the normal-form games are introduced for simplicity. For instance, Poker games are extensive-form games where players sequentially play to earn a higher utility in the terminal state. Real-time strategy (RTS) games like StarCraft are often Markov games, where players simultaneously make decisions in each state which usually corresponds to a normal-form game. Although these two kinds of games can be transformed into an equivalent normal-form game, the actions of the resulting game will be exponential in the number of game states rather than polynomials. Besides, players in these games aim to achieve higher long-run utility, which intertwines with RL in recent years.

2.2 Reinforcement learning

RL [9] aims to find an optimal policy that maximizes the cumulative reward in a single-agent sequential decision-making problem via the interactions with the environment. RL can be formulated as a Markov decision process (MDPs) with a 5-tuple $\langle S, A, P, r, \gamma\rangle$, where S is the state space, A is the action space, $P : S \times A \times S' \mapsto [0, 1]$ is the transition function which transits from state s to s' when the agent takes action a, $r : S \times A \mapsto \mathbb{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor that accounts for the trade-off between the instant reward and the long-term return. The (behavioral) policy $\pi : S \times A \to [0, 1]$ for the agent is a conditional probability over the action space. For a given policy π, the corresponding state value function $V(s)$ on the given s and state-action value function $Q(s, a)$ on the given pair (s, a) are defined as follows:

$$V^\pi(s) = \mathbb{E}_{\tau\sim\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)|s_0 = s\right], \tag{7}$$

$$Q^\pi(s, a) = \mathbb{E}_{\tau\sim\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)|s_0 = s, a_0 = a\right], \tag{8}$$

where $\tau = (s_0, a_0, r_0, s_1, a_1, \ldots)$ is the trajectory when the agent follows π to take actions in the environment. The goal of the RL agent is to find an optimal policy that maximizes the expected trajectory return, i.e.,

$$\pi^\star = \arg\max_\pi \sum_{s_0} d(s_0)V^\pi(s_0), \tag{9}$$

where $d(s)$ is the distribution of initial states. Note that while the trajectory horizon above is infinite, it also fits into the finite horizon with a maximum length of H. A simple approach is to learn the value function $Q(s, a)$ and then derive the final policy by $\pi(a|s) = \arg\max_a Q(s, a)$. However, during the learning process, the policy used to collect the samples is perturbed to explore the whole state-action space, e.g., by an ϵ-greedy(Q) policy which follows the uniform random policy with probability ϵ. For instance, in Q-learning [53], the Q function is updated as

$$Q(s, a) \leftarrow Q(s, a) + \alpha\big(r(s, a) + \gamma \max Q(s', \cdot) - Q(s, a)\big), \tag{10}$$

where α is the learning rate, and the agent executes the ϵ-greedy(Q) policy in the environment. Another important line of research is the policy gradient (PG) method with function approximations. Let π_θ

denote the policy parameterized by θ; then by policy gradient theorem [54], various policy gradient algorithms follow the update rule on π_θ as

$$\theta \leftarrow \theta + \alpha \cdot \mathbb{E}_{\pi_\theta}[f(s,a)\nabla_\theta \log \pi_\theta(s,a)], \tag{11}$$

where α is the learning rate and $f(s,a)$ is a score function that evaluates the current policy π_θ. Candidates of the score function can be $Q(s,a)$ or the advantage function $A(s,a) = Q(s,a) - V(s)$. Under the policy gradient theorem, subtracting a baseline (the baseline should not depend on the action) from the score function does not influence the policy gradient. Besides, this technique can greatly reduce the variance of the policy gradient when performing a sample-based policy gradient update. Later we will see that the advantage function is equivalent to the instant regret.

We notice that some terms from game theory, online learning, and reinforcement learning are the same thing, e.g., the player and agent, and the strategy[3)] and (behavioral) policy. In this following, we will not distinguish these terms and interchangeably use them when necessary.

3 No-regret learning in games

The basic idea of no-regret learning algorithms is simple and intuitive: minimizing the cumulative regret during the actual play. In no-regret learning, there are two seemingly heuristic algorithms, i.e., regret matching (RM) [55] and multiplicative weight update (MWU) algorithm [56]. The former chooses actions proportion to the current accumulated regret, while the latter iteratively increases the weight of well-performing actions and decreases the weight of poorly-performing actions. Both of these heuristic algorithms have no-regret guarantees in the worst case. Subsequently, researchers proposed a unified no-regret learning analytical framework, follow-the-regularized-leader (FTRL), which unifies these two no-regret learning algorithms and other no-regret learning algorithms. Based on FTRL, researchers turned to the convergence speed of no-regret learning algorithms beyond worst-case scenarios. The main trend is to introduce an additional optimistic term in FTRL, which is a prediction of the future utility, to achieve faster convergence. This framework is called OFTRL. In subsequent research, by adjusting the learning step size or designing more advanced learning methods, the convergence speed has been greatly improved in specific games. Additionally, theoretical research also focuses on the convergence of the last-iterate convergence under the no-regret learning framework. In fact, many algorithms under the OFTRL framework have last-iterate convergence in certain games, such as two-player zero-sum games.

In practical applications, especially in imperfect information extensive-form games (IIG) like Poker, the backbone no-regret learning algorithms usually are variants of RM. This is simply because RM has no additional parameters, and due to its simplicity in computation, it is more efficient than some higher-order optimization methods [57,58]. The milestone work for solving Poker is counterfactual regret minimization (CFR) [24], which demonstrated that minimizing the regret in each information node in IIG can achieve global regret minimization, avoiding direct regret minimization on the complete game tree. Based on CFR, subsequent studies have consistently improved the solving efficiency in each information node. Unfortunately, later methods that converge significantly faster in practice currently only have a convergence speed roughly on par with the basic CFR. The success achieved in Poker games also relies on an efficient abstraction of poker, greatly reducing the state and action space. However, these abstraction methods often require domain knowledge of Poker. Recently, there have also been efforts to combine CFR with deep neural networks, aiming to remove the dependence on domain knowledge.

In the following sections, we will provide more details from these two perspectives.

3.1 No-regret learning

The basic philosophy and two representative no-regret algorithms. Regret, as the literal meaning, measures how much worse could it get for not choosing the optimal action. We will formally introduce regret through the lens of the online decision problem. In the online decision problem, the agent makes a decision each round without knowing the future and suffers a loss $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ from the online environment, where $\mathcal{X}, \mathcal{Y}$ are the decision space for the agent and the outcome space of the environment respectively (see Subsection 2 in [7] for detailed notations). Imagine in round T the agent remembers the

3) In game theory, the strategy can be more complex than the behavioral or mixed form, which is beyond the scope of this paper.

environmental outcomes in previous $T-1$ rounds and has the chance to choose again an ideal strategy that minimizes the total loss. Since the agent cannot foresee the outcome of the environment, this ideal strategy is the best he can do. Then a fundamental question raises: whether the agent can match this ideal strategy during the online decision-making process. So the actual cumulative loss suffered compared to some fixed optimal action/strategy in hindsight, i.e., the regret of the agent, is expected to be small. In terms of games, each agent learns to play a game repeatedly, where the opponent can be fixed or also maximize his utility. In general, the agent takes an action based on the history and then receives a utility each round. Regret measures the utility (or loss, which can be simply converted to the utility by adding a minus sign) difference between a comparison policy and the actual play in retrospect. The instant regret of the action $a_i \in A_i$ for the agent i in round t is calculated by

$$R_{i,t}(a_i) = u_i(a_i, a_{-i,t}) - u_i(a_{i,t}, a_{-i,t}), \tag{12}$$

where $a_{i,t}, a_{-i,t}$ are the actual plays of the agent i and the other agents, and u_i is the utility function[4)] for the agent i. The actual play of the agent i can also be replaced by the strategy in iteration t. The main focus is the cumulative (external) regret (also regret for short) of the action a_i after T rounds for the agent i,

$$R_i^T(a_i) = \sum_{t=1}^{T} R_{i,t}(a_i). \tag{13}$$

The above cumulative regret (of action a_i) measures the utility difference that the agent i could have earned if he followed an expert, i.e., a constant action all the time. It also naturally extends to a fixed mixed strategy. Intuitively, if the regret of some action is high, the agent will regret not following that action. Thus, the online learning algorithm minimizes the regret based on history. When the regret of a learning algorithm satisfies

$$\Pr\left[\frac{1}{T}\lim_{T\to\infty}\max_{a_i\in A_i} R_i(a_i) \leqslant 0\right] = 1, \tag{14}$$

then it is said to be no-regret (or Hannan consistent [59]).

If we represent each action using one-hot encoding, the fraction of time that each pure strategy is played, in the limit, is called empirical play. This also induces an average strategy as $\frac{1}{T}\sum_{t=1}^{T} a_{i,t}$. Note that when the opponent uses a fixed strategy, the average strategy from a no-regret learning algorithm approximates a best response to the opponent. Moreover, there is a well-known close connection between no-regret learning and the solution concept in games: if for every agent, the average regret $\frac{R_i}{T}$ vanishes, then the empirical play of the players converges to a CCE in general-sum games or NE in 2p0s games.

RM [55] is a simple no-regret learning algorithm where the strategy in iteration $t+1$ for player i is given as

$$\sigma_i^{t+1} = \begin{cases} \dfrac{(R_i^t(a))^+}{\sum_{a'}(R_i^t(a'))^+}, & \displaystyle\sum_{a'}(R_i^t(a'))^+ > 0, \\ \dfrac{1}{|A|}, & \text{o.w.}, \end{cases} \tag{15}$$

where $x^+ = \max(x, 0)$. RM uses a heuristic rule where the probability of an action is in proportion to the cumulative regret if the cumulative regret is strictly greater than 0, and breaks ties uniformly at random if the regret of each action is non-positive. RM can also be derived from Blackwell's approachability [60,61]. Due to its simplicity, RM is often used as the no-regret learning backbone in many real-world complex games, e.g., Poker games. Hedge [56], which is a member of the MWU algorithms, is also a no-regret algorithm. Hedge obtains the strategy for the next round as

$$\sigma^{t+1}(a_i) \propto \sigma^t(a_i) \cdot \left(\frac{1}{\beta}\right)^{\eta u_i(a_i)}, \tag{16}$$

where $\beta \in (0,1)$ which is often set as $\beta = \frac{1}{e}$ and η is a parameter. Hedge re-weights the strategy by an exponential utility to get a weight for each action, and then normalizes the new weight to get the strategy. When $\beta = \frac{1}{e}$, Hedge is equivalent to the analytic form:

$$\sigma^{t+1} = \frac{\exp(\eta R_i^t(a_i))}{\sum_{a_i'}\exp(\eta R_i^t(a_i'))} = \frac{\sigma^t(a_i)\exp(\eta R_{i,t}(a_i))}{\sum_{a_i'}\sigma^t(a_i')\exp(\eta R_{i,t}(a_i'))} = \frac{\sigma^t(a_i)\exp(\eta u_{a_i,t})}{\sum_{a_i'}\sigma^t(a_i')\exp(\eta u_{a_i',t})}. \tag{17}$$

4) Online learning often aims at minimizing the loss, which can be converted to maximize the utility.

Algorithm 1 Regret matching

Input: T (number of iterations);
Set $R^t(a) = 0$ for each action a;
for $t = 1$ to T **do**
    Compute the strategy σ^t as (15) for each action a;
    Choose action a_t according to σ^t;
    Observe utility $u_t(a)$ for each action a;
    Update regrets: $R_i^t = R_i^t + u_{a,t} - u_{a_t,t}$ for each action a;
end for

Algorithm 2 Hedge algorithm

Input: T (number of iterations), η (learning rate);
Set $R^t(a) = 0$ for each action a;
for $t = 1$ to T **do**
    Compute the strategy σ^t as (17) for each action a;
    Choose action a_t according to σ^t;
    Observe utility $u_t(a)$ for each action a;
    Compute (instant) regrets: $R_{i,t} = u_{a,t} - u_{a_t,t}$ for each action a;
end for

Hence, Hedge can be viewed in a similar way as RM, which is exponentially in proportion to the regret. Algorithms 1 and 2 show the side-by-side comparisons. The main differences are how the strategy is derived and whether the instant regret will be accumulated. The benefit of Hedge algorithm is that it updates incrementally and only needs the access to the current strategy and utility of each action, thus avoiding storing the historical strategies, utilities, or other statistics. However, it involves an extra parameter η which is often set as $O(\sqrt{T})$ to guarantee the average regret is $O(\frac{1}{\sqrt{T}})$, while RM is parameter-free.

