Empirical Policy Optimization for *n*-Player Markov Games

Yuanheng Zhu, *Senior Member, IEEE*, Weifan Li, Mengchen Zhao, Jianye Hao, and Dongbin Zhao, *Fellow, IEEE*

***Abstract*—In single-agent Markov decision processes, an agent can optimize its policy based on the interaction with the environment. In multiplayer Markov games (MGs), however, the interaction is nonstationary due to the behaviors of other players, so the agent has no fixed optimization objective. The challenge becomes finding equilibrium policies for all players. In this research, we treat the evolution of player policies as a dynamical process and propose a novel learning scheme for Nash equilibrium. The core is to evolve one's policy according to not just its current in-game performance, but an aggregation of its performance over history. We show that for a variety of MGs, players in our learning scheme will provably converge to a point that is an approximation to Nash equilibrium. Combined with neural networks, we develop an *empirical policy optimization* algorithm, which is implemented in a reinforcement-learning framework and runs in a distributed way, with each player optimizing its policy based on own observations. We use two numerical examples to validate the convergence property on small-scale MGs, and a pong example to show the potential on large games.**

***Index Terms*—Continuous-time learning dynamics (CTLD), Markov game (MG), *n*-player, Nash equilibrium, policy optimization.**

I. Introduction

MARKOV games (MGs), or stochastic games called in [1] and [2], are the extension of Markov decision processes (MDPs) from a single-agent environment to multiplayer scenarios [3], [4]. Compared to normal-form games (NFGs) that are stateless and lack the transition of states, MG players encounter multiple decision moments in one round, and at each step have to take into account current game states to make decisions. Each player aims to maximize its sum of rewards over time horizon, instead of one-stage payoff in NFGs. Another famous type of sequential games is extensive-form games (EFGs) [5], [6], in which the moves of different players are played in orders, compared to MG players acting simultaneously. EFGs use a game tree to describe the game process, and reshaping MGs to EFGs results in exponential blowup in size with respect to horizon.

In game theory, an important concept is Nash equilibrium, at which no player has the intention of deviating its strategy without sacrificing the current payoff. However, even for simple NFGs, computing Nash equilibrium is proved to be PPAD-complete [7]. Alternatively, learning-based schemes provide a computational intelligence way to approach equilibria and have now become appealing to researchers [8], [9].

Deep reinforcement learning (DRL) is a powerful tool in sequential decision makings [10]–[12], and has also received attention from the game field. One biggest challenge of DRL in MGs is the nonstationarity of optimization objectives, since each player's payoff is determined by others' behaviors. Recent progress has been made on two-player zero-sum cases [13]–[15]. For more general *n*-player games with an arbitrary number of players, one solution is to convert MGs to empirical games (seen as NFGs) with every policy being a strategic option [16]. Then, DRL optimizes the policy of each player in a game environment against a group of fixed opponents, whose strategies are mixtures of empirical policies [17], [18]. However, this approach is computationally expensive because different players are trained in different game environments with specific opponents, and the synthesis of opponent strategies may require additional computational efforts.

In recent years, reinforcement learning (RL) has promoted the development of the payoff-based learning scheme in NFGs [19], [20]. Neglecting what the others' strategies are, each player updates its strategy based on the aggregation of its on-going payoffs. If all players follow the same update rule, the evolution of their strategies converges to a Nash distribution with theoretical guarantees. However, in the field of MGs, this scheme faces obstacles because the strategy becomes a policy mapping from states to action distributions and the payoff is the expectation of sum of future rewards. Motivated by that, in this article, we propose a continuous-time learning dynamics (CTLD) for arbitrary *n*-player MGs. Instead of

Manuscript received February 22, 2022; revised May 5, 2022; accepted May 29, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101005; in part by the National Natural Science Foundation of China under Grant 62136008; in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030400; and in part by the Youth Innovation Promotion Association CAS. This article was recommended by Associate Editor J. Zhang. *(Corresponding author: Dongbin Zhao.)*

Yuanheng Zhu, Weifan Li, and Dongbin Zhao are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yuanheng.zhu@ia.ac.cn; liweifan2018@ia.ac.cn; dongbin.zhao@ia.ac.cn).

Mengchen Zhao is with the Noah's Ark Laboratory, Huawei, Beijing 100085, China (e-mail: zhaomengchen@huawei.com).

Jianye Hao is with the Noah's Ark Laboratory, Huawei, Beijing 100085, China, and also with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: haojianye@huawei.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3179775.

Digital Object Identifier 10.1109/TCYB.2022.3179775

changing the form of games, all players in CTLD interact in the same game environment and each player evolves its policy based on the aggregation of its on-going performance. To facilitate large-scale applications, an empirical policy optimization (EPO) algorithm is developed. Player policies are represented by neural networks (NNs) and the parameters are trained based on the entire history of experience in an RL way. Compared to existing methods, our contributions are threefold.

1) The learning scheme runs in a totally distributed way and players require no other game information but only own observations. Players do not need to know how many players are participating and what strategies the others are playing, so the scheme is applicable to arbitrary n-player cases.
2) Based on the fixed-point theorem and Lyapunov stability of dynamical systems, we prove that CTLD converges to a Nash distribution (an approximation to Nash equilibrium with arbitrary precision) for a variety of MGs.
3) The simplicity and distributed property makes the learning scheme compatible with the RL framework for large-scale games. Experimental results show the efficiency of EPO in approaching Nash equilibrium.

The remainder of this article is organized as follows. Sections II and III give the related work and preliminary knowledge on MGs. Section IV presents our CTLD and establishes the convergence theorem. Section V extends the learning scheme to large-scale problems by introducing NNs and DRL techniques. Section VI conducts experiments to verify the effectiveness and Section VII draws the conclusion. Appendixes give the proofs of main theorems and illustrate details of algorithm implementation.

II. Related Work

Early research on MGs was mainly value-based methods that aimed to solve Nash values of Bellman-like equations [3]. If the game model is known, one can apply dynamic programming (e.g., [21]). Otherwise, one can learn the values based on online observations like Q-learning [22]. However, value-based methods rely on the Nash computing (of NFGs) at every state, and the optimization over joint-action space suffers from combinatorial explosion as the number of players increases. To reduce joint-action space, some research [23], [24] adopted the mean-field concept and modeled the interactions among agents by the interaction between an individual and a virtual agent averaged by others. It inevitably introduces approximation error and deviates the solution from Nash equilibrium.

Policy optimization, or policy update, is efficient in optimizing agent policies [11], [25]. However, extra efforts are needed to manipulate the update direction toward the Nash equilibrium, when applying policy update to multiplayer scenarios. As a special case of MGs, two-player zero-sum games have received much attention. Srinivasan *et al.* [26] showed that when directly applying independent policy update rules in zero-sum sequential games, the regret had no sublinearity in iterations, in other words, the process may not converge. Lockhart *et al.* [14] improved the results by optimizing one player's policy against its best response opponent, and proved when using counterfactual values, the joint policies converged to a Nash equilibrium in two-player EFGs. Daskalakis *et al.* [15] chose a two-timescale learning rates for the independent learning of min-player and max-player, which can be seen as a softened "gradient descent versus best response" scheme.

For more general n-player games, the development is limited. The recent progress is policy-search response oracles (PSROs), which was first proposed by Lanctot *et al.* [16]. The main idea is to reduce MGs to empirical games, or metagames, whose policy sets are composed of empirical policies in history. Each player finds the (approximate) best response to its opponents' metastrategies, and the new policies are added into policy sets for the next iteration. The advantage of PSRO is that it provides a unified framework for different choices of metasolvers. Balduzzi *et al.* [17] proposed to use rectified Nash mixtures to encourage policy diversity. Muller *et al.* [18] introduced the α-rank multiagent evaluation metric [27] in PSRO, and showed promising performance in computing equilibria. However, PSRO is intensive in computation from two aspects: 1) the policy update of each player is separated in different game environments with different opponents and 2) additional computational efforts are required by metasolvers and empirical payoff evaluation.

In the control field, the multiplayer games mostly consider deterministic policies over continuous actions. The problem becomes solving the Hamilton–Jacobi–Isaacs equations for two-player zero-sum games [28]–[30] and the Hamilton–Jacobi equations for games with more than two players [31]–[33]. Here, we focus on discrete action sets and study stochastic policies. The optimization of policies has to take into account the expectation over all possible trajectories. There is also considerable research of multiagent RL (MARL) on partially observable environments [34]–[36]. Unfortunately, the research pays more attention to multiagent cooperation and lacks the theoretical guarantee of Nash equilibrium.

III. Background and Terminology

An MG played by a finite set of players $\mathcal{N} = \{1, 2, \ldots, n\}$ can be described by $\mathcal{MG} = (\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \gamma, \rho_0)$, where $\mathcal{S}$ is the finite set of states, $\mathcal{A}^i$ is the finite set of actions for each player $i \in \mathcal{N}$, $\mathcal{R}^i : \mathcal{S} \times \{\mathcal{A}^i\} \to \mathbb{R}$ is player i's (bounded) reward function, $\mathcal{P} : \mathcal{S} \times \{\mathcal{A}^i\} \times \mathcal{S} \to [0, 1]$ is the state transition function, and $\gamma \in (0, 1)$ is the discounted factor; ρ_0 is the initial state distribution.

In the field of RL, one is interested in a *policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that describes the action selection probability at a given s by $\pi(\cdot|s) \in \Delta(|\mathcal{A}|)$. $\Delta(|\mathcal{A}|)$ denotes the simplex $\{p \in \mathbb{R}^{|\mathcal{A}|} \mid p \geq 0$ componentwise, and $\mathbf{1}^T p = 1\}$. Assuming each player has an independent $\pi^i : \mathcal{S} \times \mathcal{A}^i \to [0, 1]$, the aggregation forms the policy profile $\boldsymbol{\pi} = (\pi^i)_{i\in\mathcal{N}}$, and player i's expected return, or *value*, starting from s_0 is defined as $V^i_{\boldsymbol{\pi}}(s_0) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r^i_k \mid \boldsymbol{a}_k \sim \boldsymbol{\pi}(s_k), r^i_k \sim \mathcal{R}^i(s_k, \boldsymbol{a}_k), s_k \sim \mathcal{P}(s_k, \boldsymbol{a}_k)]$. Another important RL concept is the state–action value, or Q value: $Q^i_{\boldsymbol{\pi}}(s, a^i) = \mathbb{E}[r^i + \gamma V^i_{\boldsymbol{\pi}}(s') \mid \boldsymbol{a}^{-i} \sim$

$\boldsymbol{\pi}^{-i}(s)$, $r^i \sim \mathcal{R}^i(s, \boldsymbol{a}^i)$, $s' \sim \mathcal{P}(s, \boldsymbol{a})$]. We use $-i$ to indicate the other players in $\mathcal{N}$ except i. The difference between $Q^i_{\boldsymbol{\pi}}$ and $V^i_{\boldsymbol{\pi}}$ is known as the *advantage*: $A^i_{\boldsymbol{\pi}}(s, a^i) = Q^i_{\boldsymbol{\pi}}(s, a^i) - V^i_{\boldsymbol{\pi}}(s)$. In what follows, we sometimes use $A_{\pi^i, \boldsymbol{\pi}^{-i}}$ to denote the observed advantage of player i when it is playing π^i and the others are playing $\boldsymbol{\pi}^{-i}$.

Given the initial state distribution ρ_0, player i's *payoff* is the expected value under the profile $\boldsymbol{\pi}$: $u^i(\pi^i, \boldsymbol{\pi}^{-i}) = \sum_s \rho_0(s) V^i_{\boldsymbol{\pi}}(s)$, and each player aims to maximize its own payoff. Once the other policies $\boldsymbol{\pi}^{-i}$ are fixed, the game is reduced to player i's MDP, and the difference in performance between player i's any two policies π^i and $\pi^i_\dagger$ follows the policy update lemma in [25]. Before restating the lemma in a game setting, we let $\rho_{\boldsymbol{\pi}}(s) = (P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \dots)$ be the *discounted visitation frequencies*, where $s_0 \sim \rho_0$ and all players follow $\boldsymbol{\pi}$.