A unified no-regret learning framework: FTRL. In general, the no-regret learning process is decentralized[5)], where the learning agent treats the environment (with other players) as a black box. The decentralized learning paradigm is formulated as uncoupled in [62] where the player is unaware of other players' utilities. So in games, the no-regret learning agent can be unaware of other players and adapt to them. Due to the connection between the no-regret property and CCE[6)], no-regret learning algorithms can be readily applied in learning in games, e.g., regret matching is originally proposed as a solver for CCE. From the perspective of prediction with expert advice problem in online learning (see Chapter 2 in [7]), the action taken in each round can be treated as following some expert's advice. One natural choice for the agent is to follow the (best) leader (FTL), which is the expert with minimum total cost so far. It is easy to see that this learning strategy is equivalent to the fictitious play process. Let $\ell_t = \ell(x_t, y_t)$ denote the loss at round t. We mainly focus on the full-information feedback setting, where the player can observe the expected loss function $\ell(\cdot, y_t)$. When the loss function ℓ is convex in $\mathcal{X}$ and the experts are constant, i.e., $\forall x, y$ and any round i, j, $\ell_i(x, y) = \ell_j(x, y)$, then the average regret is bounded by $O(\frac{\log T}{T})$ (see Subsection 3.2 in [7]).

Unfortunately, the above naive FTL process is not no-regret for general loss function. A simple counter-example is given by the following loss sequences:

$$\ell(1, y_t) = (0, 1, 0, 1, 0, 1, \ldots)$$

and

$$\ell(2, y_t) = (1/2, 0, 1, 0, 1, 0, \ldots)$$

with 2 actions. The total losses of both actions are about $T/2$ after T rounds, while FTL suffers $\Theta(T)$ regret if the agent is deterministic and does not have any knowledge of the loss sequence. However, as suggested by Hannan's work [59], this issue can be fixed by adding a small perturbation $\mu(x)$ to the cumulative loss. The spirit of perturbation also motivates the more general no-regret framework FTRL [25]. Under standard FTRL, the agent selects the strategy for the next round as

$$x_T = \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T-1} \ell_t(x) + \frac{\mu(x)}{\eta}, \tag{18}$$

5) Decentralized is more common in multi-agent learning.
6) Ref. [62] showed uncoupled learning dynamics cannot guarantee the convergence to Nash equilibrium but to a CCE.

or equivalently when it refers to utility,

$$x_T = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T-1} u_t(x) - \frac{\mu(x)}{\eta}. \tag{19}$$

The regularizer $\mu(x)$ is often strongly convex in FTRL. When $\mathcal{X}$ is a n-simplex, let $u(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{u}\rangle$ denote the expected utility in the vector form and by setting $\mu(\boldsymbol{x}) = \sum_i^n \boldsymbol{x}_i \log \boldsymbol{x}_i$ and $\mu(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$ where $\boldsymbol{x}_i$ is the i-th component of $\boldsymbol{x}$, FTRL recovers Hedge and RM for discrete action space, respectively. FTRL framework also provides a general upper bound $O(\frac{1}{\sqrt{T}})$ with a strongly convex regularizer, which matches the lower bound in the worst case given by [63] and is unimprovable in fully adversarial setting if we do not prescribe the loss sequence. Online mirror descent (MD) is another no-regret algorithm framework from online convex optimization (OCO) [64]. In each iteration of MD,

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \eta\langle x, \nabla G_t(x_t)\rangle + D_{\mathcal{R}}(x, x_t), \tag{20}$$

where $D_{\mathcal{R}}(x, y) = \mathcal{R}(x) - \mathcal{R}(y) - \langle \nabla\mathcal{R}(y), x - y\rangle$ is the Bregman divergence with respect to a strictly convex generating function $\mathcal{R}$ and G_t is a convex function. MD is derived from the proximal point view, while FTRL and MD can result in the same implementation; e.g., Hedge can be derived from both FTRL and MD. The relation between online learning and online optimization is very close (see [65, 66]), especially for FTRL and MD.

Faster convergence with optimistic terms. In general, the loss sequence encountered is not always worst-case. Although the individual regret of the player is $O(T^{-1/2})$ in a fully adversarial setting, the optimal regret in the more benign setting is of interest, e.g., when all the players follow the same learning algorithms. It is established for normal-form 2p0s games in [67] that the individual regret of each player can converge at the rate of $O(\frac{\log T}{T})$ when both players follow the proposed no-regret protocol. The performance is near-optimal in the sense that the optimal performance is $O(\frac{1}{T})$. Moreover, this result is obtained under the strongly-uncoupled learning dynamics in which the players are restricted to only a limited history or the statistics of the history in case the players can recover the full game and call a Nash equilibrium solver. The upper bound can also be greatly improved when predictive terms of the next round M_{t+1} are adopted in predictive FTRL and optimistic mirror descent (OMD) [26, 68–70]. FTRL with the predictive term is also known as optimistic FTRL (OFTRL). The strategy for the next round with discrete action in OFTRL is

$$\boldsymbol{x}_{T+1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \left\langle \boldsymbol{x}, \sum_{t=1}^{T-1} \boldsymbol{u}_t + \hat{\boldsymbol{u}}_{T+1} \right\rangle - \frac{\mu(\boldsymbol{x})}{\eta}, \tag{21}$$

where $\hat{\boldsymbol{u}}_{T+1}$ is the predictive utility of round $T+1$. Letting $\hat{\boldsymbol{u}}_{T+1} = 0$, OFTRL incorporates FTRL. The general framework of OFTRL is shown in Algorithm 3. Using predictive terms, OFTRL can yield a $O((1 + \sqrt{\sum_{i=1}^T \|\hat{\boldsymbol{u}}_t - \boldsymbol{u}_t\|_\star^2})/T)$ regret bound, where $\|\cdot\|_\star$ is the dual norm to $\|\cdot\|$. It implies that the regret is bounded by the prediction error and will be constant if the prediction is perfect. If we choose a lazy predictor, i.e., $\hat{\boldsymbol{u}}_t = \boldsymbol{u}_{t-1}$, it is easy to see that the regret is bounded by $\sum_{i=1}^T \|\hat{\boldsymbol{u}}_t - \boldsymbol{u}_t\|_\star^2$, the gradual variations of the utility sequence, which is first introduced in the regret analysis in [71]. A later work [27] generalized the gradual variation and proposed the regret bounded by variation in utilities property (`RVU`).

Definition 1. A vanishing regret algorithm satisfies the `RVU` property with parameters $\alpha > 0$ and $0 < \beta \leqslant \gamma$ and a pair of dual norm $(\|\cdot\|, \|\cdot\|_\star)$ if the regret on any sequence of utilities $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_T$ is bounded as

$$\sum_{t=1}^T \langle \boldsymbol{x}^\star - \boldsymbol{x}_t, \boldsymbol{u}_t\rangle \leqslant \alpha + \beta \sum_{t=1}^T \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|_\star^2 - \gamma \sum_{t=1}^T \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2. \tag{22}$$

Definition 1 stated in [27] is an instance of the predictor $\hat{u}_t = u_{t-1}$, however, this definition can be extended to any predictor, and the proof in that paper still holds. Then the authors of [27] showed both OMD and OFTRL satisfy the `RVU` property. More importantly, the cumulative regret can be significantly reduced, e.g, each player's individual regret increases as $O(T^{1/4})$ under some mild conditions implying the convergence to a CCE in general-sum games with a rate of $O(T^{-3/4})$. On the other hand, the sum

Algorithm 3 General (optimistic) FTRL framework

```
Set u_0 = 0;
for t = 1 to T do
    if use standard FTRL then
        Set the optimistic term û_t = 0;
    end if
    Compute the strategy as (21);
    Observe utility u_t and compute the cumulative utility;
end for
```

of each player's regret can even be a constant when the predictor and the parameter η in OFTRL and OMD are chosen appropriately (see Corollary 6 and 8 in [27]), albeit the regret can be negative for some plays [72]. Thus, `RVU` property serves as a powerful tool to design novel OFTRL algorithms with faster convergence. The results from [27] demonstrate the sum and individual regrets of each player under optimistic Hedge are remarkably lower than the vanilla Hedge.

The pursuit of faster convergence of FTRL family never stops. Later studies that extend OFTRL to more general settings [73, 74], with faster convergence rate in specific games. The regret for standard optimistic Hedge is improved to $O(T^{1/6})$ in [75] for 2-player general-sum games in terms of external and swap regrets, which are stronger regret notions[7]. More importantly, the above regret for optimistic Hedge is further strengthened to $O(\log^4 T)$ in [76] for multi-player general-sum games. Note the predictive terms $\hat{\boldsymbol{u}}_{t+1}$ in [75,76] are both set as u_t, resulting in $\boldsymbol{x}_{t+1} \propto \exp(\eta(2\boldsymbol{u}_t - \boldsymbol{u}_{t-1}))$. Kernelized MWU/OMWU [77] generalizes these results of MWU and OMWU for normal-form games to extensive-form games with a kernel-based reduction, using only linear time per iteration, and obtains $O(\log^4 T)$ in extensive-form games. The authors of [78] develop new techniques to extend the results of [76] from external regret to internal and swap regret, thus obtaining $O(\log^4 /T)$ bound. Recently, LRL-OFTRL [79] improves the regret for OFTRL to $O(\log T)$ with the lifting trick in the more general convex game setting, which linearly scales the strategy space first (in the lifting space) and then normalizes back to the original space to get a valid strategy. There are also extensive studies that present faster convergence rate for OFTRL (especially optimistic Hedge), and other variants of FTRL and MD [72, 80–84].

Beyond the average convergence: last-iterate convergence. Besides the convergence of the average policy, an interesting byproduct of optimistic Hedge/FTRL is its last-iterate convergence (also point-wise convergence) in games. Late-iteration convergence ensures that the most recent policy will converge, allowing for the adoption of the latest policy when deriving the average policy is non-trivial. In practice, the last-iterate convergence is also noted for some variants of no-regret algorithms, e.g., in CFR+ [30] for Poker games. CFR-BR [85] proposes an asymmetric learning dynamics which runs a no-regret and a best response solver for each agent in 2p0s games, and shows the current policy[8] converges to a part of equilibrium with high probability. A line of recent studies [28, 81, 86–93] have studied the last-iterate convergence of the optimistic variants of no-regret methods, mainly from the online optimization perspective, e.g., OMD and extragradient. These studies usually do not follow the standard approaches in no-regret literature and have covered two-player zero-sum games, multi-player general-sum games, extensive-form games, and Markov games. However, the last-iterate policy often converges more slowly than the average convergence under the same assumption on specific games or converges to an approximate perturbed equilibrium, e.g., to the quantal response equilibrium (QRE) which adds the entropy of the current policy to the utility [94].

Another line of studies have studied the learning dynamics of no-regret algorithms in zero-sum games and general-sum games [62, 95–102], which tracks how the day-to-day behavior of the policy changes, i.e., the change of the current policy. Many studies show that the latest policy of MWU or the standard FTRL cycles and thus does not converge to a fully mixed Nash equilibrium in 2p0s games [96, 98, 99, 102]. On the contrary, by the volume analysis tool, the last-iterate policy of OMWU shrinks towards the equilibrium in zero-sum games [101], implying the optimistic term helps MWU achieve the last-iterate convergence. These studies provide empirical and theoretic understandings of the dynamics of learning in games and also motivate new variants of no-regret algorithms [39, 81] that achieve last-iterate convergence.

Remarks. Some of the analyses introduce advanced techniques to improve the upper bound for "old" algorithms, rather than proposing novel algorithms, while other studies utilize new learning rules. This implies some algorithms may perform better in practice. Interestingly, in large-scale games, those "faster"

7) The definition of external and swap regret can be found in Subsection 4.6 in [7].

8) In fact, the convergence property holds for any policy during the iteration.

no-regret learning algorithms may not outperform simple versions, partly because of computational issues. Although the theoretical foundation of no-regret learning is well-built, we will soon see the gap between theory and practice.

3.2 Regret matching and CFR family in Poker

Besides the study from online learning community, imperfect-information extensive-form games (IIG)[9], such as Poker games, are perhaps the most prevalent and successful playgrounds for no-regret algorithms. One fundamental change from the perfect-information game is that every player cannot observe the private information of others, e.g., in Poker games, even when the player observes the same play history and public information (his own cards and revealed cards), the cards in the others' hands may have multiple possibilities. These distinguishable information nodes fall into the same information state. Under the perfect recall assumption, Kuhn's theorem [103] shows the realization equivalence between the mixed strategy[10] and the behavioral strategy in IIG where each player can make decisions on the information state sequentially, and thus later studies focus on the behavioral form.