Lemma 1 (Restatement of Policy Update [25]): Given the other policy profile $\boldsymbol{\pi}^{-i}$, player i's payoffs under two policies π^i and $\pi^i_\dagger$ satisfy

$$u^i\big(\pi^i_\dagger, \boldsymbol{\pi}^{-i}\big) = u^i\big(\pi^i, \boldsymbol{\pi}^{-i}\big) + \sum_s \rho_{\pi^i_\dagger, \boldsymbol{\pi}^{-i}}(s) \sum_{a^i} \pi^i_\dagger\big(s, a^i\big) A_{\pi^i, \boldsymbol{\pi}^{-i}}\big(s, a^i\big).$$

Lemma 1 follows directly from the proof of [25, Appendix A]. Player i's *best response* to a given profile $\boldsymbol{\pi}^{-i}$ is the policy that maximizes its payoff: $\beta^i(\boldsymbol{\pi}^{-i}) = \arg\max_{\pi^i \in \Pi^i} u^i(\pi^i, \boldsymbol{\pi}^{-i})$, where Π^i represents the policy space of player i. If in a profile $\boldsymbol{\pi}_* = (\pi^i_*)_{i \in \mathcal{N}}$ each policy is the best response of the others, the profile is called the *Nash equilibrium* and satisfies $u^i(\pi^i_*, \boldsymbol{\pi}^{-i}_*) \geq u^i(\pi^i, \boldsymbol{\pi}^{-i}_*)$ $\forall \pi^i \in \Pi^i$ and $\forall i \in \mathcal{N}$.

For any profile in the joint policy space $\boldsymbol{\Pi} = (\times \Pi^i)_{i \in \mathcal{N}}$, NashConv provides a metric to measure the distance to Nash, that is, $\text{NashConv}(\boldsymbol{\pi}) = \sum_i \max_\pi u^i(\pi, \boldsymbol{\pi^{-i}}) - u^i(\pi^i, \boldsymbol{\pi^{-i}})$. It always has $\text{NashConv}(\boldsymbol{\pi}) \geq 0$ for any profile and is equal to 0 at the Nash equilibrium.

IV. Convergent Continuous-Time Learning Scheme

A. Continuous-Time Learning Dynamics

We now establish the learning scheme for the Nash equilibrium of n-player MGs. The outline is that each player keeps a score function that records its on-going performance, and then maps the score to a policy that is played with the others to evaluate performance. The process is modeled in continuous time, repeated with an infinitesimal time step between three stages described below. A block diagram of the dynamical system is given in Fig. 1.

1) Assessment Stage: Consider the current time t and all players' profile $\boldsymbol{\pi}_t = (\pi^i_t)_{i \in \mathcal{N}}$. Player i's *score* y^i_t keeps the running average of past weighted advantages $\rho_{\boldsymbol{\pi}_\tau}(s) A^i_{\boldsymbol{\pi}_\tau}(s, a^i)$, $\tau \in [0, t)$, at every state–action pair, based on the exponential discounting aggregation

$$y^i_t(s, a^i) = e^{-\eta t} y^i_0(s, a^i) + \eta \int_0^t e^{-\eta(t-\tau)} \rho_{\boldsymbol{\pi}_\tau}(s) A^i_{\boldsymbol{\pi}_\tau}\big(s, a^i\big) d\tau \quad \forall s \in \mathcal{S} \quad \forall a^i \in \mathcal{A}^i$$

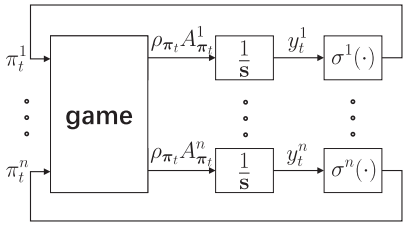


Fig. 1. Block diagram of CTLD. $(1/\mathbf{s})$ indicates the integrator block.

where $\eta > 0$ is the learning rate and y^i_0 is an arbitrary starting point. By formally defining an operator w^i for each player mapping from policy to weighted advantage: $[w^i(\boldsymbol{\pi})](s, a^i) = \rho_{\boldsymbol{\pi}}(s) A^i_{\boldsymbol{\pi}}(s, a^i)$, the evolution of score can be described in a differential form[1]

$$\dot{y}^i_t = \eta\big(w^i(\boldsymbol{\pi}_t) - y^i_t\big) \tag{1}$$

where the over-dot indicates time derivative.

2) Choice Stage: Once obtained the score, each player is able to map it to a policy by selecting the greedy action $\arg\max_{a^i} y^i_t(s, a^i)$ at every state. To ensure the map is continuous and single-valued, a smooth and strongly convex regularizer is used to yield the *choice map* σ^i from score to policy

$$\sigma^i \colon y^i \to \arg\max_{\pi^i \in \Pi^i} \left\{ \sum_{a^i} y^i(s, a^i) \pi^i(s, a^i) - h^i(\pi^i(\cdot|s)) \right\}_{s \in \mathcal{S}}. \tag{2}$$

Such h^i is also called the penalty or smoothing function in [9] and [37], and is assumed to satisfy the following properties.

Assumption 1 [19]: Let $\mathcal{C}$ be a compact convex subset of a finite-dimensional normed space, and $h \colon \mathcal{C} \to \mathbb{R}$ be a regularizer function on $\mathcal{C}$. The following properties hold for all $x, x' \in \mathcal{C}$ and all $\alpha \in [0, 1]$:

1) h is continuous;
2) h is strongly convex, that is, there exists $K > 0$ such that

$$h\big(\alpha x + (1-\alpha)x'\big) \leq \alpha h(x) + (1-\alpha) h(x') - \frac{1}{2} K \alpha (1-\alpha) \big\|x - x'\big\|^2.$$

Assumption 1 describes the characteristics of regularizer function $h^i(\pi^i(\cdot|s))$ at every state with $\mathcal{C}$ being $\Delta(|\mathcal{A}^i|)$. By abuse of notation, we let $h^i(\pi^i) = \sum_s h^i(\pi^i(\cdot|s))$, where $h^i(\pi^i(\cdot|s))$ satisfies Assumption 1 and is K^i-strongly convex.

There are a variety of forms of regularizers, such as the Tsallis entropy and Burg entropy [9], but a commonly used one in RL is the (negative) Gibbs entropy

$$h^i\big(\pi^i(\cdot|s)\big) = \epsilon \sum_{a^i} \pi^i\big(s, a^i\big) \log \pi^i\big(s, a^i\big) \tag{3}$$

which is continuously differentiable and ϵ-strongly convex with respect to L^1-norm. $\epsilon > 0$ is known as entropic parameter. A straightforward benefit with the entropic regularizer is

[1]One should distinguish between the two time indices t and k: the former indicates the evolution of score or policy in learning process, while the latter indicates the state transition in game process.

the closed-form expression of choice map, which is a softmax function

$$\left[\sigma^i\left(y^i\right)\right](s,a^i)=\frac{\exp\left(\frac{1}{\epsilon}y^i\left(s,a^i\right)\right)}{\sum_b \exp\left(\frac{1}{\epsilon}y^i(s,b)\right)}.$$

When $\epsilon \to 0$, the choice map tends to select the greedy action with the highest score at every state. When ϵ is arbitrarily large, the policy is like to be uniformly random.

3) Game Stage: With the mapped policy $\pi_t^i=\sigma^i(y_t^i)\ \forall i \in \mathcal{N}$, all players play in the game and observe $\rho_{\boldsymbol{\pi}_t}$ and $A^i_{\boldsymbol{\pi}_t}$ at every state and action. Thus, the learning system in (1) operates continuously. For finite MGs, if the game model (reward and transition functions) is known, the exact solutions of $\rho_{\boldsymbol{\pi}_t}$ and $A^i_{\boldsymbol{\pi}_t}$ at given $\boldsymbol{\pi}_t$ can be analytically calculated by linear algebra (as shown in Appendix A).

B. Convergence to Nash Distribution

With all players following the scheme as per above, the CTLD of the entire system can be written in a stacked form

$$\begin{cases}\dot{\boldsymbol{y}}_t=\eta\left(\boldsymbol{w}(\boldsymbol{\pi}_t)-\boldsymbol{y}_t\right)\\ \boldsymbol{\pi}_t=\boldsymbol{\sigma}(\boldsymbol{y}_t)\end{cases}\qquad\text{(CTLD)}$$

where $\boldsymbol{y}_t=(y_t^i)_{i\in\mathcal{N}}$, $\boldsymbol{w}(\boldsymbol{\pi}_t)=(w^i(\boldsymbol{\pi}_t))_{i\in\mathcal{N}}$, and $\boldsymbol{\sigma}(\boldsymbol{y}_t)=(\sigma^i(y_t^i))_{i\in\mathcal{N}}$. Bounded reward and softmax choice map make $\boldsymbol{w}\circ\boldsymbol{\sigma}$ a continuous and bounded function. Hence, the existence of a fixed point of (CTLD) is guaranteed by Brouwer's fixed-point theorem [38]. Denote $\bar{\boldsymbol{y}}=(\bar{y}_i)_{i\in\mathcal{N}}$ as the fixed point satisfying $\bar{\boldsymbol{y}}=\boldsymbol{w}\circ\boldsymbol{\sigma}(\bar{\boldsymbol{y}})$, and let $\bar{\boldsymbol{\pi}}=(\bar{\pi}_i)_{i\in\mathcal{N}}$ be the induced policy profile with $\bar{\boldsymbol{\pi}}=\boldsymbol{\sigma}(\bar{\boldsymbol{y}})$.

Theorem 1:

1) If $\boldsymbol{\pi}_*=(\pi_*^i)_{i\in\mathcal{N}}$ is a Nash equilibrium to the MG with *regularized* payoff, that is, $U^i(\pi^i,\boldsymbol{\pi}^{-i})=u^i(\pi^i,\boldsymbol{\pi}^{-i})-h^i(\pi^i)$ and $U^i(\pi_*^i,\boldsymbol{\pi}_*^{-i})\geq U^i(\pi^i,\boldsymbol{\pi}_*^{-i})\ \forall\pi^i\in\Pi^i, i\in\mathcal{N}$, then $\boldsymbol{y}_*=\boldsymbol{w}(\boldsymbol{\pi}_*)$ is the fixed point of (CTLD).
2) The converse is true if each player's original payoff u^i is individually concave in the sense that $u^i(\pi^i,\boldsymbol{\pi}^{-i})$ is concave in π^i for all $\boldsymbol{\pi}^{-i}\in\boldsymbol{\Pi}^{-i}\ \forall i\in\mathcal{N}$.

The proof is presented in Appendix B. Note that in the theorem, the equilibrium is modified to take into account the influence of regularizer. It is sometimes referred to as *Nash distribution* [8], [20] to distinguish from the Nash equilibrium with original payoffs. If the regularizer in (2) is sufficiently close to 0, the Nash distribution coincides with the Nash equilibrium. One condition for the global equivalence between fixed-point policy and Nash distribution is the individual concavity of game payoffs. In many scenarios [39], [40], a local Nash is sometimes easier to use than an expensive global solution. The following corollary extends the second part of Theorem 1 to local cases by restricting the interested domain to a neighbor of the fixed point. Its proof is therefore omitted to avoid unnecessary duplication.

Corollary 1: Let $\bar{\boldsymbol{y}}$ be the fixed point of $\boldsymbol{w}\circ\boldsymbol{\sigma}$. If u^i is locally individually concave around $\bar{\boldsymbol{\pi}}=\boldsymbol{\sigma}(\bar{\boldsymbol{y}})$ for all players, then $\bar{\boldsymbol{\pi}}$ is a *local* Nash distribution.