RM is favored by many practitioners since it is parameter-free, e.g., Hedge should choose η appropriately, and more computationally efficient than other no-regret variants. Counterfactual regret minimization (CFR) [24] is a milestone for solving heads-up limit Texas Hold'em (HULHE), which is a two-player zero-sum Poker game with about 10^{14} decision nodes. The authors of CFR propose a counterfactual utility of strategy profile σ at the information set I as

$$u_i(\sigma, I) = \frac{\sum_{h\in I, h'\in Z} \pi^{\sigma}_{-i}(h)\pi^{\sigma}(h, h')u_i(h')}{\pi^{\sigma}_{-i}(I)}, \tag{23}$$

where $\pi^{\sigma}_{-i}(h)$ is the counterfactual reach probability to the information node h, $\pi^{\sigma}_{-i}(I) = \sum_{h\in I} \pi^{\sigma}_{-i}(h)$ is the counterfactual reach probability to information state I, and $\pi^{\sigma}(h, h')$ is the probability of reaching terminal node h' from h. It is easy to see that player i always takes actions that will lead to h before reaching h, which is counterfactual, and then all the players follow σ. The counterfactual reach probability also cancels out player i's contribution to the probability of reaching h. Thus, the counterfactual utility denotes the expected utility of the information state I. Then the (average) immediate counterfactual regret is defined as

$$R^T_{i,\text{imm}} = \frac{1}{T}\max_{a\in A(I)}\sum_{t=1}^{T}\pi^{\sigma^t}_{-i}(I)(u_i(\sigma^t|_{I\to a}, I) - u_i(\sigma^t, I)), \tag{24}$$

where $\sigma|_{I\to a}$ denotes player i takes action a at I. Intuitively, the immediate counterfactual regret measures the average regret of not playing the best action a at the state I in hindsight, where I would be reached on each round if the player had tried to do so. Most importantly, the authors of [24] also prove that the overall (average) regret is bounded by the sum of the immediate counterfactual regret of all the information state of player i, which is a key property for more efficient implementation and computation. Since then, minimizing the immediate counterfactual regret at each state I is the main technique for finding an approximate Nash equilibrium using self-play in 2p0s IIGs. When applied in Poker games like heads-up limit Texas Hold'em, the authors first abstract the dealt cards, resulting in approximately 1.65×10^{12} game states and 5.73×10^{7} information states, and also use a domain-specific chance sampling trick to reduce the computation cost per iteration (the vanilla CFR needs to traverse the whole game tree). The final CFR agent is able to compute an approximate Nash equilibrium with roughly two orders of magnitude larger than previous methods and beats other strong Poker bots, including the AAAI 2006 Computer Poker Competition's winner. MCCFR [29] is a general framework for sampling in CFR and provides a domain-independent sample-based CFR variant. MCCFR only traverses a small portion of the full game which further reduces the per-iteration cost, thus leading to drastically faster convergence in the empirical studies, even though it requires more iterations. There are also variants of MCCFR [104, 105] using different sampling techniques.

9) We omit the formulation of IIG here. The readers may refer to Section 5 in [52] for details.

10) Intuitively, the mixed strategy of IIG is a distribution over all the possible pure strategies, where each pure strategy corresponds to a decision sequence from the root node to the leaf.

CFR+ [106] is a refinement of CFR, which replaces RM with RM+. Concretely, RM+ rectifies the regret as

$$R_i^{+,T}(a) = \begin{cases} (u_i(a, \sigma_{-i}^t) - u_i(\sigma^t))^+, \ T = 1, \\ \left(R_i^{+,T-1}(a) + u_i(a, \sigma_{-i}^T) - u_i(\sigma^T)\right)^+, \ T > 1. \end{cases} \tag{25}$$

Intuitively, RM+ neglects the negative part of the regret, which will be helpful when the "best action" suddenly changes. RM requires more iterations to identify the bad actions, while RM+ can quickly adapt to the good action. In addition, CFR+ uses no sampling (the performance degrades with sampling) and does alternating updates, where CFR described in [24] should simultaneously update the regrets for both players. The authors of CFR+ find the current policy almost converges to the equilibrium empirically and the weighted averaging schema where the weight for round t is proportion to t converges faster. This new variant, CFR+, is later proven to have at least the same regret bound of vanilla CFR [30], but significantly outperforms CFR in Poker games with both the weighted average and the last-iterate strategy. The authors of [107] showed that RM and RM+ are the instances of FTRL and online mirror descent, and thus it is able to embody advanced techniques from no-regret learning into RM+. The weighting schema also motivates the discounted variant[11] of CFR (DCFR) [108], where not only the early strategy is less weighted but also is the early regret. This idea is very intuitive: when "bad" actions incur large regrets at earlier iterations, discounting these less significant regrets may dramatically speed the convergence. DCFR has three parameters α, β, γ, denoted by $\text{DCFR}_{\alpha,\beta,\gamma}$ and is defined by multiplying the positive regrets by $\frac{t^\alpha}{t^\alpha+1}$, negative regrets by $\frac{t^\beta}{t^\beta+1}$ and using $(\frac{t}{t+1})^\gamma$ as the weight for the stragtegy on iteration t. The setting $\alpha = 3/2, \beta = 0, \gamma = 2$ in DCFR consistently outperforms CFR+. Unfortunately, the performance gain in practice is theoretically justified, since these later variants have only been proven to have at least the same convergence rate, and have not been directly proven to have a faster convergence rate.

The success of CFR in Poker games also relies on techniques that efficiently reduce the size of the game, mainly from three aspects. (1) Pruning: decide which parts of the game tree to traverse each iteration [109–111]; (2) subgame solving: solve a subgame and cast the strategy to the full game [112–114]; (3) abstraction: bucket similar cards or "strategically similar" actions together and treat them identically [115, 116]. For example, the abstraction algorithm can reduce the game size of heads-up no-limit Texas Hold'em from 10^{161} decision points down to 10^{12}. However, techniques like abstraction may require domain-specific knowledge. Regression CFR (RCFR) [117] makes use of a regression tree to learn the regret, and thus the abstraction is implicit and also learned. DeepCFR [118] incorporates deep models into CFR, combining with linear CFR ($\text{DCFR}_{1,1,1}$) and external sampling from MCCFR. DeepCFR outperforms domain-specific abstractions in a subgame of Texas Hold'em, appearing to be the first non-tabular variant of CFR to be successful in large games.

4 Reinforcement learning and games

When RL meets games, the game is always a Markov game. For general-sum Markov games, traditional MARL methods augment the action in the value function with joint actions of multiple agents, and then employ game-theoretic approaches like minimax value and Nash equilibrium value, to evaluate the value function of the next state, updating the value function and deriving policies for each agent. This approach is a centralized training and decentralized execution (CTDE) paradigm, which is also widely used in the current cooperative MARL field. However, this paradigm does not fit non-cooperative games, where each agent should learn from an individual perspective. Fully independent training is known to have non-stationarity issues due to the changing of the opponent. Hence, new decentralized methods include combining with game-theoretic frameworks, such as fictitious self-play, studying new RL-style learning algorithms, or directly considering the impact of opponents. Moreover, since RL is mainly sample-based, the theoretical analysis of MARL focuses more on sample complexity[12], that is, how many samples are needed for successful learning, while no-regret learning frameworks typically analyze time complexity.

Another category widely used in practice is referred to as open-ended learning, also known as population-based training. The theoretical foundation is empirical game-theoretic analysis, which aims to study various properties of the game based on the payoff of different strategies, especially when only black-box

11) Discount the regret is also investigated in [7] (Subsection 2.11).

12) Some studies also directly analyze the time complexity.

Algorithm 4 General multi-agent Q-learning

Require: N (number of agents), M (number of episodes), T (number of steps per episode), α (learning rate), γ (discount factor), ϵ (exploration rate), and subroutine Eval (Minimax, NE, CCE, FFQ, etc.);
Initialize $Q_i(s, \boldsymbol{a})$ for all agents $i = 1, \ldots, N$, states s, and actions $\boldsymbol{a}$;
for $m = 1$ to M **do**
  Initialize state s;
  for $t = 1$ to T **do**
    for $i = 1$ to N **do**
      Compute action $(a_1^*, \ldots, a_N^*) = \text{Eval}(Q_1(s, \boldsymbol{a}), \ldots, Q_N(s, \boldsymbol{a}))$;
      With probability ϵ, agent i selects a random action a_i; otherwise, agent i selects action $a_i = a_i^*$;
    end for
    All agents execute their action a_i, observe state s', and receive their reward r_i;
    for $i = 1$ to N **do**
      Compute action for next state $\boldsymbol{a}'^* = (a_1'^*, \ldots, a_N'^*) = \text{Eval}(Q_1(s', a'), \ldots, Q_N(s', a'))$;
      $Q_i(s, \boldsymbol{a}) \leftarrow Q_i(s, \boldsymbol{a}) + \alpha[r_i + \gamma Q_i(s', \boldsymbol{a}'^*) - Q_i(s, \boldsymbol{a})]$;
    end for
    Update state: $s \leftarrow s'$;
  end for
end for

access to the game is available. In practice, the game environment is typically treated as a black box, with RL and other oracles to solve the optimal policy for a given opponent. The research focus is on how to select opponents (solving meta-strategies) so that the trained policies can converge to desired objectives. Open-ended learning also includes some heuristic designs, such as promoting the diversity of the policy population.

The subsequent contents will delve into these two categories in detail.

4.1 Reinforcement learning in games

4.1.1 *Centralized training*

Although single-agent RL algorithm, e.g., Q-learning, can be applied to the multi-agent case as done in [119], naive independent single-agent RL algorithm may not converge in multi-agent systems (MAS). The naive independent learner treats other agents as parts of the environment, while when all the agents are simultaneously learning, they all face a constantly changing environment. This is identified as the non-stationarity issue of the multi-agent learning problem. The theoretic guarantees are violated since traditional single-agent RL algorithm like Q-learning only converges in stationary environment. A quick fix to Q-learning for multi-agent learning is to condition the Q function on the joint action $\boldsymbol{a} = (a_i, a_{-i})$. Minimax-Q [31] is such an extension of the Q-learning method to solve two-player zero-sum Markov games, which update Q_i for agent i as

$$Q_i(s, \boldsymbol{a}) \leftarrow Q_i(s, \boldsymbol{a}) + \alpha(r(s, \boldsymbol{a}) + \gamma V_i(s') - Q_i(s, a_i, a_{-i})), \tag{26}$$

where $V_i(s') = \max_{\pi_i} \min_{a_{-i}} \mathbb{E}_{a_i \sim \pi_i}[Q(s, a_i, a_i)]$ is evaluated with a minimax strategy. Friend-or-foe Q-learning (FFQ) [32] generalizes minimax-Q with a team of maximizing agents and a team of minimizing agents, and thus fits in cooperative, competitive, and mixed settings. Nash-Q and CE-Q [33,120] replace the evaluation of V_i by Nash equilibrium and correlated equilibrium strategy, respectively. A general multi-agent Q-learning framework is shown in Algorithm 4. However, many of these extensions use a linear program in the evaluation, thus slowing down these algorithms in practice. Note although the Q function may reply on centralized information (the joint action) for training, the derived policy should be decentralized to execute. These studies pioneer the CTDE framework [121], which lays the foundation of many recent multi-agent reinforcement learning algorithms in fully cooperative POMDPs [122–125] or mixed cooperative-competitive environments [126]. There also exists other decentralized RL studies applied to multi-agent learning [127–130], with variable learning rates or predictions to avoid the non-stationarity.

4.1.2 *Decentralized training*

RL with self-play. For non-cooperative games, especially 2p0s games, a domain-knowledge-free training framework called self-play with reinforcement learning has become the first choice in practice, which plays against itself and improves the policy by RL. It plays an important role in large-scale games like AlphaGo and AlphaStar, and appears to be overwhelming. Fictitious self-play (FSP) [34] implements

Algorithm 5 General fictitious self-play

1: **for** each iteration **do**
2: Find the approximate best response ϵ-BR($\hat{\sigma}_{-i}$) by RL;
3: Sample and store trajectories against the empirical strategy $\hat{\sigma}_{-i}$ of the opponent;
4: Update the empirical of the opponent;
5: Update the average policy only with best response actions by supervised learning;
6: **end for**

the generalized weaken fictitious play (GWFP) framework [131] in a sample-based fashion with RL, and applies to Leduc Poker to find an approximate NE by self-play. Algorithm 5 shows the main process of FSP[13)], where it replaces the solver for the best response and the average strategy with RL and supervised learning, respectively, and supports function approximation tools. Neural FSP (NFSP) [35] proposes a deep networks implementation of FSP, solving the best response policy with DQN and distilling the average policy from a reservoir buffer [132]. The FSP framework is agnostic about the game's specifics, except that it solves a two-player zero-sum game, and can also be applied in Markov games [133, 134]. Though FSP is general enough, it is less efficient than DeepCFR in Poker from the experimental results [118].