Now, we analyze the convergence property of (CTLD) based on the Lyapunov stability theory of dynamical systems [41]. Consider the Fenchel-coupling function [19] and by summing over all states, define

$$F^i\left(\pi^i,y^i\right)=\max_{\pi\in\Pi^i}\sum_s\left(\sum_{a^i}y^i\left(s,a^i\right)\pi\left(s,a^i\right)-h^i(\pi(\cdot|s))\right)-\sum_s\left(\sum_{a^i}y^i\left(s,a^i\right)\pi^i\left(s,a^i\right)-h^i\left(\pi^i(\cdot|s)\right)\right)$$

for any (π^i,y^i) pair. Naturally, $F^i(\pi^i,y^i)\geq 0$. By staying at the fixed-point policy $\bar{\boldsymbol{\pi}}$, we can take $\sum_i F^i(\bar{\pi}^i,y_t^i)$ as the Lyapunov function and calculate its time derivative along the solution of (CTLD).

Assumption 2: The choice map σ^i and the Fenchel coupling F^i induced by h^i are both continuously differentiable, $\forall i\in\mathcal{N}$.

Continuous differentiability is a common assumption used in Lyapunov stability analysis. It holds for the Fenchel coupling function if we specify the regularizer to the Gibbs entropy given in (3). For ease of notation and analysis, functions over state and action sets are considered as matrices of size $|\mathcal{S}|\times|\mathcal{A}^i|$. Let $\langle\cdot,\cdot\rangle$ be the Frobenius inner product for the sum of the componentwise product of two matrices, and $\|\cdot\|$ be the induced matrix norm with $\|\pi\|^2=\sum_s\|\pi(\cdot|s)\|^2$. For n-player aggregation, the above two notations indicate the sum over all $i\in\mathcal{N}$.

Definition 1 (Monotonicity and Hypomonotonicity): An MG is called *monotone* if for any policy profiles $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_\dagger$, it has $\langle\boldsymbol{w}(\boldsymbol{\pi})-\boldsymbol{w}(\boldsymbol{\pi}_\dagger),\boldsymbol{\pi}-\boldsymbol{\pi}_\dagger\rangle\leq 0$. If the inequality holds only for $\langle\boldsymbol{w}(\boldsymbol{\pi})-\boldsymbol{w}(\boldsymbol{\pi}_\dagger),\boldsymbol{\pi}-\boldsymbol{\pi}_\dagger\rangle\leq\mu\|\boldsymbol{\pi}-\boldsymbol{\pi}_\dagger\|^2$ with some $\mu>0$, the game is called *μ-hypomonotone*.

Theorem 2: Consider the MG and the learning scheme provided in (CTLD). Assume there are a finite number of isolated fixed points $\bar{\boldsymbol{y}}$ of $\boldsymbol{w}\circ\boldsymbol{\sigma}$. Under Assumptions 1 and 2, if the game is μ-hypomonotone ($\mu\geq 0$) and the regularizers are K^i-strongly convex with $K>2\mu$ where $K=\min_{i\in\mathcal{N}}\{K^i\}$, then:

1) players' scores $\boldsymbol{y}_t=(y_t^i)_{i\in\mathcal{N}}$ converge to a fixed point $\bar{\boldsymbol{y}}$;
2) if further the game is individually concave, players' policies $\boldsymbol{\pi}_t=(\pi_t^i)_{i\in\mathcal{N}}$ converge to a Nash distribution $\bar{\boldsymbol{\pi}}=\boldsymbol{\sigma}(\bar{\boldsymbol{y}})$;
3) if instead the game is only locally individually concave around $\bar{\boldsymbol{\pi}}=\boldsymbol{\sigma}(\bar{\boldsymbol{y}})$, players' policies $\boldsymbol{\pi}_t$ converge to a local Nash distribution $\bar{\boldsymbol{\pi}}$.

The proof is provided in Appendix C. We here consider the monotonicity as a special case of hypomonotonicity with $\mu=0$. Take the entropic regularizer in (3) for instance, where $K=\epsilon$. If the game is monotone, players are able to converge to Nash equilibrium by taking arbitrarily small ϵ. When $\mu>0$, to ensure convergence, the system has to choose large enough $\epsilon>2\mu$ to compensate the shortage of monotonicity. But too large ϵ deviates Nash distribution away from Nash equilibrium, so a tradeoff exists. The following proposition illustrates that any MGs are hypomonotone with certain hypomonotone values. Hence, by choosing sufficient ϵ, our CTLD is convergent for any MGs.

Proposition 1: For any MGs, there always exists a finite $\mu\geq 0$ such that $\langle\boldsymbol{w}(\boldsymbol{\pi})-\boldsymbol{w}(\boldsymbol{\pi}_\dagger),\boldsymbol{\pi}-\boldsymbol{\pi}_\dagger\rangle\leq\mu\|\boldsymbol{\pi}-\boldsymbol{\pi}_\dagger\|^2$ holds for any two policy profiles.

Proof: For any policy profile $\boldsymbol{\pi}$ of a finite MG, we can analytically calculate $\rho_{\boldsymbol{\pi}}$ and $A^i_{\boldsymbol{\pi}}$, and show that the weighted advantage function $w^i(\boldsymbol{\pi})$ is a continuous function with bounded derivative on the compact set $\boldsymbol{\Pi}$. Considering two profiles $(\pi^1, \dots, \pi^j, \dots, \pi^n)$ and $(\pi^1, \dots, \pi^j_\dagger, \dots, \pi^n)$ with the difference only at the jth entry, there always exists a positive constant L^i_j such that $\|w^i(\pi^1, \dots, \pi^j, \dots, \pi^n) - w^i(\pi^1, \dots, \pi^j_\dagger, \dots, \pi^n)\| \le L^i_j \|\pi^j - \pi^j_\dagger\|$. Hence, for arbitrary two profiles $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_\dagger$ with differences at any entries, we can decompose the inner product $\langle \boldsymbol{w}(\boldsymbol{\pi}) - \boldsymbol{w}(\boldsymbol{\pi}_\dagger), \boldsymbol{\pi} - \boldsymbol{\pi}_\dagger \rangle$ by

$$\begin{aligned}
&\langle \boldsymbol{w}(\boldsymbol{\pi}) - \boldsymbol{w}(\boldsymbol{\pi}_\dagger), \boldsymbol{\pi} - \boldsymbol{\pi}_\dagger \rangle \\
&= \sum_i \Big(\Big\langle w^i\Big(\pi^1, \dots, \pi^n\Big) - w^i\Big(\pi^1_\dagger, \dots, \pi^n\Big), \pi^i - \pi^i_\dagger \Big\rangle + \\
&\quad + \Big\langle w^i\Big(\pi^1_\dagger, \pi^2, \dots, \pi^n\Big) - w^i\Big(\pi^1_\dagger, \pi^2_\dagger, \dots, \pi^n\Big), \pi^i - \pi^i_\dagger \Big\rangle + \\
&\quad \vdots \\
&\quad + \Big\langle w^i\Big(\dots, \pi^{n-1}_\dagger, \pi^n\Big) - w^i\Big(\dots, \pi^{n-1}_\dagger, \pi^n_\dagger\Big), \pi^i - \pi^i_\dagger \Big\rangle \Big) \\
&\le \sum_i \sum_j L^i_j \Big\| \pi^j - \pi^j_\dagger \Big\| \big\| \pi^i - \pi^i_\dagger \big\| \\
&\le \sum_i \sum_j \frac{1}{2} \Big(L^i_j + L^j_i \Big) \big\| \pi^i - \pi^i_\dagger \big\|^2
\end{aligned}$$

where the first inequality follows Cauchy–Schwarz inequality and the second inequality follows Young's inequality. So, an MG is always μ-hypomonotone with $\mu \le \max_i \sum_j (1/2)(L^i_j + L^j_i)$. ∎

V. Empirical Policy Optimization

Applications of CTLD to practical large games face obstacles from two aspects: 1) it is very difficult, if not impossible, to analytically evaluate players' policies on large state/action sets and evolve the learning process in continuous time and 2) policies in large-scale problems are not explicitly expressed but are parameterized by approximators like NNs [10], [42]. In this section, we develop an EPO algorithm to learn parameterized policies via RL.

We first transform CTLD to a discrete-time learning dynamics (DTLD) based on stochastic approximation [43]. The evolution of all players follows the discrete-time update rule:

$$\begin{cases} y^i_{l+1} = y^i_l + \alpha_l \eta \big(\hat{w}^i_l - y^i_l \big) \\ \pi^i_{l+1} = \sigma^i \big(y^i_{l+1} \big) \end{cases} \qquad \text{(DTLD)}$$

where l indicates the discrete-time iteration, $\hat{w}^i_l$ is the observed (noisy) weighted advantage of $\boldsymbol{\pi}_l$, and α_l is the update step. After transforming (CTLD) into (DTLD), the system runs in a discrete-time, iterative fashion, which is more easily implemented. According to the stochastic approximation theory, the long-term behavior of (DTLD) is related to that of solution trajectories of its mean-field ordinary differential equation, which can coincide with (CTLD) under certain conditions.

Theorem 3: Consider the MG and the learning scheme provided in (DTLD). Assume there are a finite number of isolated fixed points $\bar{\boldsymbol{y}}$ of $\boldsymbol{w} \circ \boldsymbol{\sigma}$. At every iteration l, each player's $\hat{w}^i_l$ is an unbiased estimate of $w^i(\boldsymbol{\pi}_l)$, that is, $\mathbb{E}[\hat{w}^i_l] = w^i(\boldsymbol{\pi}_l)$, and has $\mathbb{E}[\|\hat{w}^i_l - w^i(\boldsymbol{\pi}_l)\|^2] \le C$, for some $C \ge 0$. $\|y^i_l\|$ is always finite during the learning process. $\{\alpha_l\}$ is a deterministic sequence satisfying $\sum_{l=0}^\infty \alpha_l = \infty$ and $\sum_{l=0}^\infty \alpha_l^2 < \infty$. Under Assumptions 1 and 2, if the game is μ-hypomonotone ($\mu \ge 0$) and the regularizers is K^i-strongly convex with $K > 2\mu$ where $K = \min_{i \in \mathcal{N}} \{K^i\}$, then players' scores $\boldsymbol{y}_l$ converge almost surely to a fixed point $\bar{\boldsymbol{y}}$.

The proof is presented in Appendix D. Note that under Theorem 3, the almost sure convergence of $\boldsymbol{\pi}_l$ to a (local) Nash distribution follows the proof of Theorem 2 under the (local) individual concavity of game payoffs.

In large games, assume each player defines a policy network $\hat{\pi}^i$, parameterized by θ^i. The choice map σ^i with input y^i_l becomes finding a group of parameters θ^i_l that minimize the loss

$$\min_{\theta^i} \mathcal{L}^i\big(\theta^i\big) = \min_{\theta^i} \sum_{s,a^i} y^i_l\big(s, a^i\big) \hat{\pi}^i\big(s, a^i | \theta^i\big) - h^i\big(\hat{\pi}^i\big(\theta^i\big)\big).$$

To avoid extreme change of policy behaviors $\{\hat{\pi}^i_l\}$ along iterations, we restrict the new $\hat{\pi}^i(\theta^i_l)$ is trained along the loss gradient $\partial \mathcal{L}^i / \partial \theta^i$, starting from last θ^i_{l-1}, and use an early stop [11] to bound the KL divergence between the new and old policies, that is, $\mathbb{E}_{s \sim \rho_{l-1}}[D_{\mathrm{KL}}(\hat{\pi}^i(\theta^i_{l-1}) \| \hat{\pi}^i(\theta^i_l))] \le c$.

If we specify DTLD with $\alpha_l = (1/l)$ and $\eta = 1$ and ignore the noise effect, y^i_l is actually the average of past weighted advantages

$$\begin{aligned}
y^i_l\big(s, a^i\big) &= \frac{1}{l} \sum_{j=0}^{l-1} \rho_j(s) A^i_j\big(s, a^i\big) \\
&= \frac{1}{l} \sum_{j=0}^{l-1} \rho_j(s) Q^i_j\big(s, a^i\big) - \frac{1}{l} \sum_{j=0}^{l-1} \rho_j(s) V^i_j(s).
\end{aligned}$$

Because state-dependent terms make no difference to the gradient $\partial \mathcal{L}^i / \partial \theta^i$, the above sum of values can be replaced by an *empirical* value network $\hat{V}^i(s|\phi^i)$ that learns the average of historical weighted values, that is, $\hat{V}^i(s|\phi^i) \approx (1/[\sum_{j=0}^{l-1} \rho_j(s)]) \sum_{j=0}^{l-1} \rho_j(s) V^i_j(s)$. The weighted calculation $\rho_j(s)[\dots]$ is equivalent to the expectation $(1/1-\gamma)\mathbb{E}_{s \sim \rho_j}[\dots]$, and can be further approximated by samples observed at every iteration [25].

With the value network, the score becomes $y^i_l(s, a^i) = (1/l) \sum_{j=0}^{l-1} \rho_j(s)(Q^i_j(s, a^i) - \hat{V}^i(s|\phi^i))$. The return $G^i_{j,k}$ on the on-policy trajectory $(s_k, \boldsymbol{a}_k, s_{k+1}, \boldsymbol{a}_{k+1}, \dots)$ generated by $\hat{\boldsymbol{\pi}}_j$ is an unbiased estimate of $Q^i_j(s_k, a^i_k)$, but suffers from high variance. A commonly used form in modern RL is generalized advantage estimator (GAE) [44], which is a biased but low-variance estimate. For any segment of trajectory $(s_k, a^i_k, r^i_k, s_{k+1}, a^i_{k+1}, r^i_{k+1}, \dots)$ in the historical experience, with the support of $\hat{V}^i(\phi^i)$, player i's λ-GAE is defined as $\hat{A}^i_k = \sum_{\nu=0}^\infty (\gamma\lambda)^\nu \delta^i_{k+\nu}$, where $\delta^i_{k+\nu} = r^i_{k+\nu} + \gamma \hat{V}^i(s_{k+\nu+1}|\phi^i) - \hat{V}^i(s_{k+\nu}|\phi^i)$ is the temporal difference, and $\lambda \in [0, 1]$ is a constant that balances the bias and variance of estimate. The policy loss now becomes

$$\mathcal{L}^i\big(\theta^i\big) = \sum_{\mathcal{D}^i} \Big[\hat{A}^i\big(s_k, a^i_k\big) \hat{\pi}^i\big(s_k, a^i_k | \theta^i\big) - h^i\big(\hat{\pi}^i\big(s_k | \theta^i\big)\big) \Big]$$

Algorithm 1 EPO for n-Player MGs

1: Initialize policy and value parameters, $\boldsymbol{\theta}_0 = (\theta_0^i)_{i\in\mathcal{N}}$, $\boldsymbol{\phi}_0 = (\phi_0^i)_{i\in\mathcal{N}}$; define experience buffer $\mathcal{D}^i = \emptyset$, $\forall i \in \mathcal{N}$; select entropic parameter ϵ, GAE parameter λ, KL divergence threshold c;
2: **for** $l = 0, 1, \ldots,$ **do**
3: Players play their own $\hat{\pi}^i(\theta_l^i)$ in game and observe trajectories $\tau_l^i = \{(s_k, a_k^i, r_k^i, s_{k+1})\}$;
4: **for** each player i **do**
5: Calculate return G_k^i along τ_l^i and store $\{(s_k, a_k^i, r_k^i, s_{k+1}, G_k^i)\}$ in $\mathcal{D}^i$;
6: Fit empirical value network $\hat{V}^i(\phi^i)$ over $\mathcal{D}^i$ by regression on mean-squared error $\min_{\phi^i} \sum_{s_k\in\mathcal{D}^i} \left(\hat{V}^i(s_k|\phi^i) - G_k^i\right)^2$;
7: Compute λ-GAE $\hat{A}_k^i$ for every sample in $\mathcal{D}^i$ based on the regressed $\hat{V}^i(\phi^i)$;
8: Train policy network along the gradient of policy loss $\mathcal{L}^i(\theta^i)$, starting from the current θ_l^i with KL divergence threshold $\mathbb{E}_{s_k\sim\tau_l^i}[D_{\mathrm{KL}}(\hat{\pi}^i(\theta_l^i)\|\hat{\pi}^i(\theta^i))] \le c$;
9: Take the trained θ^i as θ_{l+1}^i.
10: **end for**
11: **end for**

where $\mathcal{D}^i$ is the experience buffer of player i throughout the entire history. Empirically, the clipping technique proposed by Schulman *et al.* [11] is helpful to stabilize the optimization.

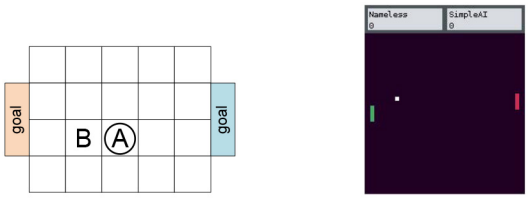
After combining (DTLD) with DRL techniques, the EPO is proposed and its entire process is summarized in Algorithm 1. The learning is totally distributed in the sense that each player trains its value and policy networks based on own observations of states, actions, and rewards (lines 5–9). It requires no knowledge of game structure (how many players are playing and what the others' rewards are defined) and does not need to monitor the other behaviors, so the scheme is applicable to an arbitrary number of players. All players play their current policies (line 3) in the same game, while another n-player framework–PSRO [16] has to match each player with specific opponents.

The procedure of EPO shows similarity to that of the proximal policy optimization (PPO) [11], but with fundamental differences. EPO follows the idea of CTLD that updates multiple players' policies based on the aggregation of their in-game performance over historical iterations, in contrast to PPO that optimizes a single-agent policy with the experience of the current iteration. With the entire historical experience, EPO updates multiplayer policies in the direction of the Nash equilibrium.

VI. Experiments

In experiments, we consider 2-player *Soccer* game [3], [22], 3-player *Cournot-Competition* game [19], and 2-player *Wimblepong* game.[2]

[2]https://github.com/aalto-intelligent-robotics/wimblepong



(a) (b)

Fig. 2. Illustrations of games in experiments. (a) Soccer. (b) Wimblepong.

1) *Soccer:* On a 4 × 5 board, shown in Fig. 2(a), two players play a ball to goal. The player possessing the ball is marked by a circle. Each player can move Up, Down, Left, Right, or Stay. At an instant, two-side moves are executed in random order. If the move would take the player to the opponent square, the possession of the ball yields to the stationary player, and the move is canceled. The winning player receives a reward of +100, while the opponent receives −100. Every step has a probability of 0.01 terminating the game as a draw. γ selects 0.95.
2) *Cournot Competition:* The original cournot competition [19] is a continuous game, but here we modify it to an MG. At step k in each round, the market price of the same good is determined by the production of each firm, modeled by $P(\boldsymbol{x}_k) = a - \sum_i b^i x_k^i$, where x_k^i is firm i's production, and a, b^i are constants. Each firm can choose to increase or decrease its next-step production by Δx^i, or remain unchanged. But due to technical defects and incorrect manipulation, the decision has only p^i probability of being successfully executed and $(1-p^i)$ probability of leading to the other two outcomes. The production capacity is bounded by C^i. The reward of firm i is given by $r^i(\boldsymbol{x}_k) = x_k^i P(\boldsymbol{x}_k) - c^i x_k^i$, where c^i represents its marginal production cost. In our 3-player setting, $p^i = 0.8$, $\Delta x^i = 20$, $a = 400$, $b^i = 2$, and $C^i = 100$, and $c^1 = 40$, $c^2 = 35$, and $c^3 = 42$. γ selects 0.9.
3) *Wimblepong:* It is a 2-player version of Atari game Pong [10], as illustrated in Fig. 2(b). Each player controls a paddle to play a ball with the other, and can take one of three actions: moving up or down, or staying in place. The game state consists of positions of two paddles, and position and velocity of the ball. If a player misses a ball, it receives −10 reward and the opponent receives +10 reward. γ selects 0.99.

A. Numerical Examples

We apply the proposed CTLD to learn Nash equilibria for the first two games.[3] Gibbs entropy is adopted as the regularizer for the benefit of softmax choice map. For comparison, we consider the iterated best response (IBR) [45], fictitious play (FP) [5], PSRO [16], and exploitability descent (ED) in tabular forms [14], and use policy iteration [46] as their

[3]The MATLAB implementation is available in https://github.com/YuanhengZhu/Continuous-time-learning-dynamics.

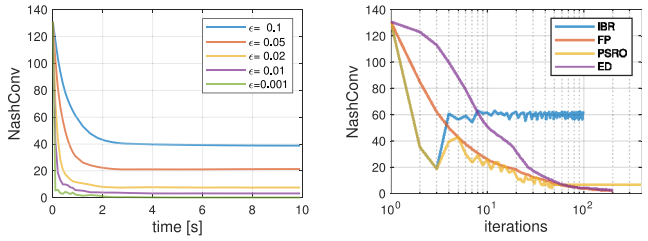


Fig. 3. NashConv learning curves on soccer. (a) CTLD. (b) IBR versus FP versus PSRO versus ED.

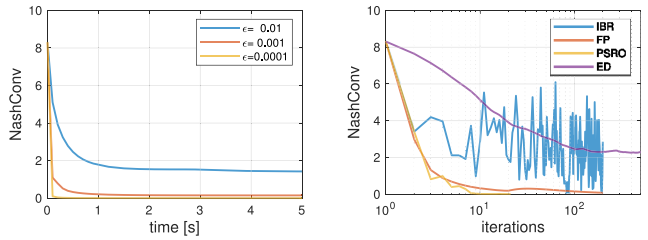


Fig. 4. NashConv learning curves on Cournot competition. (a) CTLD. (b) IBR versus FP versus PSRO versus ED.

oracles for best response. PSRO relies on a metasolver to synthesize metastrategies for each player, so we choose linear programming [47] for the 2-player case and the EXP-D-RL method proposed in [20] for the 3-player case. ED is originally proposed in [14] for 2-player zero-sum game, but here is also applied to the 3-player game. Hyperparameters have been empirically selected, and implementation details are presented in Appendix E.

The NashConvs of each method along the learning process are plotted in Figs. 3 and 4. Fig. 3(a) is obtained by the same $\eta = 1$ and different $\epsilon = 0.1$, 0.05, 0.02, 0.01, and 0.001. Fig. 4(a) is obtained by the same $\eta = 1$ and different $\epsilon = 0.1$, 0.01, and 0.001. Since CTLD and the other methods run in different time scales, their results are presented separately in different plots. In both experiments, CTLD remains convergent under any regularizer parameter ϵ, and is able to approach Nash equilibria with arbitrary precision if ϵ is close enough to 0. Another empirical learning method, FP, also shows a consistent convergence property. PSRO is remarkable in approaching Nash equilibrium in Cournot Competition, but ends up with a noticeable NashConv gap in Soccer game. The convergence of ED in the 2-player case is guaranteed by the theoretical results in [14], but the argument is not valid for more than two players, resulting in a large gap of NashConv in Cournot Competition. IBR suffers from strategic cycles, so it is hard to converge.

We also numerically investigate the hypomonotone values μ of two games. By randomly choosing two policy profiles $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_\dagger$, the result of $\langle \boldsymbol{w}(\boldsymbol{\pi}) - \boldsymbol{w}(\boldsymbol{\pi}_\dagger), \boldsymbol{\pi} - \boldsymbol{\pi}_\dagger \rangle / \|\boldsymbol{\pi} - \boldsymbol{\pi}_\dagger\|^2$ is an underestimate of true μ. The distributions of 1000 samples in two games are plotted in Fig. 5. The true μ is inferred to be greater than 0.0129 in Soccer and greater than 0.0032 in Cournot Competition. It reflects that the convergence condition

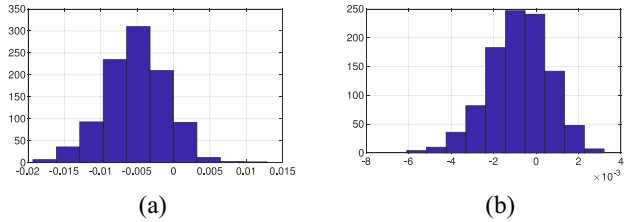


Fig. 5. Histograms of sampled $\langle \boldsymbol{w}(\boldsymbol{\pi}) - \boldsymbol{w}(\boldsymbol{\pi}_\dagger), \boldsymbol{\pi} - \boldsymbol{\pi}_\dagger \rangle / \|\boldsymbol{\pi} - \boldsymbol{\pi}_\dagger\|^2$. (a) Soccer game. (b) Cournot competition.