Policy gradient of learning in games. Recalling the equivalence between the advantage function and the instant regret, it is interesting to study the policy gradient RL families when learning in games, which use the advantage function to weight the gradient. From the policy gradient perspective, it has been revealed that the V function of PG method is the scaled counterfactual regret value in imperfect-information games [36], leading to a new interpretation of actor-critic algorithms in POMDPs. For discrete action space, the policy is often parameterized as a softmax function over the logits of an action a in RL, i.e., $\pi(a) = \frac{\exp(y(a))}{\sum_a' \exp(y(a'))}$, where $y(a) \in \mathbb{R}$ is the logits of action a. NeuRD [37] studies the connection between the softmax PG algorithm and replicator dynamics (RD), which is the widely-used model in evolutionary game theory [135]. NeuRD implements RD through the lens of PG-style updates on all actions, contrasting to all-actions PG as

$$\text{All-actions PG: } y_t(a) = y_{t-1}(a) + \eta_t \pi_t(a) \underbrace{\left[u_t(a) - \sum_{a' \in \mathcal{A}} \pi_t(a') u_t(a')\right]}_{\text{the advantage } A(a)}, \tag{27}$$

$$\text{NeuRD: } y_t(a) = y_{t-1}(a) + \eta_t \cancel{\pi_t(a)} \underbrace{\left[u_t(a) - \sum_{a' \in \mathcal{A}} \pi_t(a') u_t(a')\right]}_{\text{the advantage } A(a)}, \tag{28}$$

where $y_t(a)$ is the logits before the softmax function. NeuRD differs from standard PG by removing the extra $\pi_t(a)$ weighted on the advantage, which can be derived from the RD perspective. NeuRD also proves to recover Hedge algorithm in the tabular single-state setting. Based on these findings, we can perform regret minimization using PG-style reinforcement learning algorithms, with very minor modifications. Motivated by CFR-BR in 2p0s games, exploitability descent (ED) [38] evaluates values of the current policy against its best response (the exploitability of the current policy), $q^{\boldsymbol{b}}(s)$, and then uses PG to update the current policy based on $q^{\boldsymbol{b}}(s)$. However, as the exploitability can now be tracked, ED can pick the best iterate rather than uniformly sampling from a proportion of p iterations as CFR-BR, and thus improves the probabilistic guarantee of the CFR-BR to a deterministic one and reducing the equilibrium gap. Although ED does not require the average policy, the best iterate may be different from the last-iterate policy. Moreover, independent PG update has been proven to converge to the min-max equilibrium in two-player zero-sum Markov games if both players run independent PG in tandem and the learning rates follow the two-timescale rules [136]. This is the first finite-sample convergence result for fully independent learning with RL in competitive games. Ref. [39] showed that the cyclic behavior of the current policy in the original FTRL in 2p0s games can be circumvented, by adding a policy-dependent term into the reward. It first generalizes the negative cyclic results of [98] from norm-form games to IIG, and then shows the last-iterate convergence can be achieved by model-free RL with the

13) In order to improve the sample efficiency, the original FSP uses the average policy to pit against the opponent and off-policy RL algorithm to update the BR.

transformed reward in monotone games[14]. This is an important step for reinforcement learning in games, and later helps to build top AI in large-scale IIG with purely PG-style learning (NeuRD is adopted in the experiment).

Learning with opponent-awareness. The above approaches treat the opponent as a black box. Intuitively, if the agent knows how others will act in advance, multi-agent learning will become simpler. This idea falls into the opponent modeling approach in multi-agent learning. With the opponent model, the agent will have the ability to adapt to the specific opponent. For example, in fictitious play, the player implicitly models the opponent by tracking the historical plays and always best responds to them. A line of studies model the opponent policy by generative models [40,137–139], and update the agent's own policy with the opponent policy. There exists a more complicated recursive reasoning model in opponent modeling, such as level-K thinking [140] which originates from the theory of bounded rationality [141] and is a popular behavior economic model. Level-K model has been widely used in multiagent learning as an opponent modeling approach [142–144]. Learning with opponent learning awareness (LOLA) [41] considers how the opponent's learning will impact the learning of the primary agent. Compared with the above opponent modeling approaches which mostly rely on past experiences with the opponent, LOLA looks one step ahead: LOLA agent updates itself after the opponent takes a gradient update, i.e.,

$$\theta_1 \leftarrow \theta_1 + \eta \cdot \nabla_{\theta_1} V_1(\theta_1, \theta_2 + \Delta\theta_2), \tag{29}$$

where $\Delta\theta_2 = \delta \cdot \nabla_{\theta_2} V_2(\theta_1, \theta_2)$. When both agents use LOLA in iterated prisoner's dilemma (IPD), reciprocity-based cooperation emerges while the independent naive learner fails to cooperate. LOLA also leads to more stable learning of the NE. The original implementation of LOLA uses a first-order approximation, while DiCE [145] introduces a higher order estimator to avoid the first-order approximation and greatly improves the sample efficiency. LOLA model other agents as naive learners, while other agents are also LOLA agents. Thus, the actual other agents encountered are inconsistent with the ones anticipated. The following work COLA [42] addresses this inconsistency issue by minimizing a pair of mutual consistent losses. POLA [43] identifies that the original LOLA is sensitive to policy parameterization which is another failure mode of LOLA, and then it uses a proximal update with a unique solution assumption to guarantee POLA update is invariant to policy parameterization. Thus, if the original policies are the same, the new policies after the POLA update will also be the same. However, due to the computation burden of the higher-order derivatives, experiments on LOLA variants have only been conducted on small-scale games.

Theoretical aspects and sample complexity. Recently, extensive articles have studied the sample complexity of learning in Markov games from the RL perspective, in the bandit feedback setting [146–150], in the full-information feedback setting [91, 93, 151–154], or with function approximation [155–157]. Importantly, learning ϵ-stationary CCE/NE is intractable (PPAD-hard) in general-sum Markov games [150], which is in stark contrast to single-agent RL where sample efficient algorithms for near-optimal stationary policy exist. The authors of [150] then provide a decentralized algorithm to learn a nonstationary Markov CCE policy with polynomial time and sample complexity. The most recent [158] designed a new decentralized RL algorithm for general-sum Markov games in tabular and linear function approximation settings, respectively, improving the result of [150] by an order of magnitude.

Remarks on cooperative MARL. It is noteworthy that there is currently another prominent topic in MARL that focuses on cooperative games (also coalition game theory). The formulation usually used is the Decentralized POMDP, where each agent only has a partial observation of the environment and follows its decentralized policy. Importantly, there is not an individual reward signal for each agent; instead, the team members share a single reward signal, and these agents aim to maximize the cumulative reward collectively. A crucial challenge in cooperative MARL is the credit assignment, which facilitates collaboration between agents. Representative methods include value-decomposition approaches like VDN [122], QMIX [124], QTRAN [159], QPLEX [125], and multi-agent policy gradient methods such as MADDPG [126] and MAPPO [160]. VDN aims to decompose the team value function into the individual value functions of each agent through a simple additive decomposition. Following the Individual-Global-Maximum (IGM) principle [159], QMIX improves VDN by learning a nonlinear mixing network. QPLEX introduces the Advantage-Individual-Global-Maximum principle, employing a dual-dueling network architecture to decompose the joint value function. MADDPG utilizes the "actor-critic" architecture,

14) As the authors' note, this definition is hard to interpret, but it captures a wider class of games, e.g., 2p0s games and polymatrix zero-sum games.

Algorithm 6 Policy space response oracle

Input: Initial policy sets Π for all players, compute utilities U^{Π} for each joint $\pi \in \Pi$, initialize meta-strategies $\sigma_i^1 = \text{UNIFORM}(\Pi_i)$;
while round $t = 1, 2, \ldots$ **do**
  for player $i = 1, 2, \ldots$ **do**
    Sample opponent by $\pi_{-i}^t \sim \sigma_{-i}^t$; /* Choose the opponent
    Training over π_{-i}^t and obtain π_i'; /* Call training oracle
    $\Pi_i \leftarrow \{\pi_i'\} \cup \Pi_i$;
  end for
  Evaluate current population and compute meta-strategy σ^{t+1}; /* Call meta-strategy solver
end while
Output: Meta-strategy σ_i and population Π_i for player i;

optimizing each agent's policy through the single-agent DDPG algorithm [161]. However, the high variance of MADDPG makes it less competitive. MAPPO and many of their variants apply PPO [162], which is widely used in single-agent RL, to multi-agent reinforcement learning, effectively enhancing the system's collaborative capabilities. These approaches are primarily trained using the CTDE paradigm. Recent research has identified potential conflicts of the learning objectives in the multi-agent policy gradient update targets of previous studies, leading to the introduction of sequential update methods like HAPPO [163] and MAT [164]. Nevertheless, the above methodologies primarily focus on how teams learn. In contrast, this paper mainly focuses on individual learning, i.e., learning in non-cooperative games.

4.2 Empirical game and open-ended learning

A unified open-ended learning framework: PSRO. The goal of empirical game-theoretic analysis (EGTA) [165] is to analyze or estimate various properties of a game, given only noisy black-box access to it. Thus, it can be used for analyzing complex games, which extends strategies iteratively based on experience with prior strategies and solves for the meta-strategy over the stored strategies to apply in the original game. The empirical game (also meta-game) is modeled by an open-ended payoff table where each item is a payoff vector under a joint strategy. The payoff table will extend when adding a new strategy and thus is open-ended[15]. Note that a single strategy in EGTA can be very complex, while only its empirical performance is the focus. Double oracle (DO) [166] is the first work that tries to find the minimax equilibrium for two-player games under EGTA. DO randomly initializes two policy sets for the row and column players, and in each iteration, DO takes the following steps.

(1) Solve for the equilibrium (p, q) of the matrix game M induced by the current policy sets $\mathcal{R}, \mathcal{C}$, and take p over $\mathcal{R}$ and q over $\mathcal{C}$ as the current policy of each player.

(2) Find a pure best response to the opponent's current policy and add it to the policy set.

As DO requires the best response to be pure, it is guaranteed to converge when no pure best response can be added. In this case, DO will run an extra step to solve for the equilibrium with the policy sets, and the solution is proven to be the equilibrium of the original game. In the worst case, DO will end up adding all the pure strategies. Fortunately, it is reasonable to expect that the support of the equilibrium is much smaller in complex games.

Policy space response oracle (PSRO) [44] generalizes DO by replacing steps 1 and 2 with a meta-strategy solver and a training oracle, respectively, as summarized in Algorithm 6. PSRO recovers DO when the meta-strategy solver is the NE and the training oracle is the BR oracle, so it shares the same convergence guarantee when all the pure strategies are included. PSRO can also recover fictitious play if the meta-strategy solver is a uniform distribution over all the past policies. For instance, the opponent pool with uniform sampling in AlphaGo can be viewed as an instance of fictitious play.

Refinements of PSRO. To generate useful and better opponents, PSRO$_{rN}$ [45] uses NE as the meta-strategy solver and ignores policies it loses to. By ignoring the weakness, PSRO$_{rN}$ uncovers strategic diversity more efficiently. For complex games, solving for exact BR is intractable, thus approximate BR solvers like RL are widely adopted in the PSRO family. Thus, the total iterations that PSRO needs to converge become the bottleneck. PSRO provides a naive parallel variant named DCH, but it may fail to converge even in small games. PSRO$_{rN}$ is shown to diverge in small games in conjunction with DCH in [46]. Pipeline PSRO (P2SRO) [46] is a scalable parallel version of PSRO which shares the convergence guarantee as PSRO by maintaining a hierarchical pipeline of reinforcement learning workers, each training against the policies generated by lower levels in the hierarchy. The hierarchical structure speeds up the

15) Some literature also uses the term unrestricted.

training when the parallel studies increase. For IIG, as PSRO may need to expand all pure strategies in the meta-game, which grows exponentially in the number of information states, it may require an exponential number of iterations to approximate. Extensive-form double Oracle (XDO) [167] is designed for IIG and is guaranteed to converge to an approximate Nash equilibrium linearly in the number of states for 2p0s games. Another practical issue of PSRO is that the exploitability of meta-strategy in PSRO may increase from iteration to iteration, so it is not guaranteed to always get lower exploitability when the number of iterations increases. Two concurrent studies, Anytime PSRO (APSRO) [168] and Efficient PSRO (EPSRO) [169], allow the opponent or player to access the full policy space when solving for the meta-strategy, thus solving an unrestricted-restricted (URR) game, while PSRO restricts both players to the current policy population for meta-strategy. Both APSRO and EPSRO prove to have a non-increasing exploitability guarantee.

How to select the policy from the population: meta-strategy solvers. As for the meta-strategy, Nash equilibrium, the uniform distribution, and other preference-based distributions, such as Elo ratings [170] and prioritized FSP in [16], have been widely adopted. For general games, solving for NE is PPAD-complete and other distribution may become unsafe. Even for symmetric 2p0s games, the Elo rating cycle in the intransitive part of the game and NE may suffer from the equilibrium selection problem. For instance, in Rock-Paper-Scissors, if we add a duplicate of Rock, there will be infinite many NE which assign 1/3 to Paper and Scissor, respectively, and a total of 1/3 to the two duplicate Rock strategies. MaxEnt NE (or Nash averaging) can provide a unique solution in zero-sum games, and it automatically adapts to redundancy policies [171]. Unfortunately, the selection issue persists for Nash in general games. The author of [172] proposed an alternative solution concept, α-Rank, through the lens of evolutionary theory. They model the fixation probability of a focal population (a previously-monomorphic population wherein a rare mutation has appeared) under the single mutant strategy and then construct the Markov transition matrix for each pair of strategies. The stationary distribution of the Markov chain gives the surviving probability of each monomorphic population after a long-term evolution. As the transition matrix is irreducible and aperiodic, there always exists a unique stationary distribution, and it can be computed in polynomial time. α-PSRO [47] replaces the meta-strategy solver by α-Rank, and can apply to the multi-player general-sum setting more efficiently than prior PSRO applications.

Practical methods for improving the efficiency. The theoretical guarantee of PSRO and other EGTA methods is essentially based on an accurate evaluation of a pair of policies in the current population, while for complex games, practitioners often evaluate the performance by many rounds of simulations. Thus the evaluation is noisy and expensive. There exist studies [173–175] that handle the uncertainty and reduce the rounds of simulations to improve the efficiency. Another approach focuses on how the population is maintained and used to improve the learning efficiency. Mixed-Oracles and Mixed-Opponent [176] can generate a new BR policy or opponent by a combination of previous Q functions, thus transferring knowledge from previous iterations. NeuPL [177] represents a population of policies within a single conditional model. It also enables the transfer of previous knowledge across policies. Simplex NeuPL [178] generalizes NeuPL to learn the best responses to any mixture over the simplex of diverse basis policies in symmetric 2p0s games.

Population diversity. Population diversity is crucial for population-based learning like PSRO, since it measures the population rather than just an aggregated joint policy. Its importance is twofold: (1) it helps explore the strategy space sufficiently which is akin to the exploration of RL, and (2) it avoids being exploited by unknown malicious opponents, especially in non-transitive games. Many studies apply RL exploration techniques to learn a novel policy [179, 180]. Determinantal point processes (DPP) [181] is a probabilistic framework that produces diverse subsets by sampling proportionally to the determinant of the kernel matrix of points within the subset. Motivated by the application of DPP in machine learning, some studies use DPP as a tool to promote diversity for population-based learning [48, 49]. DvD [48] applies DPP to the behavioral embeddings of multiple policies, while G-DPP [49] applies DPP to a Gram matrix spawned by the payoff matrix. These studies fall into two categories: behavioral diversity (BD) and response diversity (RD), where BD focuses on the trajectory-wise or state-action diversity, and RD focuses on the payoff diversity. A later work [50] unifies BD and RD, and proposes a new approach to jointly optimize the trajectory diversity and the payoff discrepancy against previous policies. However, there is a subtle difference between different policies and diverse policies. For example, in Rock-Paper-Scissors, the policy $(0.99, 0.01, 0)$ is different from $(0.98, 0.02, 0)$, but they are not diverse. To better measure the diversity, in the paper of PSRO$_{rN}$ [45], the authors propose effective diversity (ED) to quantify how the best agents (those with support in the maximum entropy Nash) exploit each other.

Population diversity (PD) in [48] uses the determinant of the kernel matrix which represents the volume of a parallelepiped spanned by feature maps. Expected cardinality (EC) [49] is used to measure the cardinality of a random sample from G-DPP, so that the policy population will be more diverse in terms of the payoff if EC is higher. Population effectivity (PE) proposed in [50] is a more general opponent-free concept of exploitability by considering the optimal aggregation of the current population in the worst case. Recently, UDM [51] proposes a unified diversity measure that captures ED, PD, EC, and PE by a general function which is a power series with a positive first-order derivative everywhere. Though increasing diversity is very intuitive, whether it leads to better learning efficiency is currently a bit unclear in theory.

5 Towards a unified learning framework

RL plays a more and more important role in learning in games, and the game-theoretic approach to (multi-agent) RL, or the so-called game-theoretic RL, has drawn much attention. Many algorithms have been proposed under game-theoretic RL. The design philosophy of these algorithms is roughly concerning the following aspects.

Decentralized/uncoupled vs. centralized/coupled learning. Canonical learning models such as fictitious play and regret minimization study the behavior of each individual so that they are fully decentralized. CTDE paradigm is proposed for addressing the non-stationarity issue in MARL. In a more general sense, centralized training may include knowledge of the game, the opponent's strategy and even the use of global oracle information. These techniques input the centralized information to the value function and thus reduce the non-stationary training difficulty, at the cost of higher dimensional strategy space. Some strongly coupled learning dynamics can even achieve much faster convergence speed [83], while uncoupled learning can avoid the curse of dimensionality when the number of players is large.

Average vs. last-iterate convergence. These two kinds of convergence can approximate NE, and as mentioned above, some algorithms can have both no-regret property and last-iterate convergence. Nevertheless, the choice can be different for specific tasks. No-regret learning naturally studies the average convergence by definition, while showing last-iterate convergence usually requires non-standard techniques other than no-regret analysis tools. As the exact average policy in complex games is computationally inefficient to obtain, we should adopt algorithms with a last-iterate convergence guarantee if we aim to approach the equilibrium in complex games. For instance, as mentioned in ACH [182], the authors add an entropy regularization to the current policy which is theoretically justified in [39], without the calculation of the average policy. From the practical view, the last-iterate convergence is always preferred, or the average is easy to conduct. On the other hand, if we aim to adapt to a non-stationary opponent/environment, no-regret algorithms may be a better choice where the convergence properties of no-regret families are usually better.

Population learning vs. single policy. DeepNash [18] reveals that a single policy can be trained to master complex games without pitting against historical policies, saving space and time for evaluating the population. Population learning, especially PSRO variants, requires an aggregate of the population, which may be intractable for complex games. Mixed Oracles [176] and NeuPL [177, 178] work towards solving the aggregation challenge with a single policy. The reasons for population learning are that these historical policies can provide strong but diverse baselines when the game is highly intransitive, and also make it possible to adapt to a specific opponent. As an aside, real-world games consist of many highly intransitive parts as demonstrated in [183]. Besides, population learning combined has facilitated curriculum learning, e.g., in automatically generating a curriculum of increasingly challenging tasks/environments [184–186].

In the realm of RL, policy optimization or reward maximization refers to finding a policy that maximizes the expected return with or without some regularized terms. Regret minimization methods usually find a strategy in each iteration that minimizes the cumulative regret with regularizers. Earlier studies of no-regret learning and RL seem to develop independently since RL solves an MDP, which is fundamentally different from the matrix game that no-regret learning usually handles. Thanks to the approximate dynamic programming (ADP) techniques, the policy optimization problem in RL can be decomposed as independent policy optimization problems in every state. While for the regret minimization family, CFR proves that if the regret on each information state is bounded then the total regret of the EFG is also bounded, and thus extends RM from normal-form games to extensive-form games. These contributions

make tools that solve the single-state problem sound again. With recent advances in deep RL and no-regret/online learning, they greatly overlap with each other. For instance, deep regret minimization methods, e.g., DeepCFR, ARMAC [187], and DREAM [188], use the advantage function in RL to calculate the regret. RLCFR [189] incorporates an RL agent to select the strategy for calculating and updating the regret by RM, DiscountCFR, and Hegde. These are examples of no-regret learning meeting RL. On the other hand, ARM [190] is the first to introduce CFR+ into RL for solving POMDP which approximates the regret by the cumulative advantage function. To our knowledge, ARM is the first step towards policy optimization with no-regret learning. RegretPG [36] and NeuRD [37] both guide the policy gradient toward a no-regret direction.

Recall the connection of FTRL and MD, which can derive the same no-regret algorithm, e.g., Hedge. By appropriately choosing the objective function in MD, MD can be implemented in the policy optimization manner. MDPO [191] is such work that directly applies mirror descent in RL as a new policy optimization approach, which is closely related to two popular RL algorithms PPO [162] and SAC [192]. These signs reveal that policy optimization and regret minimization can be implemented under the same learning framework. In other words, we can move from reward maximization to regret minimization freely. We have noticed the techniques of training RL agents can be seamlessly applied to learning in games. We believe no-regret learning and RL are stepping towards a unified learning framework, just as FTRL unifies no-regret learning.

The unified approach to regret minimization and reward maximization can be implemented by transforming the reward to a policy-dependent version as [18, 39] or adding extra divergence terms to the original policy optimization objective like [193–195]. For both approaches, the objective in each iteration can be unified as the following objective:

$$\pi_{t+1} = \arg\max_{\pi} Q_t^{\pi} - \alpha D_{\mathcal{R}_1}(\pi, \pi_{\text{ref}}) - \eta D_{\mathcal{R}_2}(\pi, \pi_t), \tag{30}$$

where $\mathcal{R}_1, \mathcal{R}_2$ are strongly convex regularizers and $\eta_1, \eta_2 > 0$. π_{ref} is a reference policy. Note Q_t^{π} should be learned under the perturbed reward by $D_{\mathcal{R}_1}(\pi, \pi_{\text{ref}})$, rather than the original reward. When it is the uniform distribution and $\mathcal{R}_1 = \sum x \cdot \log x$, $D_{\mathcal{R}_1}(\pi, \pi_{\text{ref}})$ becomes the entropy of π. Intuitively, π_t stores historical regret information thus removing the cumulative regret. If all the agents follow (30), the policy of each agent can be no-regret and has last-iterate convergence for the very specific 2p0s games. Besides, Eq. (30) is an RL-style objective, so it can be updated with deep RL algorithms in a fully decentralized way for large-scale games. Thus, we can perform regret minimization and RL with the same base learning algorithm.

6 Applications in large-scale games

Extensive-form games (turn-based). In perfect-information board games, one of the key techniques is Monte Carlo tree search (MCTS). Based on MCTS, AlphaGo [12] combines human play initialization, RL, and population training to defeat the top Go professional in 2016, and starts the era of the Alpha series in board games. Following studies AlphaGoZero [196] and AlphaZero [197] conquer Go and other board games without human initialization. MuZero [198] further removes the dependency on a perfect simulator (the transition is known and can reset to any given state) by performing the tree search in a learned model. MuZero, without any knowledge of the underlying game dynamics, matches the performance of AlphaZero and also achieves superhuman performance on a collection of Atari video games.

Some studies manage to combine RL and search for IIG [17,199,200], which use various forms of perfect information during training to make it technically sound. However, standard MCTS is not technically sound for imperfect-information EFG like Poker. For Poker, DeepStack [201] uses deep networks to predict the expected value for the purpose of depth-limited re-solving with CFR. Libratus [13, 202] uses the CFR variant and multiple CPU servers for real-time solving. Both DeepStack and Libratus defeat top humans in 1v1 no-limit Texas Hold'em in 2017, and Libratus even defeats top professionals by a large margin, which is a premiere for Poker AI. Modicum [203], as the name suggests, achieves a master-level play and runs in a 4-CPU laptop to beat some previous top bots. Pluribus [15] is a mature version of Libratus and Modicum, and is also the first AI applied to play with human players in 6-player Texas Hold'em in 2019. In both 1 AI vs. 5 humans and 5 AIs vs. 1 human mode, Pluribus outperforms human players, which is thought to be a major breakthrough.

Mahjong is a 4-player IIG which is very challenging for AI research due to its complex playing/scoring rules and rich hidden information. Suphx [17] combines RL, global reward prediction, oracle guiding, and Monte Carlo policy adaption. Suphx uses global reward prediction to give a more accurate evaluation of each action. The oracle agent in Suphx sees all the perfect information about a state to speed up the learning of the RL agent and gradually drops out the perfect features to transit to a normal agent in the end. ACH extends the actor-critic framework to minimize the counterfactual regret by Hedge and also beats a human champion in 1v1 Mahjong [182]. Doudizhu is another popular 3-player zero-sum card game in China, where two players (Peasants) cooperate to pit against one player (the Landlord). DeltaDou [199] is essentially a reproduction of AlphaGo in Doudizhu which uses Bayesian methods to infer hidden information and searches the moves with MCTS. After training for two months, the policy is shown to have on par performance with top human players. DouZero [204] uses Monte Carlo return as the target value for Q function and follows a ϵ-greedy(Q) policy and names it deep Monte Carlo (DMC). DouZero surpasses DeltaDou in ten days. DouZero+ [205] improves DouZero by opponent modeling and coach-guide learning. PerfectDou [206] is the current state-of-the-art Doudizhu AI which shares a similar spirit of Suphx. PerfectDou utilizes perfect imperfection distillation that allows the agents to utilize the global information to learn the critic but learn the actor under imperfect information.