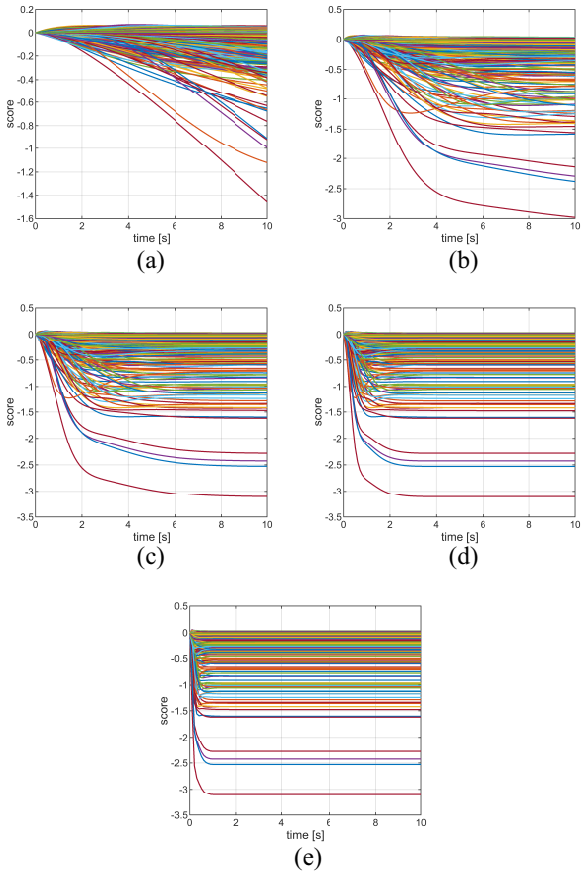
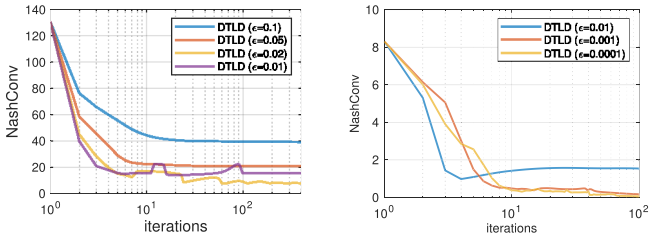


Fig. 6. Evolution of CTLD scores at the same $\epsilon = 0.05$ but different η. (a) $\eta = 0.1$. (b) $\eta = 0.5$. (c) $\eta = 1$. (d) $\eta = 3$. (e) $\eta = 10$.

$K(= \epsilon) > 2\mu$ in Theorem 2 is not that strict, since we have observed with smaller $\epsilon < 2\mu$, the CTLD still converges in both games.

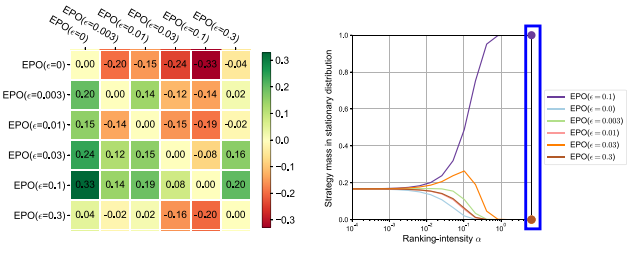
Another hyperparameter in CTLD is the learning rate η, so we repeat CTLD in Soccer game at the same $\epsilon = 0.05$ but different $\eta = 0.1$, 0.5, 1, 3, and 10, to observe the effect of the learning rate. The evolution of scores is plotted in Fig. 6. It is observed that large η has no influence on the converged results, but is able to accelerate the convergence rate.

We repeat the two numerical experiments with DTLD and plot the learning curves in Fig. 7. The behaviors of DTLD are consistent with CTLD among most experiments except the one in Soccer game with $\epsilon = 0.01$. The curve ends up with an obvious NashConv gap in contrast to its counterpart in CTLD. One explanation is that when ϵ is nearly 0, the choice-mapped



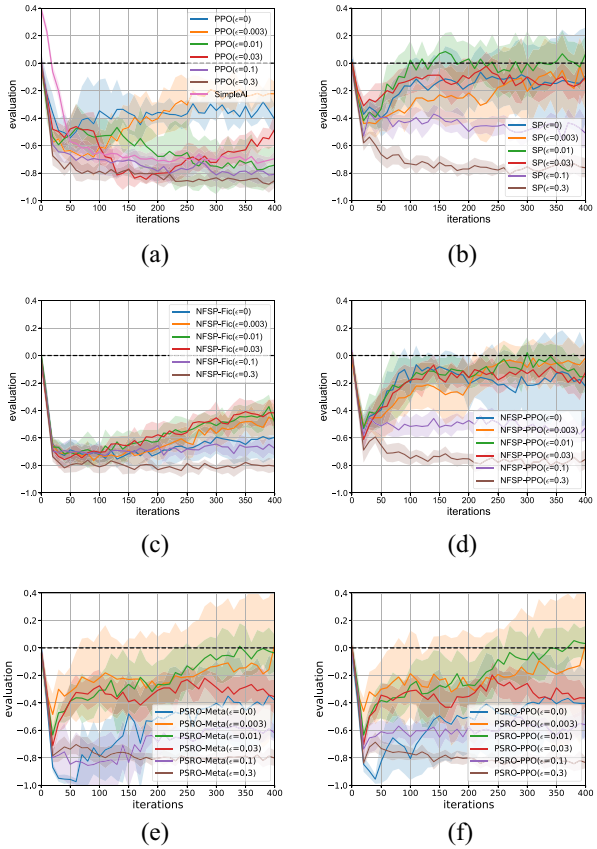
(a) (b)

Fig. 7. NashConv learning curves of DTLD on numerical examples. (a) Soccer. (b) Cournot competition.



(a) (b)

Fig. 8. Evaluation of EPO agents learned under different ϵ. (a) Payoff table. (b) Ranking-intensity sweep.



(a) (b) (c) (d) (e) (f)

Fig. 9. Performance of SP, NFSP, PSRO, PPO, and SimpleAI relative to baseline EPO($\epsilon = 0.1$) in the learning process. The curves are averaged over random seeds with solid lines indicating mean values and shadow areas indicating standard variance. NFSP-Fic indicates the fictitious player and NFSP-PPO indicates the fellow PPO agent. PSRO-Meta indicates the metastrategy and PSRO-PPO indicates the fellow PPO agent. (a) PPO/SimpleAI versus EPO. (b) SP versus EPO. (c) NFSP-Fic versus EPO. (d) NFSP-PPO versus EPO. (e) PSRO-Meta versus EPO. (f) PSRO-PPO versus EPO.

policy tends to greedily select extreme actions, instead of a smooth action distribution. Discrete-time dynamical system further enlarges the discontinuity of the update of policies, so the asymptotic behavior of DTLD does not follow the flow of CTLD. While in Cournot Competition, DTLD shows no significant degradation even when ϵ chooses quite small values, probably because the two problems have different score ranges.

B. Large-Scale Example

The third Wimblepong game is large scale, so EPO is applied. We run the experiments with different regularizer parameters ϵ and select common values in the RL literature for the rest algorithm parameters. To reduce random errors, each experiment is repeated three times. After 400 iterations, the learned agents under different ϵ are matched in pairs to evaluate their agent-level payoff table. The payoff value is calculated by the difference between two-side win rates, and is averaged over matches played by agents that are obtained in different runs. We use the multiagent evaluation and ranking metric, α-Rank [27], to evaluate agent rankings, and present the results in Fig. 8. EPO with $\epsilon = 0.1$ shows dominance in playing against the other EPO agents. Small ϵ causes the algorithm to prematurely stop exploration and fall into local optima, while large ϵ disturbs action selection.

For comparison, we choose self-play (SP), neural fictitious SP (NFSP) [48], Nash-based PSRO [16], and PPO [11] against a script-based SimpleAI opponent. For fairness, the RL parts of SP, NFSP, and PSRO are all based on a PPO agent. The fictitious player in NFSP is trained by supervised learning based on the historical behavior of the fellow agent. The opponent metastrategy in PSRO is the Nash mixture of historical policies. The algorithms choose the same parameters as EPO and vary the entropic parameter ϵ in training objectives to produce a variety of agents. We take the learning process of EPO with the best $\epsilon = 0.1$ as baseline and evaluate the relative performance of these algorithms against EPO along the same number of iterations.

The curves of relative performance are plotted in Fig. 9, and a common phenomenon is that all curves immediately drop below zero once the learning starts. It indicates no algorithm improves policies as fast as EPO, and in other words, EPO is advantageous in finding policy gradient toward Nash equilibrium. If only playing against a fixed opponent, PPO agents are not possible to approach Nash equilibrium, shown by the low relative performance against EPO in Fig. 9(a). With the increase of iterations, SP, NFSP-PPO, and PSRO-PPO agents with specific entropic parameters are able to narrow the gap. Wimblepong game does not severely suffer from strategic cycles [17], so even for simple SP, it is possible to approach Nash equilibrium by beating ever-improving opponents. The fictitious player of NFSP makes much slower progress than its PPO fellow.

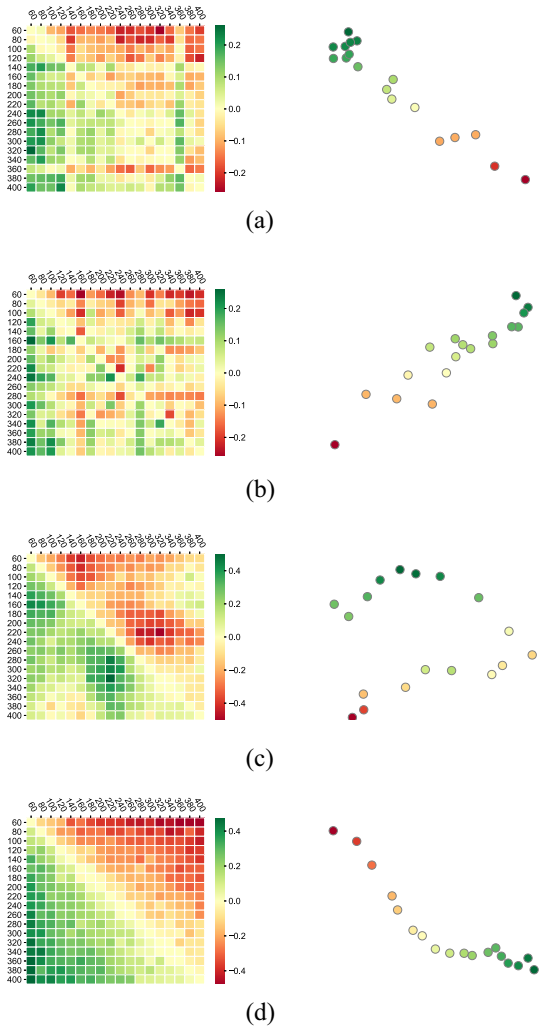


Fig. 10. Payoff tables and visualization of EPO populations under different sizes of experience buffers. *Left*: Row and column labels indicate agents at different iterations. *Right*: 2-D embedding of payoff tables by using the first two dimensions of Schur decomposition; Color corresponds to average payoff of an agent against entire population. (a) Latest 1-iteration replay. (b) Latest 10-iteration replay. (c) Latest 100-iteration replay. (d) Full historical replay.

Ablation Study: We now investigate the effect of historical experience in EPO and run experiments with different sizes of experience buffer. Note that when the buffer stores only experience of the latest iteration, the algorithm becomes SP. Agents after different iterations in an experiment form a population and their payoff table is drawn in Fig. 10. We also plot the 2-D visualization by Schur decomposition [17] at the right of the figure. Full replay of historical experience makes EPO update policies in a transitive and monotone mode. Limited replay makes the algorithm suffer from policy forgetting, in the sense that new policies may forget how to beat some old policies in history. It corresponds to cyclic or mixed shapes in the 2-D embedding of policy populations.