DeepNash [18], a large-scale extension of the regularized Nash dynamics (R-NaD) from [39], masters the complex imperfect information board game Stratego by a policy gradient algorithm with the transformed reward. It is the first time an AI algorithm is able to learn to play at a human-expert level in a complex board game by purely RL-style algorithm end-to-end from scratch without deploying any search method or explicit opponent modeling. More importantly, except R-NaD, DeepNash neither trains against past versions of the agent, nor adds reward-shaping or expert data in the training algorithm, which is used to stabilize the learning in AlphaGo, OpenAI Five, AlphaStar, etc. Thus, this is a very different approach and may unlock further applications of RL methods in imperfect information tasks. Another recent work using a novel human regularized approach named distributional lambda piKL (DiL-piKL) [194] masters the complex No-Press Diplomacy, which is a complex strategy game with both cooperation and competition. While previous methods are mainly tailored to competitive games and thus are inefficient for No-Press Diplomacy, DiL-piKL regularizes a reward-maximizing policy towards a human imitation-learned policy. Theoretically, Dil-piKL is proven to have both the no-regret property in general games and last-iterate convergence in 2p0s games. Besides, there also exist studies [207, 208] that solve other complex card games.

Markov games (RTS and MOBA). There are also many successful applications in Markov games, including real-time strategy (RTS) games and multi-player online battle arena (MOBA) games. OpenAI Five [14] first releases its AI in 2018 and finally defeats the world champion team in 2019. Both OpenAI Five and AlphaStar [16] train the agent with self-play RL. OpenAI Five trains against a mixture of 20% past opponents and 80% the current policy, while AlphaStar samples opponents by pFSP that favors the most competitive or the closest opponents. Besides, AlphaStar also uses human demonstrations and a league exploiter that trains against the current agent. These self-play approaches are also applied in a popular Chinese mobile MOBA game, Honor of Kings, in the 1v1 mode [209], and the 5v5 full-game [210]. TiZero [211] combines the self-play mechanism in OpenAI Five and AlphaStar to train in the 11 vs 11 Google Research Football environment [212]. Concretely, the authors adopt pFSP to sample past opponents, while 80% of the time current agents play against the most recently saved agents. They also design a curriculum self-play mechanism, in which agents are trained on a sequence of progressively more difficult scenarios.

Game platforms. A variety of platforms are proposed for learning in games with no-regret learning, RL, and search. OpenSpiel [213] contains a collection of games, including NFG, EFG, and Markov games. OpenSpiel also provides many algorithms and evaluation approaches. RLCard [214] is mainly designed for RL in Poker games and provides some baselines. Google Research Football [212] is a platform for the multi-player Markov game, and it also contains small scenarios such as 1v1 and 1v3. Melting Pot [215] and PettingZoo [216] both provide a variety of competitive-cooperative games. Melting Pot focuses on evaluating generalization to novel social situations and uses RL to reduce the human labor required to create novel test scenarios. PettingZoo wraps previous multi-agent environments with a universal OpenAI Gym-like API, which is friendly for RL practitioners.

7 Closing remarks

With the power of current deep models and the success of learning in games, we believe the future directions may focus on the learning objective for real-world tasks and solve more realistic open-environment problems. For realistic scenarios, we think the following aspects are crucial for future research and applications.

Novel learning objectives. The main solution concepts of learning in games are still towards equilibrium, or equilibrium under specific games, e.g., the team-maximin equilibrium for multi-player mixed games [217]. There have been some alternatives to NE [171, 172], which may be studied further from other perspectives. Another goal of learning in games is to adapt to unseen opponents, either achieving zero-shot human-AI coordination [218, 219] or performing real-time solving like Libratus.

Learning in dynamic games. The game played may be time-varying rather than static in the real world, e.g., the financial trading market and e-commerce platforms. There have been some studies on the dynamic games setting [220–222].

Offline learning in games. For most current successful studies, both RL and learning in games require a perfect simulator, while for open-world learning problems, such simulators may be inaccessible. Offline reinforcement learning [223] attempts to learn from a batch of static data without interacting with the simulator during the training. A line of studies turned to offline learning in games [224–228], thus enlarging the applicability of learning in games.

Large pretrained models. A very recent trend is to apply large pretrained models based on Transformer [229] to multi-agent learning [230] or to play a broad class of games by a large model [231]. These large pretrained models have the potential to efficiently transfer knowledge across multiple tasks.

An intelligent game AI system should not only be unexploitable, but also can exploit opponents. Ideally, it should be endowed with an unexploitable policy, but can identify the opponent and adapt to it fast. Previous solutions for policy adaption rely on real-time solving techniques as used in Libratus, which requires massive computational resources. Large pretrained models on multiple tasks may alleviate this issue. If the AI system will cooperate with other unseen teammates to play against a team of opponents, it should be able to adapt to the teammate, too. The adaptation ability will save the time of retraining from scratch and improve the reliability in the open environment.

Recently, numerous studies have modeled their problems from the game-theoretic view and borrowed techniques from learning in games, let alone the applications of learning in games. We expect this review will serve as a catalyst for learning in games and its applications in more realistic tasks in the future.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2020AAA0107200) and National Natural Science Foundation of China (Grant No. 61921006).

References

1 Goldberg P W. A survey of PPAD-completeness for computing Nash equilibria. 2011. ArXiv:1103.2709
2 Daskalakis C, Goldberg P W, Papadimitriou C H. The complexity of computing a Nash equilibrium. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, 2006. 71–78
3 Chen X, Deng X. Settling the complexity of two-player Nash equilibrium. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), Berkeley, 2006. 261–272
4 Brown G W. Iterative solutions of games by fictitious play. In: Proceedings of Activity Analysis of Production and Allocation, 1951
5 Fudenberg D, Levine D K. The Theory of Learning in Games. Cambridge: MIT Press, 1998
6 Shoham Y, Powers R, Grenager T. If multi-agent learning is the answer, what is the question? Artif Intell, 2007, 171: 365–377
7 Cesa-Bianchi N, Lugosi G. Prediction, Learning, and Games. Cambridge: Cambridge University Press, 2006
8 Facchinei F, Kanzow C. Generalized Nash equilibrium problems. Ann Oper Res, 2010, 175: 177–211
9 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018
10 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533
11 Badia A P, Piot B, Kapturowski S, et al. Agent57: outperforming the Atari human benchmark. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 507–517
12 Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529: 484–489
13 Brown N, Sandholm T. Libratus: the superhuman AI for no-limit poker. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, 2017. 5226–5228
14 Berner C, Brockman G, Chan B, et al. Dota 2 with large scale deep reinforcement learning. 2019. ArXiv:1912.06680
15 Brown N, Sandholm T. Superhuman AI for multiplayer poker. Science, 2019, 365: 885–890
16 Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 2019, 575: 350–354
17 Li J, Koyamada S, Ye Q, et al. Suphx: mastering mahjong with deep reinforcement learning. 2020. ArXiv:2003.13590

18 Perolat J, de Vylder B, Hennes D, et al. Mastering the game of Stratego with model-free multiagent reinforcement learning. Science, 2022, 378: 990–996
19 Busoniu L, Babuska R, de Schutter B. A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybern C, 2008, 38: 156–172
20 Zhang K, Yang Z, Basar T. Multi-agent reinforcement learning: a selective overview of theories and algorithms. 2019. ArXiv:1911.10635
21 Yang Y, Wang J. An overview of multi-agent reinforcement learning from game theoretical perspective. 2020. ArXiv:2011.00583
22 Lu Y, Li W. Techniques and paradigms in modern game AI systems. Algorithms, 2022, 15: 282
23 Yin Q Y, Yang J, Huang K Q, et al. AI in human-computer gaming: techniques, challenges and opportunities. Mach Intell Res, 2023, 20: 299–317
24 Zinkevich M, Johanson M, Bowling M H, et al. Regret minimization in games with incomplete information. In: Proceedings of Advances in Neural Information Processing Systems 20, Vancouver, 2007. 1729–1736
25 Kalai A, Vempala S. Efficient algorithms for online decision problems. J Comput Syst Sci, 2005, 71: 291–307
26 Cesa-Bianchi N, Mansour Y, Stoltz G. Improved second-order bounds for prediction with expert advice. Mach Learn, 2007, 66: 321–352
27 Syrgkanis V, Agarwal A, Luo H, et al. Fast convergence of regularized learning in games. In: Proceedings of Advances in Neural Information Processing Systems 28, Montreal, 2015. 2989–2997
28 Daskalakis C, Panageas I. Last-iterate convergence: zero-sum games and constrained min-max optimization. In: Proceedings of the 10th Innovations in Theoretical Computer Science Conference, San Diego, 2019
29 Lanctot M, Waugh K, Zinkevich M, et al. Monte Carlo sampling for regret minimization in extensive games. In: Proceedings of Advances in Neural Information Processing Systems 22, Vancouver, 2009. 1078–1086
30 Tammelin O, Burch N, Johanson M, et al. Solving heads-up limit Texas Hold'em. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, 2015. 645–652
31 Littman M L. Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th International Conference on Machine Learning, Rutgers University, New Brunswick, 1994. 157–163
32 Littman M L. Friend-or-foe Q-learning in general-sum games. In: Proceedings of the 18th International Conference on Machine Learning, 2001. 322–328
33 Hu J, Wellman M P. Nash Q-learning for general-sum stochastic games. J Machine Learning Res, 2003, 4: 1039–1069
34 Heinrich J, Lanctot M, Silver D. Fictitious self-play in extensive-form games. In: Proceedings of the 32nd International Conference on Machine Learning, Lille, 2015. 805–813
35 Heinrich J, Silver D. Deep reinforcement learning from self-play in imperfect-information games. 2016. ArXiv:1603.01121
36 Srinivasan S, Lanctot M, Zambaldi V F, et al. Actor-critic policy optimization in partially observable multiagent environments. In: Proceedings of Advances in Neural Information Processing Systems 31, Montréal, 2018
37 Hennes D, Morrill D, Omidshafiei S, et al. Neural replicator dynamics: multiagent learning via hedging policy gradients. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, Auckland, 2020
38 Lockhart E, Lanctot M, Pérolat J, et al. Computing approximate equilibria in sequential adversarial games by exploitability descent. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 464–470
39 Pérolat J, Munos R, Lespiau J, et al. From poincaré recurrence to convergence in imperfect information games: finding equilibrium via regularization. In: Proceedings of the 38th International Conference on Machine Learning, 2021
40 He H, Boyd-Graber J L. Opponent modeling in deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, New York City, 2016. 1804–1813
41 Foerster J N, Chen R Y, Al-Shedivat M, et al. Learning with opponent-learning awareness. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, 2018. 122–130
42 Willi T, Letcher A, Treutlein J, et al. COLA: consistent learning with opponent-learning awareness. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 23804–23831
43 Zhao S, Lu C, Grosse R B, et al. Proximal learning with opponent-learning awareness. In: Proceedings of Advances in Neural Information Processing Systems, 2022
44 Lanctot M, Zambaldi V F, Gruslys A, et al. A unified game-theoretic approach to multiagent reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 4190–4203
45 Balduzzi D, Garnelo M, Bachrach Y, et al. Open-ended learning in symmetric zero-sum games. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019. 434–443
46 McAleer S, Lanier J B, Fox R, et al. Pipeline PSRO: a scalable approach for finding approximate Nash equilibria in large games. In: Proceedings of Advances in Neural Information Processing Systems, 2020
47 Muller P, Omidshafiei S, Rowland M, et al. A generalized training approach for multiagent learning. In: Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, 2020
48 Parker-Holder J, Pacchiano A, Choromanski K M, et al. Effective diversity in population based reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2020
49 Nieves N P, Yang Y, Slumbers O, et al. Modelling behavioural diversity for learning in open-ended games. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 8514–8524
50 Liu X, Jia H, Wen Y, et al. Towards unifying behavioral and response diversity for open-ended learning in zero-sum games. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 941–952
51 Liu Z, Yu C, Yang Y, et al. A unified diversity measure for multiagent reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2022
52 Shoham Y, Leyton-Brown K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge: Cambridge University Press, 2009
53 Watkins C J C H, Dayan P. Q-learning. Mach Learn, 1992, 8: 279–292
54 Sutton R S, McAllester D A, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation. In: Proceedings of Advances in Neural Information Processing Systems, 1999. 1057–1063
55 Hart S, Mas-Colell A. A simple adaptive procedure leading to correlated equilibrium. Econometrica, 2000, 68: 1127–1150
56 Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci, 1997, 55: 119–139
57 Nesterov Y. Excessive gap technique in nonsmooth convex minimization. SIAM J Optim, 2005, 16: 235–249
58 Kroer C, Farina G, Sandholm T. Solving large sequential games with the excessive gap technique. In: Proceedings of

Advances in Neural Information Processing Systems, 2018. 872–882
59 Hannan J. Approximation to Bayes risk in repeated play. In: Contributions to the Theory of Games. Princeton: Princeton University Press, 1957. 3: 97–139
60 Blackwell D. An analog of the minimax theorem for vector payoffs. Pac J Math, 1956, 6: 1–8
61 Abernethy J D, Bartlett P L, Hazan E. Blackwell approachability and no-regret learning are equivalent. In: Proceedings of the 24th Annual Conference on Learning Theory, Budapest, 2011. 27–46
62 Hart S, Mas-Colell A. Uncoupled dynamics do not lead to Nash equilibrium. Am Economic Rev, 2003, 93: 1830–1836
63 Cesa-Bianchi N, Freund Y, Haussler D, et al. How to use expert advice. J ACM, 1997, 44: 427–485
64 Hazan E. Introduction to online convex optimization. FNT Optimization, 2016, 2: 157–325
65 Shalev-Shwartz S. Online learning and online convex optimization. FNT Machine Learn, 2012, 4: 107–194
66 Waugh K, Bagnell J A. A unified view of large-scale zero-sum equilibrium computation. In: Proceedings of Computer Poker and Imperfect Information, 2015
67 Daskalakis C, Deckelbaum A, Kim A. Near-optimal no-regret algorithms for zero-sum games. In: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, 2011. 235–254
68 Rakhlin A, Sridharan K, Tewari A. Online learning: stochastic, constrained, and smoothed adversaries. In: Proceedings of Advances in Neural Information Processing Systems, 2011. 1764–1772
69 Rakhlin A, Sridharan K. Online learning with predictable sequences. In: Proceedings of the 26th Annual Conference on Learning Theory, 2013
70 Rakhlin A, Sridharan K. Optimization, learning, and games with predictable sequences. In: Proceedings of Advances in Neural Information Processing Systems, 2013. 3066–3074
71 Hazan E, Kale S. Extracting certainty from uncertainty: regret bounded by variation in costs. In: Proceedings of the 21st Annual Conference on Learning Theory, 2008. 57–68
72 Hsieh Y, Antonakopoulos K, Mertikopoulos P. Adaptive learning in continuous games: optimal regret bounds and convergence to Nash equilibrium. In: Proceedings of Conference on Learning Theory, Boulder, 2021. 2388–2422
73 Foster D J, Li Z, Lykouris T, et al. Learning in games: robustness of fast convergence. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 4727–4735
74 Abernethy J D, Lai K A, Levy K Y, et al. Faster rates for convex-concave games. In: Proceedings of Conference on Learning Theory, 2018. 1595–1625
75 Chen X, Peng B. Hedging in games: faster convergence of external and swap regrets. In: Proceedings of Advances in Neural Information Processing Systems, 2020
76 Daskalakis C, Fishelson M, Golowich N. Near-optimal no-regret learning in general games. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 27604–27616
77 Farina G, Lee C, Luo H, et al. Kernelized multiplicative weights for 0/1-polyhedral games: bridging the gap between learning in extensive-form and normal-form games. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 6337–6357
78 Anagnostides I, Daskalakis C, Farina G, et al. Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, 2022. 736–749
79 Farina G, Anagnostides I, Luo H, et al. Near-optimal no-regret learning dynamics for general convex games. In: Proceedings of Advances in Neural Information Processing Systems, 2022
80 Daskalakis C, Golowich N. Fast rates for nonparametric online learning: from realizability to learning in games. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, 2022. 846–859
81 Abe K, Sakamoto M, Iwasaki A. Mutation-driven follow the regularized leader for last-iterate convergence in zero-sum games. In: Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence, Eindhoven, 2022. 1–10
82 Anagnostides I, Farina G, Kroer C, et al. Uncoupled learning dynamics with $O(\log T)$ swap regret in multiplayer games. In: Proceedings of Advances in Neural Information Processing Systems, 2022
83 Piliouras G, Sim R, Skoulakis S. Beyond time-average convergence: near-optimal uncoupled online learning via clairvoyant multiplicative weights update. In: Proceedings of Advances in Neural Information Processing Systems, 2022
84 Farina G, Kroer C, Lee C W, et al. Clairvoyant regret minimization: equivalence with Nemirovski's conceptual prox method and extension to general convex games. In: Proceedings of Optimization for Machine Learning, 2022
85 Johanson M, Bard N, Burch N, et al. Finding optimal abstract strategies in extensive-form games. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, 2012
86 Golowich N, Pattathil S, Daskalakis C. Tight last-iterate convergence rates for no-regret learning in multi-player games. In: Proceedings of Advances in Neural Information Processing Systems, 2020
87 Wei C, Lee C, Zhang M, et al. Linear last-iterate convergence in constrained saddle-point optimization. In: Proceedings of the 9th International Conference on Learning Representations, 2021
88 Lei Q, Nagarajan S G, Panageas I, et al. Last iterate convergence in no-regret learning: constrained min-max optimization for convex-concave landscapes. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, 2021. 1441–1449
89 Azizian W, Iutzeler F, Malick J, et al. The last-iterate convergence rate of optimistic mirror descent in stochastic variational inequalities. In: Proceedings of Conference on Learning Theory, Boulder, 2021. 326–358
90 Cen S, Wei Y, Chi Y. Fast policy extragradient methods for competitive games with entropy regularization. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 27952–27964
91 Lee C, Kroer C, Luo H. Last-iterate convergence in extensive-form games. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 14293–14305
92 Cai Y, Oikonomou A, Zheng W. Finite-time last-iterate convergence for learning in multi-player games. In: Proceedings of Advances in Neural Information Processing Systems 35, 2022
93 Cen S, Chi Y, Du S S, et al. Faster last-iterate convergence of policy optimization in zero-sum Markov games. In: Proceedings of International Conference on Learning Representations, 2023
94 McKelvey R D, Palfrey T R. Quantal response equilibria for normal form games. Games Economic Behav, 1995, 10: 6–38
95 Daskalakis C, Frongillo R M, Papadimitriou C H, et al. On learning algorithms for Nash equilibria. In: Proceedings of the 3rd International Symposium on Algorithmic Game Theory, Athens, 2010. 114–125
96 Balcan M F, Constantin F, Mehta R. The weighted majority algorithm does not converge in nearly zero-sum games. In: Proceedings of International Conference on Machine Learning Workshop on Markets, Mechanisms, and Multi-Agent Models, Edinburgh, 2012

97 Papadimitriou C H, Piliouras G. From Nash equilibria to chain recurrent sets: solution concepts and topology. In: Proceedings of the ACM Conference on Innovations in Theoretical Computer Science, Cambridge, 2016. 227–235
98 Mertikopoulos P, Papadimitriou C H, Piliouras G. Cycles in adversarial regularized learning. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, 2018. 2703–2717
99 Bailey J P, Piliouras G. Multiplicative weights update in zero-sum games. In: Proceedings of the ACM Conference on Economics and Computation, 2018. 321–338
100 Bailey J P, Piliouras G. Fast and furious learning in zero-sum games: vanishing regret with non-vanishing step sizes. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 12977–12987
101 Cheung Y K, Piliouras G. Chaos, extremism and optimism: volume analysis of learning in games. In: Proceedings of Advances in Neural Information Processing Systems, 2020
102 Vlatakis-Gkaragkounis E, Flokas L, Lianeas T, et al. No-regret learning and mixed Nash equilibria: they do not mix. In: Proceedings of Advances in Neural Information Processing Systems, 2020
103 Kuhn H W. Extensive games and the problem of information. In: Proceedings of Contributions to the Theory of Games, 1953
104 Gibson R G, Lanctot M, Burch N, et al. Generalized sampling and variance in counterfactual regret minimization. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, 2012
105 Johanson M, Bard N, Lanctot M, et al. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In: Proceedings of International Conference on Autonomous Agents and Multiagent Systems, Valencia, 2012. 837–846
106 Tammelin O. Solving large imperfect information games using CFR+. 2014. ArXiv:1407.5042
107 Farina G, Kroer C, Sandholm T. Faster game solving via predictive Blackwell approachability: connecting regret matching and mirror descent. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021
108 Brown N, Sandholm T. Solving imperfect-information games via discounted regret minimization. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, 2019. 1829–1836
109 Brown N, Sandholm T. Regret-based pruning in extensive-form games. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 1972–1980
110 Brown N, Kroer C, Sandholm T. Dynamic thresholding and pruning for regret minimization. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017. 421–429
111 Brown N, Sandholm T. Reduced space and faster convergence in imperfect-information games via pruning. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 596–604
112 Burch N, Johanson M, Bowling M. Solving imperfect information games using decomposition. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, Québec City, 2014. 602–608
113 Ganzfried S, Sandholm T. Endgame solving in large imperfect-information games. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, Istanbul, 2015. 37–45
114 Brown N, Sandholm T. Safe and nested subgame solving for imperfect-information games. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 689–699
115 Ganzfried S, Sandholm T. Action translation in extensive-form games with large action spaces: axioms, paradoxes, and the pseudo-harmonic mapping. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, 2013. 120–128
116 Brown N, Sandholm T. Baby tartanian8: winning agent from the 2016 annual computer poker competition. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, 2016. 4238–4239
117 Waugh K, Morrill D, Bagnell J A, et al. Solving games with functional regret estimation. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, 2015. 2138–2145
118 Brown N, Lerer A, Gross S, et al. Deep counterfactual regret minimization. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, 2019. 793–802
119 Sen S, Sekaran M, Hale J. Learning to coordinate without sharing information. In: Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, 1994. 426–431
120 Greenwald A, Hall K. Correlated Q-learning. In: Proceedings of the 20th International Conference on Machine Learning, Washington, 2003. 242–249
121 Oliehoek F A, Amato C. A Concise Introduction to Decentralized POMDPs. Cham: Springer, 2016
122 Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, Stockholm, 2018. 2085–2087
123 Foerster J N, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, the 30th Innovative Applications of Artificial Intelligence, and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, 2018. 2974–2982
124 Rashid T, Samvelyan M, de Witt C S, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, 2018. 4292–4301
125 Wang J, Ren Z, Liu T, et al. QPLEX: duplex dueling multi-agent Q-learning. In: Proceedings of the 9th International Conference on Learning Representations, Austria, 2021
126 Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 6379–6390
127 Bowling M, Veloso M. Multiagent learning using a variable learning rate. Artif Intell, 2002, 136: 215–250
128 Leslie D S, Collins E J. Individual Q-learning in normal form games. SIAM J Control Optim, 2005, 44: 495–514
129 Zhang C, Lesser V R. Multi-agent learning with policy prediction. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010
130 Arslan G, Yuksel S. Decentralized Q-learning for stochastic teams and games. IEEE Trans Automat Contr, 2017, 62: 1545–1558
131 Leslie D S, Collins E J. Generalised weakened fictitious play. Games Economic Behav, 2006, 56: 285–298
132 Vitter J S. Random sampling with a reservoir. ACM Trans Math Softw, 1985, 11: 37–57
133 Pérolat J, Piot B, Pietquin O. Actor-critic fictitious play in simultaneous move multistage games. In: Proceedings of International Conference on Artificial Intelligence and Statistics, 2018. 919–928
134 Kawamura K, Tsuruoka Y. Neural fictitious self-play on ELF Mini-RTS. 2019. ArXiv:1902.02004
135 Hofbauer J, Sigmund K. Evolutionary Games and Population Dynamics. Cambridge: Cambridge University Press, 1998