VII. Conclusion

A game-theoretic learning framework for n-player MGs was proposed. The convergence of the dynamical learning system to an approximate Nash equilibrium is proved by the Lyapunov stability theory, and is also verified on different n-player MG examples. The combination of NNs makes the EPO algorithm applicable to large games. The distributed implementation and no need of game interactions with specific opponents makes it appealing to companies and groups that are less intensive in computing resources.

There is still space for improvement. The existence of multiple Nash equilibria may pose a risk to our work, leading to the decrease of social welfare. Correlated equilibrium [2], [6] is a potential solution, since it can be seen as a superset of Nash equilibrium. After drawing an action profile from the distribution of correlated equilibrium, playing the suggested action is a best response for each player, given that everyone else will play their suggested actions. We encourage research to investigate how small a common knowledge can be introduced to achieve a promising outcome in coordination games. Another issue lies in the ever-increasing size of historical experience. For really large-scale problems, it is infeasible to store and replay the entire history. There are two potential solutions. One is to use a minibatch set, instead of the entire history, to conduct EPO at each training step. The second is to restrict the experience buffer to store only data of the last finite number of iterations. These will be our future research directions.

Appendix A
Linear Algebra in CTLD

Consider each player's policy $\pi^i : \mathcal{S} \times \mathcal{A}^i \to [0,1]$ as a matrix with rows indicating states, columns indicating actions, and elements indicating probabilities. Similarly, given a policy profile $\boldsymbol{\pi}$, the state transition function $\mathcal{P}_{\boldsymbol{\pi}} : \mathcal{S} \times \mathcal{S} \to [0,1]$ is expressed in matrix with each entry equal to

$$\mathcal{P}_{\boldsymbol{\pi}}(s,s') = \sum_{a^1}\sum_{a^2}\cdots\sum_{a^n}\pi^1\left(a^1|s\right)\pi^2\left(a^2|s\right)\ldots\pi^n\left(a^n|s\right)\times \\ \times\ \mathcal{P}\left(s'|s,a^1,a^2,\ldots,a^n\right)$$

and the expected reward function $\mathcal{R}^i_{\boldsymbol{\pi}} : \mathcal{S} \to \mathbb{R}$ under $\boldsymbol{\pi}$ is a vector with each entry equal to

$$\mathcal{R}^i_{\boldsymbol{\pi}}(s) = \sum_{a^1}\sum_{a^2}\cdots\sum_{a^n}\pi^1\left(a^1|s\right)\pi^2\left(a^2|s\right)\ldots\pi^n\left(a^n|s\right)\times \\ \times\ \mathcal{R}^i\left(s,a^1,a^2,\ldots,a^n\right).$$

Then, the state visitation frequency $\rho_{\boldsymbol{\pi}}$, value $V^i_{\boldsymbol{\pi}}$, Q value $Q^i_{\boldsymbol{\pi}}$, and advantage $A^i_{\boldsymbol{\pi}}$ are analytically calculated by

$$\begin{aligned}\rho_{\boldsymbol{\pi}} &= \left(I-\gamma\mathcal{P}^T_{\boldsymbol{\pi}}\right)^{-1}\rho_0\\ V^i_{\boldsymbol{\pi}} &= (I-\gamma\mathcal{P}_{\boldsymbol{\pi}})^{-1}\mathcal{R}^i_{\boldsymbol{\pi}}\\ Q^i_{\boldsymbol{\pi}} &= \mathcal{R}^i_{\boldsymbol{\pi}^{-i}}+\gamma\mathcal{P}_{\boldsymbol{\pi}^{-i}}V^i_{\boldsymbol{\pi}}\\ A^i_{\boldsymbol{\pi}} &= Q^i_{\boldsymbol{\pi}}-V^i_{\boldsymbol{\pi}}.\end{aligned}$$

Note that $I-\gamma\mathcal{P}^T_{\boldsymbol{\pi}}$ and $I-\gamma\mathcal{P}_{\boldsymbol{\pi}}$ are always invertible due to the property of transition matrix. The weighted advantage in CTLD is obtained by inserting above analytic solutions into

$$w^i(\boldsymbol{\pi}) = \rho_{\boldsymbol{\pi}} \odot A^i_{\boldsymbol{\pi}}$$

where $\odot$ indicates the Hadamard (elementwise) product.

APPENDIX B
PROOF OF THEOREM 1

Before proving Theorem 1, a useful lemma on Nash equilibrium is first introduced.

Lemma 2: If $\boldsymbol{\pi}_*$ is a Nash equilibrium to MG with regularized payoff, then

$$\sum_s \rho_{\boldsymbol{\pi}_*}(s) \sum_{a^i} \pi_*^i(s,a^i) A^i_{\boldsymbol{\pi}_*}(s,a^i) - h^i(\pi_*^i) \geq \sum_s \rho_{\boldsymbol{\pi}_*}(s) \sum_{a^i} \pi^i(s,a^i) A^i_{\boldsymbol{\pi}_*}(s,a^i) - h^i(\pi^i)$$

for all $\pi^i \in \Pi^i$, $i \in \mathcal{N}$.

Proof: For any π^i, let $\pi_\dagger^i = (1-\alpha)\pi_*^i + \alpha\pi^i$, where $\alpha \in [0,1]$. According to Lemma 1

$$u^i(\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}) = u^i(\pi_*^i, \boldsymbol{\pi}_*^{-i}) + \sum_s \rho_{\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}}(s) \sum_{a^i} \pi_\dagger^i(s,a^i) A^i_{\boldsymbol{\pi}_*}(s,a^i).$$

After substituting it into $U^i(\pi_\dagger^i, \boldsymbol{\pi}_*^{-i})$ and based on the convexity of h^i with $h^i(\pi_\dagger^i) \leq (1-\alpha)h^i(\pi_*^i) + \alpha h^i(\pi^i)$, we have

$$\begin{aligned} &U^i(\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}) \\ &= u^i(\pi_*^i, \boldsymbol{\pi}_*^{-i}) + \sum_s \rho_{\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}}(s) \sum_{a^i} \pi_\dagger^i(s,a^i) A^i_{\boldsymbol{\pi}_*}(s,a^i) - h^i(\pi_\dagger^i) \\ &\geq u^i(\pi_*^i, \boldsymbol{\pi}_*^{-i}) - h^i(\pi_*^i) + \sum_s \rho_{\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}}(s) \sum_{a^i} \pi_\dagger^i(s,a^i) A^i_{\boldsymbol{\pi}_*}(s,a^i) \\ &\quad + \alpha\big(h^i(\pi_*^i) - h^i(\pi^i)\big) \\ &= U^i(\pi_*^i, \boldsymbol{\pi}_*^{-i}) + g(\alpha) \end{aligned}$$

where

$$g(\alpha) = \sum_s \rho_{\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}}(s) \sum_{a^i} \pi_\dagger^i(s,a^i) A^i_{\boldsymbol{\pi}_*}(s,a^i) + \alpha\big(h^i(\pi_*^i) - h^i(\pi^i)\big).$$

Based on the fact that $\boldsymbol{\pi}_*$ is Nash equilibrium to the regularized payoff, $U^i(\pi_*^i, \boldsymbol{\pi}_*^{-i}) \geq U^i(\pi_\dagger^i, \boldsymbol{\pi}_*^{-i})$, $g(\alpha)$ must have $g(\alpha) \leq 0$ $\forall \alpha \in [0,1]$. When $\alpha = 0$, $\pi_\dagger^i = \pi_*^i$ and $g(0) = 0$ by definition, it is inferred that $\nabla_\alpha g(\alpha)|_{\alpha=0_+} \leq 0$. Calculating the derivative of $g(\alpha)$

$$\nabla_\alpha g(\alpha) = \sum_s \sum_{a^i} \Big(\nabla_\alpha \rho_{\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}}(s) \pi_\dagger^i(s,a^i) A_{\pi_*^i, \boldsymbol{\pi}_*^{-i}}(s,a^i) + \rho_{\pi_\dagger^i, \boldsymbol{\pi}_*^{-i}}(s) \nabla_\alpha \pi_\dagger^i(s,a^i) A_{\pi_*^i, \boldsymbol{\pi}_*^{-i}}(s,a^i) \Big) + h^i(\pi_*^i) - h^i(\pi^i)$$

and using the fact that $\sum_{a^i} \pi_*^i(s,a^i) A_{\pi_*^i, \boldsymbol{\pi}_*^{-i}}(s,a^i) = 0$ yields

$$\begin{aligned} &\nabla_\alpha g(\alpha)|_{\alpha=0_+} \\ &= \sum_s \rho_{\boldsymbol{\pi}_*}(s) \sum_{a^i} \big(\pi^i(s,a^i) - \pi_*^i(s,a^i)\big) A^i_{\boldsymbol{\pi}_*}(s,a^i) + h^i(\pi_*) - h^i(\pi^i) \leq 0 \end{aligned}$$

which implies the conclusion. ■

Proof of Theorem 1: 1) If $\boldsymbol{\pi}_*$ is a Nash equilibrium with respect to U^i, Lemma 2 implies that $\boldsymbol{\pi}_* = \boldsymbol{\sigma}(\boldsymbol{y}_*)$ where $\boldsymbol{y}_* = \boldsymbol{w}(\boldsymbol{\pi}_*)$. After inserting into $\boldsymbol{y}_* = \boldsymbol{w}(\boldsymbol{\pi}_*)$, we conclude $\boldsymbol{y}_*$ is the fixed point.

2) With the fixed point $\bar{\boldsymbol{y}}$ and the induced $\bar{\boldsymbol{\pi}}$, consider any $\pi^i \in \Pi^i$ and let $\pi_\dagger^i = (1-\alpha)\bar{\pi}^i + \alpha\pi^i$ where $\alpha \in [0,1]$.

Lemma 1 implies that

$$\begin{aligned} &u^i(\pi_\dagger^i, \bar{\boldsymbol{\pi}}^{-i}) \\ &= u^i(\bar{\pi}^i, \bar{\boldsymbol{\pi}}^{-i}) + \sum_s \rho_{\pi_\dagger^i, \bar{\boldsymbol{\pi}}^{-i}}(s) \sum_{a^i} \pi_\dagger^i(s,a^i) A^i_{\bar{\boldsymbol{\pi}}}(s,a^i) \\ &= u^i(\bar{\pi}^i, \bar{\boldsymbol{\pi}}^{-i}) + \sum_s \rho_{\pi_\dagger^i, \bar{\boldsymbol{\pi}}^{-i}}(s) \sum_{a^i} \alpha\big(\pi^i(s,a^i) - \bar{\pi}^i(s,a^i)\big) A^i_{\bar{\boldsymbol{\pi}}}(s,a^i) \end{aligned}$$

where the second equality uses the fact that $\sum_{a^i} \bar{\pi}^i(s,a^i) A^i_{\bar{\boldsymbol{\pi}}}(s,a^i) = 0$. Based on the individual concavity of u^i, that is, $u^i(\pi_\dagger^i, \bar{\boldsymbol{\pi}}^{-i}) \geq (1-\alpha)u^i(\bar{\pi}^i, \bar{\boldsymbol{\pi}}^{-i}) + \alpha u^i(\pi^i, \bar{\boldsymbol{\pi}}^{-i})$, it is inferred that

$$\sum_s \rho_{\pi_\dagger^i, \bar{\boldsymbol{\pi}}^{-i}}(s) \sum_{a^i} \big(\pi^i(s,a^i) - \bar{\pi}^i(s,a^i)\big) A^i_{\bar{\boldsymbol{\pi}}}(s,a^i) \geq u^i(\pi^i, \bar{\boldsymbol{\pi}}^{-i}) - u^i(\bar{\pi}^i, \bar{\boldsymbol{\pi}}^{-i}). \tag{4}$$

Since $\bar{\boldsymbol{y}} = \boldsymbol{w} \circ \boldsymbol{\sigma}(\bar{\boldsymbol{y}})$ and $\bar{\boldsymbol{\pi}} = \boldsymbol{\sigma}(\bar{\boldsymbol{y}})$, then

$$\sum_s \rho_{\bar{\boldsymbol{\pi}}}(s) \sum_{a^i} \bar{\pi}^i(s,a^i) A^i_{\bar{\boldsymbol{\pi}}}(s,a^i) - h^i(\bar{\pi}^i) \geq \sum_s \rho_{\bar{\boldsymbol{\pi}}}(s) \sum_{a^i} \pi^i(s,a^i) A^i_{\bar{\boldsymbol{\pi}}}(s,a^i) - h^i(\pi^i).$$

After taking $\alpha = 0$, we conclude that

$$u^i(\bar{\pi}^i, \bar{\boldsymbol{\pi}}^{-i}) - h^i(\bar{\pi}^i) \geq u^i(\pi^i, \bar{\boldsymbol{\pi}}^{-i}) - h^i(\pi^i)$$

holds for all $\pi^i \in \Pi^i$ $\forall i \in \mathcal{N}$ and, thus, $\bar{\boldsymbol{\pi}}$ is a Nash equilibrium to the game with regularized payoffs. ■

APPENDIX C
PROOF OF THEOREM 2

The following lemmas of Fenchel-coupling function $F^i(\pi^i, y^i)$ are useful for the proof of Theorem 2.