136 Daskalakis C, Foster D J, Golowich N. Independent policy gradient methods for competitive reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2020
137 Raileanu R, Denton E, Szlam A, et al. Modeling others using oneself in multi-agent reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, 2018. 4254–4263
138 Zheng Y, Meng Z, Hao J, et al. A deep Bayesian policy reuse approach against non-stationary agents. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 962–972
139 Han Y, Gmytrasiewicz P J. Learning others' intentional models in multi-agent settings using interactive POMDPs. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 5639–5647
140 Costa-Gomes M A, Crawford V P. Cognition and behavior in two-person guessing games: an experimental study. Am Economic Rev, 2006, 96: 1737–1768
141 Simon H A. Bounded rationality. In: Utility and Probability. London: Palgrave Macmillan, 1990. 15–18
142 Wen Y, Yang Y, Luo R, et al. Probabilistic recursive reasoning for multi-agent reinforcement learning. In: Proceedings of the 7th International Conference on Learning Representations, New Orleans, 2019
143 Wen Y, Yang Y, Wang J. Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020. 414–421
144 Ma X, Isele D, Gupta J K, et al. Recursive reasoning graph for multi-agent reinforcement learning. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. 7664–7671
145 Foerster J N, Farquhar G, Al-Shedivat M, et al. DiCE: the infinitely differentiable Monte Carlo estimator. In: Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, 2018. 1524–1533
146 Bai Y, Jin C. Provable self-play algorithms for competitive reinforcement learning. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 551–560
147 Bai Y, Jin C, Yu T. Near-optimal reinforcement learning with self-play. In: Proceedings of Advances in Neural Information Processing Systems, 2020
148 Liu Q, Yu T, Bai Y, et al. A sharp analysis of model-based reinforcement learning with self-play. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 7001–7010
149 Mao W, Yang L, Zhang K, et al. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 15007–15049
150 Daskalakis C, Golowich N, Zhang K. The complexity of Markov equilibrium in stochastic games. 2022. ArXiv:2204.03991
151 Sayin M O, Zhang K, Leslie D S, et al. Decentralized Q-learning in zero-sum Markov games. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 18320–18334
152 Song Z, Mei S, Bai Y. When can we learn general-sum Markov games with a large number of players sample-efficiently? In: Proceedings of the 10th International Conference on Learning Representations, 2022
153 Ding D, Wei C, Zhang K, et al. Independent policy gradient for large-scale Markov potential games: sharper rates, function approximation, and game-agnostic convergence. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 5166–5220
154 Yang Y, Ma C. $O(T^{-1})$ convergence of optimistic-follow-the-regularized-leader in two-player zero-sum markov games. In: Proceedings of the 11th International Conference on Learning Representations, 2023
155 Xie Q, Chen Y, Wang Z, et al. Learning zero-sum simultaneous-move Markov games using function approximation and correlated equilibrium. In: Proceedings of Conference on Learning Theory, 2020. 3674–3682
156 Huang B, Lee J D, Wang Z, et al. Towards general function approximation in zero-sum Markov games. In: Proceedings of the 10th International Conference on Learning Representations, 2022
157 Jin C, Liu Q, Yu T. The power of exploiter: provable multi-agent RL in large state spaces. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 10251–10279
158 Cui Q, Zhang K, Du S S. Breaking the curse of multiagents in a large state space: RL in Markov games with independent linear function approximation. 2023. ArXiv:2302.03673
159 Son K, Kim D, Kang W J, et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019. 5887–5896
160 Yu C, Velu A, Vinitsky E, et al. The surprising effectiveness of PPO in cooperative multi-agent games. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022. 24611–24624
161 Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. In: Proceedings of the 4th International Conference on Learning Representations, San Juan, 2016
162 Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017. ArXiv:1707.06347
163 Kuba J G, Chen R, Wen M, et al. Trust region policy optimisation in multi-agent reinforcement learning. In: Proceedings of the 10th International Conference on Learning Representations, 2022
164 Wen M, Kuba J G, Lin R, et al. Multi-agent reinforcement learning is a sequence modeling problem. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022. 16509–16521
165 Wellman M P. Methods for empirical game-theoretic analysis. In: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference, Boston, 2006. 1552–1556
166 McMahan H B, Gordon G J, Blum A. Planning in the presence of cost functions controlled by an adversary. In: Proceedings of the 20th International Conference on Machine Learning, Washington, 2003. 536–543
167 McAleer S, Lanier J B, Wang K A, et al. XDO: a double oracle algorithm for extensive-form games. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 23128–23139
168 McAleer S, Wang K, Lanier J B, et al. Anytime PSRO for two-player zero-sum games. 2022. ArXiv:2201.07700
169 Zhou M, Chen J, Wen Y, et al. Efficient policy space response oracles. 2022. ArXiv:2202.00633
170 Elo A E. The Rating of Chess Players, Past and Present. New York: Arco Pub., 1978
171 Balduzzi D, Tuyls K, Pérolat J, et al. Re-evaluating evaluation. In: Proceedings of Advances in Neural Information Processing Systems, Montréal, 2018. 3272–3283
172 Omidshafiei S, Papadimitriou C, Piliouras G, et al. α-rank: multi-agent evaluation by evolution. Sci Rep, 2019, 9: 9937
173 Rowland M, Omidshafiei S, Tuyls K, et al. Multiagent evaluation under incomplete information. In: Proceedings of Advances in Neural Information Processing Systems, Vancouver, 2019. 12270–12282
174 Rashid T, Zhang C, Ciosek K. Estimating α-rank by maximizing information gain. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021. 5673–5681
175 Yan X, Du Y, Ru B, et al. Learning to identify top Elo ratings: a dueling bandits approach. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. 8797–8805

176 Smith M O, Anthony T, Wellman M P. Iterative empirical game solving via single policy best response. In: Proceedings of the 9th International Conference on Learning Representations, 2021
177 Liu S, Marris L, Hennes D, et al. NeuPL: neural population learning. In: Proceedings of the 10th International Conference on Learning Representations, 2022
178 Liu S, Lanctot M, Marris L, et al. Simplex neural population learning: any-mixture Bayes-optimality in symmetric zero-sum games. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 13793–13806
179 Cohen A, Qiao X, Yu L, et al. Diverse exploration via conjugate policies for policy gradient methods. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, 2019. 3404–3411
180 Masood M A, Doshi-Velez F. Diversity-inducing policy gradient: using maximum mean discrepancy to find a set of diverse policies. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 5923–5929
181 Kulesza A, Taskar B. Determinantal point processes for machine learning. FNT Machine Learn, 2012, 5: 123–286
182 Fu H, Liu W, Wu S, et al. Actor-critic policy optimization in a large-scale imperfect-information game. In: Proceedings of the 10th International Conference on Learning Representations, 2022
183 Czarnecki W M, Gidel G, Tracey B D, et al. Real world games look like spinning tops. In: Proceedings of Advances in Neural Information Processing Systems, 2020
184 Dennis M, Jaques N, Vinitsky E, et al. Emergent complexity and zero-shot transfer via unsupervised environment design. In: Proceedings of Advances in Neural Information Processing Systems, 2020
185 Gur I, Jaques N, Miao Y, et al. Environment generation for zero-shot compositional reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 4157–4169
186 Samvelyan M, Khan A, Dennis M D, et al. MAESTRO: open-ended environment design for multi-agent reinforcement learning. In: Proceedings of the 11th International Conference on Learning Representations, 2023
187 Gruslys A, Lanctot M, Munos R, et al. The advantage regret-matching actor-critic. 2020. ArXiv:2008.12234
188 Steinberger E, Lerer A, Brown N. DREAM: deep regret minimization with advantage baselines and model-free learning. 2020. ArXiv:2006.10410
189 Li H, Wang X, Jia F, et al. RLCFR: minimize counterfactual regret by deep reinforcement learning. Expert Syst Appl, 2022, 187: 115953
190 Jin P H, Keutzer K, Levine S. Regret minimization for partially observable deep reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, 2018
191 Tomar M, Shani L, Efroni Y, et al. Mirror descent policy optimization. In: Proceedings of the 10th International Conference on Learning Representations, 2022
192 Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, 2018
193 Sokota S, D'Orazio R, Kolter J Z, et al. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In: Proceedings of International Conference on Learning Representations, 2023
194 Bakhtin A, Wu D J, Lerer A, et al. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. In: Proceedings of the 11th International Conference on Learning Representations, 2023
195 Qin R, Luo F, Qian H, et al. Unified policy optimization for continuous-action reinforcement learning in non-stationary tasks and games. 2022. ArXiv:2208.09452
196 Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. Nature, 2017, 550: 354–359
197 Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science, 2018, 362: 1140–1144
198 Schrittwieser J, Antonoglou I, Hubert T, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. Nature, 2020, 588: 604–609
199 Jiang Q, Li K, Du B, et al. DeltaDou: expert-level DouDizhu AI through self-play. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 1265–1271
200 Brown N, Bakhtin A, Lerer A, et al. Combining deep reinforcement learning and search for imperfect-information games. In: Proceedings of Advances in Neural Information Processing Systems, 2020
201 Moravčík M, Schmid M, Burch N, et al. DeepStack: expert-level artificial intelligence in heads-up no-limit poker. Science, 2017, 356: 508–513
202 Brown N, Sandholm T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. Science, 2018, 359: 418–424
203 Brown N, Sandholm T, Amos B. Depth-limited solving for imperfect-information games. In: Proceedings of Advances in Neural Information Processing Systems, Montréal, 2018. 7674–7685
204 Zha D, Xie J, Ma W, et al. DouZero: mastering DouDizhu with self-play deep reinforcement learning. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 12333–12344
205 Zhao Y, Zhao J, Hu X, et al. DouZero+: improving DouDizhu AI by opponent modeling and coach-guided learning. In: Proceedings of IEEE Conference on Games, Beijing, 2022. 127–134
206 Yang G, Liu M, Hong W, et al. PerfectDou: dominating DouDizhu with perfect information distillation. In: Proceedings of Advances in Neural Information Processing Systems, 2022
207 Liu T, Zheng Z, Li H, et al. Playing card-based RTS games with deep reinforcement learning. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 4540–4546
208 Serrino J, Kleiman-Weiner M, Parkes D C, et al. Finding friend and foe in multi-agent games. In: Proceedings of Advances in Neural Information Processing Systems, Vancouver, 2019. 1249–1259
209 Ye D, Liu Z, Sun M, et al. Mastering complex control in MOBA games with deep reinforcement learning. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 6672–6679
210 Ye D, Chen G, Zhang W, et al. Towards playing full MOBA games with deep reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2020
211 Lin F, Huang S, Pearce T, et al. TiZero: mastering multi-agent football with curriculum learning and self-play. 2023. ArXiv:2302.07515
212 Kurach K, Raichuk A, Stanczyk P, et al. Google research football: a novel reinforcement learning environment. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 4501–4510
213 Lanctot M, Lockhart E, Lespiau J, et al. OpenSpiel: a framework for reinforcement learning in games. 2019. ArXiv:1908.09453

214 Zha D, Lai K, Huang S, et al. RLCard: a platform for reinforcement learning in card games. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020. 5264–5266
215 Leibo J Z, Duéñez-Guzmán E A, Vezhnevets A, et al. Scalable evaluation of multi-agent reinforcement learning with melting pot. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 6187–6199
216 Terry J K, Black B, Grammel N, et al. PettingZoo: Gym for multi-agent reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 15032–15043
217 Zhang Y, An B, Subrahmanian V S. Correlation-based algorithm for team-maxmin equilibrium in multiplayer extensive-form games. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022. 606–612
218 Strouse D, McKee K R, Botvinick M M, et al. Collaborating with humans without human data. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 14502–14515
219 Cui B, Hu H, Lupu A, et al. Off-team learning. In: Proceedings of Advances in Neural Information Processing Systems, 2022
220 Zhang M, Zhao P, Luo H, et al. No-regret learning in time-varying zero-sum games. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 26772–26808
221 Harris K, Anagnostides I, Farina G, et al. Meta-learning in games. In: Proceedings of the 11th International Conference on Learning Representations, 2023
222 Anagnostides I, Panageas I, Farina G, et al. On the convergence of no-regret learning dynamics in time-varying games. 2023. ArXiv:2301.11241
223 Levine S, Kumar A, Tucker G, et al. Offline reinforcement learning: tutorial, review, and perspectives on open problems. 2020. ArXiv:2005.01643
224 Cui Q, Du S S. When is offline two-player zero-sum Markov game solvable? In: Proceedings of Workshop on Gamification and Multiagent Solutions, 2022
225 Zhong H, Xiong W, Tan J, et al. Pessimistic minimax value iteration: provably efficient equilibrium learning from offline datasets. In: Proceedings of International Conference on Machine Learning, Baltimore, 2022. 27117–27142
226 Li S, Wang X, Cerný J, et al. Offline equilibrium finding. 2022. ArXiv:2207.05285
227 Zhang Y, Bai Y, Jiang N. Offline learning in Markov games with general function approximation. 2023. ArXiv:2302.02571
228 Zhang F, Jia C, Li Y C, et al. Discovering generalizable multi-agent coordination skills from multi-task offline data. In: Proceedings of the 11th International Conference on Learning Representations, 2023
229 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 5998–6008
230 Meng L, Wen M, Yang Y, et al. Offline pre-trained multi-agent decision transformer: one big sequence model tackles all SMAC tasks. 2021. ArXiv:2112.02845
231 Reed S, Zolna K, Parisotto E, et al. A generalist agent. 2022. ArXiv:2205.06175