Lemma 3: Let h^i be a K^i-strongly convex regularizer. For all $\pi^i \in \Pi^i$ and $y^i \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}^i|}$, the induced Fenchel coupling has

$$F^i(\pi^i, y^i) \geq \frac{1}{2} K^i \big\| \sigma^i(y^i) - \pi^i \big\|^2.$$

Proof: According to [19, Proposition 4.3(b)], the inequality holds at every state with vector norm. The conclusion for matrix form is obtained by summing up all states. ■

Lemma 4: Let h^i be the regularizer, σ^i be the choice map, and F^i be the Fenchel coupling. Under Assumptions 1 and 2, the following holds for all $\pi^i \in \Pi^i$ and $y^i \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}^i|}$:

$$\nabla_{y^i} F^i(\pi^i, y^i) = \sigma^i(y^i) - \pi^i$$

Proof: Viewing π^i and y^i as matrices of size $|\mathcal{S}| \times |\mathcal{A}^i|$, according to the envelope theorem [49, Th. 1.F.1], the derivative of $\max_{p \in \Delta(|\mathcal{A}^i|)} \sum_{a^i} y^i(s,a^i) p_{a^i} - h^i(p)$ toward the sth row

of y^i is equal to

$$\nabla_{y_s^i}\left(\max_{p\in\Delta(|\mathcal{A}^i|)}\sum_{a^i}y^i(s,a^i)p_{a^i}-h^i(p)\right)=\left[\sigma^i(y^i)\right]_s.$$

Besides, $\nabla_{y_s^i}(\sum_{a^i}y^i(s,a^i)\pi^i(s,a^i)-h^i(\pi^i(s)))=\pi_s^i$. By stacking the above derivatives in rows and based on the definition of F^i, we have $\nabla_{y^i}F^i(\pi^i,y^i)=\sigma^i(y^i)-\pi^i$. ∎

Proof of Theorem 2: 1) Consider the semi-positive function $F^i(\bar{\pi}^i,y_t^i)$ on y_t^i. By Lemma 4, its derivative toward y_t^i equals $\nabla_{y_t^i}F^i(\bar{\pi}^i,y_t^i)=\sigma^i(y_t^i)-\bar{\pi}^i$. Along the solution of (CTLD), the time derivative of $F^i(\bar{\pi}^i,y_t^i)$ has

$$\begin{aligned}
&\dot{F}^i(\bar{\pi}^i,y_t^i)\\
&=\eta\langle w^i(\boldsymbol{\pi}_t)-y_t^i,\sigma^i(y_t^i)-\bar{\pi}^i\rangle\\
&=\eta\big(\langle w^i(\boldsymbol{\pi}_t),\sigma^i(y_t^i)-\bar{\pi}^i\rangle-\langle y_t^i,\sigma^i(y_t^i)-\bar{\pi}^i\rangle\big)\\
&=\eta\big(\langle w^i(\boldsymbol{\pi}_t),\sigma^i(y_t^i)-\bar{\pi}^i\rangle+h^i(\bar{\pi}^i)-h^i(\pi_t^i)-F^i(\bar{\pi}^i,y_t^i)\big)\\
&\le\eta\big(\langle w^i(\boldsymbol{\pi}_t),\sigma^i(y_t^i)-\bar{\pi}^i\rangle-\langle\bar{y}^i,\pi_t^i-\bar{\pi}^i\rangle-F^i(\bar{\pi}^i,y_t^i)\big)\\
&=\eta\big(\langle w^i(\boldsymbol{\pi}_t)-w^i(\bar{\boldsymbol{\pi}}),\pi_t^i-\bar{\pi}^i\rangle-F^i(\bar{\pi}^i,y_t^i)\big)
\end{aligned}$$

where the third equality follows the definition of $F^i(\bar{\pi}^i,y_t^i)$, and the inequality is based on $\bar{\pi}^i=\sigma^i(\bar{y}^i)$, which yields:

$$\langle\bar{y}^i,\bar{\pi}^i\rangle-h^i(\bar{\pi}^i)\ge\langle\bar{y}^i,\pi_t^i\rangle-h^i(\pi_t^i).$$

The above analysis holds for all $i\in\mathcal{N}$, so we can take $\mathcal{V}(\boldsymbol{y}_t)=\sum_{i\in\mathcal{N}}F^i(\bar{\pi}^i,y_t^i)$ as the Lyapunov function, whose time derivative has

$$\dot{\mathcal{V}}(\boldsymbol{y}_t)\le\eta\sum_i\big(\langle w^i(\boldsymbol{\pi}_t)-w^i(\bar{\boldsymbol{\pi}}),\pi_t^i-\bar{\pi}^i\rangle-F^i(\bar{\pi}^i,y_t^i)\big).$$

The assumption on game hypomonotonicity and Lemma 3 implies

$$\dot{\mathcal{V}}(\boldsymbol{y}_t)\le-\frac{1}{2}\eta(K-2\mu)\big\|\boldsymbol{\sigma}(\boldsymbol{y}_t)-\boldsymbol{\sigma}(\bar{\boldsymbol{y}})\big\|^2.$$

Thus, under $K\ge2\mu$, $\dot{\mathcal{V}}(\boldsymbol{y}_t)\le0\ \forall\boldsymbol{y}_t$, and $\dot{\mathcal{V}}(\boldsymbol{y}_t)=0$ for all $\boldsymbol{y}_t\in\mathcal{E}=\{\boldsymbol{y}\mid\boldsymbol{\sigma}(\boldsymbol{y})=\boldsymbol{\sigma}(\bar{\boldsymbol{y}})\}$, let $\mathcal{M}$ be the largest invariant set in $\mathcal{E}$. By LaSalle's invariance principle [41, Th. 4.4], from any starting point, $\boldsymbol{y}_t$ approaches $\mathcal{M}$ as $t\to\infty$. On $\mathcal{E}$ the dynamics of (CTLD) becomes

$$\dot{\boldsymbol{y}}_t=\eta\big(\boldsymbol{w}(\boldsymbol{\sigma}(\bar{\boldsymbol{y}}))-\boldsymbol{y}_t\big)=\eta\big(\bar{\boldsymbol{y}}-\boldsymbol{y}_t\big)$$

so $\boldsymbol{y}_t\to\bar{\boldsymbol{y}}$ as $t\to\infty$. Thus, no other solution except $\bar{\boldsymbol{y}}$ can stay forever in $\mathcal{E}$, and $\mathcal{M}$ consists only of fixed points. Since we assume the system has a finite number of isolated fixed points, so $\boldsymbol{y}_t$ converges to one of them.

2) If the payoff u^i is individually concave, Theorem 1-2) implies that $\bar{\boldsymbol{\pi}}=\boldsymbol{\sigma}(\bar{\boldsymbol{y}})$ is the Nash distribution. By continuity of $\boldsymbol{\sigma}$, we conclude that $\boldsymbol{\pi}_t=\boldsymbol{\sigma}(\boldsymbol{y}_t)$ converges to the Nash distribution.

3) Similarly, the convergence to local Nash distribution is supported by Corollary 1. ∎

VIII. Proof of Theorem 3

We borrow the idea of [20] in NFGs to study the learning process of MGs. Based on the assumptions on $\hat{w}_l$, DTLD is the stochastic approximation of CTLD, and conversely, CTLD is the mean dynamics of DTLD. Let $\underline{\boldsymbol{y}}_t$ be the continuous-time linear interpolation associated to the discrete-time process $\{\boldsymbol{y}_l\}$. With satisfying $\{\alpha_l\}$, the asymptotic behavior of $\{\boldsymbol{y}_l\}$ and $\underline{\boldsymbol{y}}_t$ is the same [43]. Furthermore, under [43, Propositions 4.1 and 4.2], the continuous-time linear interpolation $\underline{\boldsymbol{y}}_t$ of $\{\boldsymbol{y}_l\}$ is a precompact asymptotic pseudotrajectory of the flow associated to (CTLD), and by [43, Th. 5.7], the limit set L of $\underline{\boldsymbol{y}}_t$ is nonempty and an internally chain transitive set of the flow of (CTLD). In Theorem 2, we have proved that CTLD always converges to $\mathcal{E}$, so by [43, Proposition 5.3] and by [50, Proposition 3.27], it follows that L is compact invariant for (CTLD) and $L\subset\mathcal{E}$. Recall that the largest invariant subset of $\mathcal{E}$ consists only of fixed points, so L consists only of fixed points. Thus, for any $\underline{\boldsymbol{y}}_0$, $\underline{\boldsymbol{y}}_t$ converges to one of them as $t\to\infty$, and hence, $\boldsymbol{y}_l$ converges almost surely to one of them as $l\to\infty$.

Appendix E
Details of Algorithm Implementation

A. IBR

IBR runs following the update formula $\pi_{l+1}^i=\beta^i(\boldsymbol{\pi}_l^{-i})$, where the best response oracle uses policy iteration.

B. FP

FP runs following the update formula $\pi_l^i=\beta^i(\tilde{\boldsymbol{\pi}}_l^{-i})$, where $\tilde{\boldsymbol{\pi}}_l^{-i}=(\tilde{\pi}_l^j)_{j\in\mathcal{N}\setminus i}$ and $\tilde{\pi}_l^j=(1/l)(\pi_0^j+\cdots+\pi_{l-1}^j)$ is the fictitious policy that averages the empirical behaviors of player j.

C. PSRO

At the lth iteration, each player has completed its empirical payoff table $U^i[0\colon l-1,0\colon l-1,\ldots,0\colon l-1]$ over the populations containing all players' historical policies. Each element $U^i[l_1,l_2,\ldots,l_n]$ corresponds to player i's payoff under profile $(\pi_{l_1}^1,\pi_{l_2}^2,\ldots,\pi_{l_n}^n)$, where $0\le l_i\le l-1\ \forall i$. The metasolver finds the metastrategy $\mu^i=p_0\pi_0^i+p_1\pi_1^i+\cdots+p_{l-1}\pi_{l-1}^i$ for each player based on the empirical payoff tables $\{U^i\}$. Then, each player uses the oracle to find the best response to the others' metastrategies, $\pi_l^i=\beta^i(\boldsymbol{\mu}^{-i})$, and adds the new policy into the population for the next iteration.

D. ED

At every iteration of ED running, each player calculates its best response μ_l^i to the current profile $\boldsymbol{\pi}_l$, that is, $\mu_l^i=\beta^i(\boldsymbol{\pi}_l^{-i})$. Then, player i evaluates its policy π_l^i against the others' best response $(\mu_l^j)_{j\in\mathcal{N}\setminus i}$ and obtains the advantage $A_{\pi_l^i,\boldsymbol{\mu}_l^{-i}}$. The policy is updated by taking a small step along the advantage

$$\pi_{l+1}^i=\Gamma\Big(\pi_l^i+\alpha_lA_{\pi_l^i,\boldsymbol{\mu}_l^{-i}}\Big)$$

where Γ is a projection to the policy space and α_l is the update step. Note that the above update formula differs from the one

TABLE I
HYPERPARAMETERS OF EPO, SP, NFSP, PSRO, AND PPO ALGORITHMS IN WIMBLEPONG GAME

Hyperparameter	Value
hidden layers	[64, 64]
steps_per_epoch	4000
num_cpu	6
epochs	400
clip_ratio	0.2
pi_lr	0.0003
vf_lr	0.001
train_pi_iters	80
train_v_iters	80
lam	0.97
max_ep_len	600
target_kl	0.01

proposed by Lockhart *et al.* [14] in the selection of update direction. Here, we use advantage while Lockhart *et al.* [14] used a Q value, but they are equivalent in policy gradient direction. In Soccer experiment we vary α_l among 0.001, 0.003, 0.01, and 0.03, and the best $\alpha_l = 0.001$ is selected as representative for comparison in Fig. 3(b). In Cournot Competition, we also try $\alpha_l = 0.003, 0.01, 0.03, 0.1, 0.3$ for ED and choose the result by $\alpha_l = 0.1$ as the best representative in Fig. 4(b).

E. EPO/SP/NFSP/PSRO/PPO

These algorithms are realized in the framework of PPO-Clip implementation released in OpenAI's Spinning Up package (under the MIT License). In Wimblepong experiments, they share the values of hyperparameters listed in Table I.[4] The entropic parameter ϵ in policy loss varies among 0, 0.003, 0.01, 0.03, 0.1, and 0.3. Random seed selects among 100, 200, and 300.

The experiments are conducted on Intel 32 Core E5-2620 CPU and Nvidia 2080TI GPU with 64G RAM.

Payoff or relative performance of two agents is evaluated by setting up 1000 matches between them and calculating the difference of their win rates. Positive payoff indicates the first agent is stronger than the latter one, and negative payoff indicates the opposite. Learned at different random seeds, the multiple solutions of an agent are matched with the multiple solutions of its opponent with uniform probability, in order to reduce random effects.

[4]Meanings of hyperparameters are available in https://spinningup.openai.com/en/latest/algorithms/ppo.html.

REFERENCES

[1] L. S. Shapley, "Stochastic games," *Proc. Nat. Acad. Sci.*, vol. 39, no. 10, pp. 1095–1100, 1953.

[2] V. Hakami and M. Dehghan, "Learning stationary correlated equilibria in constrained general-sum stochastic games," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1640–1654, Jul. 2016.

[3] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 157–163.

[4] J. Perolat, B. Scherrer, B. Piot, and O. Pietquin, "Approximate dynamic programming for two-player zero-sum Markov games," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 1321–1329.

[5] J. Heinrich, M. Lanctot, and D. Silver, "Fictitious self-play in extensive-form games," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 805–813.

[6] A. Celli, A. Marchesi, G. Farina, and N. Gatti, "No-regret learning dynamics for extensive-form correlated equilibrium," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2020, pp. 7722–7732.

[7] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM J. Comput.*, vol. 39, no. 1, pp. 195–259, 2009.

[8] D. S. Leslie and E. J. Collins, "Individual Q-learning in normal form games," *SIAM J. Control Optim.*, vol. 44, no. 2, pp. 495–514, 2005.

[9] P. Coucheney, B. Gaujal, and P. Mertikopoulos, "Penalty-regulated dynamics and robust learning procedures in games," *Math. Oper. Res.*, vol. 40, no. 3, pp. 611–633, 2015.

[10] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[12] Y. Zhu and D. Zhao, "Vision-based control in the open racing car simulator with deep and reinforcement learning," *J. Ambient Intell. Humanized Comput.*, pp. 1–13, Sep. 2019. [Online]. Available: https://doi.org/10.1007/s12652-019-01503-y

[13] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[14] E. Lockhart *et al.*, "Computing approximate equilibria in sequential adversarial games by exploitability descent," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 464–470.

[15] C. Daskalakis, D. J. Foster, and N. Golowich, "Independent policy gradient methods for competitive reinforcement learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2020, pp. 5527–5540.

[16] M. Lanctot *et al.*, "A unified game-theoretic approach to multiagent reinforcement learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4193–4206.

[17] D. Balduzzi *et al.*, "Open-ended learning in symmetric zero-sum games," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 434–443.

[18] P. Muller *et al.*, "A generalized training approach for multiagent learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–35.

[19] P. Mertikopoulos and Z. Zhou, "Learning in games with continuous action sets and unknown payoff functions," *Math. Program.*, vol. 173, nos. 1–2, pp. 465–507, 2019.

[20] B. Gao and L. Pavel, "On passivity, reinforcement learning, and higher order learning in multiagent finite games," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 121–136, Jan. 2021.

[21] M. G. Lagoudakis and R. Parr, "Learning in zero-sum team Markov games using factored value functions," in *Proc. 15th Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 1659–1666.

[22] Y. Zhu and D. Zhao, "Online minimax Q network learning for two-player zero-sum Markov games," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1228–1241, Mar. 2022.

[23] G. Luo, H. Zhang, H. He, J. Li, and F.-Y. Wang, "Multiagent adversarial collaborative learning via mean-field theory," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4994–5007, Oct. 2021.

[24] X. Wang, L. Ke, Z. Qiao, and X. Chai, "Large-scale traffic signal control using a novel multiagent reinforcement learning," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 174–187, Jan. 2021.

[25] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1889–1897.

[26] S. Srinivasan *et al.*, "Actor-critic policy optimization in partially observable multiagent environments," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3426–3439.

[27] S. Omidshafiei *et al.*, "α-rank: Multi-agent evaluation by evolution," *Sci. Rep.*, vol. 9, no. 1, pp. 1–29, 2019.

[28] Y. Zhu, D. Zhao, and X. Li, "Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 714–725, Mar. 2017.

[29] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for H_∞ optimal control of polynomial nonlinear systems via sum of squares programming," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 500–509, Feb. 2018.

[30] Y. Lv, J. Na, X. Zhao, Y. Huang, and X. Ren, "Multi-H_∞ controls for unknown input-interference nonlinear system with reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 7, 2021, doi: 10.1109/TNNLS.2021.3130092.

[31] Q. Zhang and D. Zhao, "Data-based reinforcement learning for nonzero-sum games with unknown drift dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2874–2885, Aug. 2019.

[32] J. Li, J. Ding, T. Chai, and F. L. Lewis, "Nonzero-sum game reinforcement learning for performance optimization in large-scale industrial processes," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 4132–4145, Sep. 2020.
[33] Z. Zhang, J. Xu, and M. Fu, "Q-learning for feedback Nash strategy of finite-horizon nonzero-sum difference games," *IEEE Trans. Cybern.*, early access, Mar. 12, 2021, doi: 10.1109/TCYB.2021.3052832.
[34] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4295–4304.
[35] K. Shao, Y. Zhu, and D. Zhao, "StarCraft micromanagement with reinforcement learning and curriculum transfer learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 1, pp. 73–84, Feb. 2019.
[36] J. Chai *et al.*, "UNMAS: Multiagent reinforcement learning for unshaped cooperative scenarios," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 30, 2021, doi: 10.1109/TNNLS.2021.3105869.
[37] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.
[38] D. Gale, "The game of hex and the Brouwer fixed-point theorem," *Amer. Math. Monthly*, vol. 86, no. 10, pp. 818–827, 1979.
[39] N. Schofield and I. Sened, "Local Nash equilibrium in multiparty politics," *Ann. Oper. Res.*, vol. 109, pp. 193–211, Jan. 2002.
[40] L. J. Ratliff, S. A. Burden, and S. S. Sastry, "Characterization and computation of local Nash equilibria in continuous games," in *Proc. 51st Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2013, pp. 917–924.
[41] H. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ, USA: Prentice Hall, 2002.
[42] O. Vinyals *et al.*, "Grandmaster level in starCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
[43] M. Benaïm, "Dynamics of stochastic approximation algorithms," in *Séminaire de Probabilités XXXIII*, J. Azéma, M. Émery, M. Ledoux, and M. Yor, Eds. Heidelberg, Germany: Springer, 1999, pp. 1–68.
[44] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–14.
[45] V. Naroditskiy and A. Greenwald, "Using iterated best-response to find Bayes-Nash equilibria in auctions," in *Proc. 22nd Nat. Conf. Artif. Intell.*, vol. 2, 2007, pp. 1894–1895.
[46] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[47] T. Raghavan, "Zero-sum two-person games," in *Handbook of Game Theory With Economic Applications*, vol. 2. New Oxford, U.K.: Elsevier, 1994, pp. 735–768.
[48] J. Heinrich and D. Silver, "Deep reinforcement learning from self-play in imperfect-information games," 2016, *arXiv:1603.01121*.
[49] A. Takayama, *Mathematical Economics*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
[50] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions," *SIAM J. Control Optim.*, vol. 44, no. 1, pp. 328–348, 2005.

Yuanheng Zhu (Senior Member, IEEE) received the B.S. degree in automation from Nanjing University, Nanjing, China, in 2010, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015.

From 2015 to 2017, he was an Assistant Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, where he is currently an Associate Professor. From 2017 to 2018, he was a Visiting Scholar with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA. His current research interests include deep reinforcement learning, game theory, game intelligence, and multiagent learning.

Dr. Zhu was the Chair of the IEEE Computational Intelligence Society (CIS) Travel Grant Subcommittee in 2016 and Summer Schools Subcommittee from 2020 to 2021. He currently serves as the Chair for the IEEE CIS Content Creation Subcommittee. He also serves as the Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Weifan Li received the B.S. degree in materials science and engineering from Chongqing University, Chongqing, China, in 2015, and the M.S. degree in automation from Fuzhou University, Fuzhou, Fujian, China, in 2018. He is currently pursuing the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His current research interests include reinforcement learning, deep learning, and game AI.

Mengchen Zhao received the B.S. degree in applied mathematics from Sun Yat-sen University, Guangzhou, China, in 2014, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2019.

He is currently a Senior Researcher with Noah's Ark Laboratory, Huawei, Beijing, China. His research interests include reinforcement learning and adversarial learning.

Jianye Hao received the B.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2008, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2013.

He was a Postdoctoral Fellow with the Massachusetts Institute of Technology, Cambridge, MA, USA, and the Singapore University of Technology and Design, Singapore. He is currently the Director of the Decision Making and Reasoning Laboratory, Noah's Ark Laboratory, Huawei, Beijing, China, and an Associate Professor with Tianjin University, Tianjin, China. He has authored or coauthored over 100 papers on artificial intelligence conferences and journals. His research interests include deep reinforcement learning and multiagent systems.

Dr. Hao has received the Best Paper Award of ASE2019, DAI2019, and CoRL2020.

Dongbin Zhao (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1994, 1996, and 2000, respectively.

He was a Postdoctoral Fellow with Tsinghua University, Beijing, China, from 2000 to 2002. He has been a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, since 2002, and also a Professor with the University of Chinese Academy of Sciences, Beijing. From 2007 to 2008, he was also a Visiting Scholar with the University of Arizona, Tucson, AZ, USA. He has authored or coauthored six books and over 90 international journal articles. His current research interests include deep reinforcement learning, computational intelligence, autonomous driving, game artificial intelligence, robotics, and smart grids.

Dr. Zhao is involved in organizing many international conferences. He was the Chair of the IEEE Computational Intelligence Society, the Adaptive Dynamic Programming and Reinforcement Learning Technical Committee from 2015 to 2016, the Multimedia Subcommittee from 2015 to 2016, the Beijing Chapter from 2017 to 2018, and the Technical Activities Strategic Planning Sub-Committee in 2019. He is the Chair of the Distinguished Lecturer Program. He works as a guest editor for several renowned international journals. He serves as the Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, and *IEEE Computation Intelligence Magazine*.