CFR-MIX: Solving imperfect information extensive-form games with combinatorial action space

Shuxin LI

Youzhi ZHANG

Xinrun WANG
Singapore Management University, xrwang@smu.edu.sg

Wanqi XUE

Bo AN

Citation

CFR-MIX: Solving Imperfect Information Extensive-Form Games with Combinatorial Action Space

Shuxin Li[1] , **Youzhi Zhang**[2*] , **Xinrun Wang**[1*] , **Wanqi Xue**[1] and **Bo An**[1]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Department of Computer Science, Dartmouth College, USA

{shuxin.li, xinrun.wang, boan}@ntu.edu.sg, youzhi.zhang@dartmouth.edu, wanqi001@e.ntu.edu.sg

Abstract

In many real-world scenarios, a team of agents must coordinate with each other to compete against an opponent. The challenge of solving this type of game is that the team's joint action space grows exponentially with the number of agents, which results in the inefficiency of the existing algorithms, e.g., Counterfactual Regret Minimization (CFR). To address this problem, we propose a new framework of CFR: **CFR-MIX**. Firstly, we propose a new strategy representation that represents a joint action strategy using individual strategies of all agents and a consistency relationship to maintain the cooperation between agents. To compute the equilibrium with individual strategies under the CFR framework, we transform the consistency relationship between strategies to the consistency relationship between the cumulative regret values. Furthermore, we propose a novel decomposition method over cumulative regret values to guarantee the consistency relationship between the cumulative regret values. Finally, we introduce our new algorithm CFR-MIX which employs a mixing layer to estimate cumulative regret values of joint actions as a non-linear combination of cumulative regret values of individual actions. Experimental results show that CFR-MIX outperforms existing algorithms on various games significantly.

1 Introduction

Extensive-form games provide a versatile framework capable of representing multiple agents, imperfect information, and stochastic events. Now there are many researches about solving two-player zero-sum extensive-games, such as computing Nash equilibria by linear programs [Shoham and Leyton-Brown, 2008], double oracle algorithms [McMahan *et al.*, 2003; Jain *et al.*, 2011], and CFR [Zinkevich *et al.*, 2008]. Meanwhile, these scalable algorithms have achieved many accomplishments. Double oracle algorithms [McMahan *et al.*, 2003; Jain *et al.*, 2011] have subsequently been applied to real-world attacker-defender scenarios. Heads-up limit

hold'em poker was essentially solved [Bowling *et al.*, 2015]. Later, significant progress has been made for heads-up no-limit hold'em poker based on CFR, such as Libratus [Brown *et al.*, 2017] and DeepStack [Moravčík *et al.*, 2017].

CFR is one of the most popular algorithms to solve imperfect-information extensive-form games which is an iterative algorithm to approximate a Nash equilibrium with repeated self-play between two regret-minimizing algorithms. Some sampling-based CFR variants [Lanctot *et al.*, 2009; Gibson *et al.*, 2012] are proposed to solve large games effectively. Nowadays, neural network function approximation is applied to CFR to solve larger games. Deep CFR [Brown *et al.*, 2019], Single Deep CFR [Steinberger, 2019] and Double Neural CFR [Li *et al.*, 2019] are algorithms aiming to approximate CFR using deep neural networks.

In this paper, we focus on a special type of two-player imperfect-information extensive-form zero-sum games, *Team Adversary Games*, in which a team of cooperative agents plays against an adversary and all agents in the team share one utility function. This game model captures many real-world scenarios, such as many policemen coordinate with each other to catch an attacker [Basilico *et al.*, 2017]. The challenge of solving this type of games is that the size of the team's joint action space is exponential with the number of agents. For example, if the number of agents is 8 and every agent has 10 actions, then the size of team's action space will be 10^8. CFR and its sampling-based variants use the tabular-form to record the joint action strategy which results in impracticality due to the limited memory. For algorithms using deep neural networks, e.g., Deep CFR and Double Neural CFR, it is ineffective to train the strategy network over the large action space. One way to avoid computing the joint action strategy is to let each agent compute its policy independently. However, this approach cannot promote the cooperative interaction between agents in the team and there is no theoretical guarantee to converge to the equilibrium in multiplayer games [Abou Risk *et al.*, 2010]. Therefore, it is impractical to solve this type of game with existing algorithms.

To solve the *exponential combinatorial action space problems* in Team-Adversary Games, we propose a new framework of CFR: **CFR-MIX**. Firstly, inspired by the idea of centralized training with decentralized execution [Rashid *et al.*, 2018], we propose a novel strategy representation that represents the team's joint action strategy using the individual

---

strategy of each agent. We also define a consistency relationship between these two strategy representations to maintain the cooperation among agents in the team, which guarantees that the equilibrium with new strategy representation is a special team-maxmin equilibrium. To compute Nash equilibrium with individual strategy representation under the CFR framework, we transform the consistency relationship between strategies to the consistency relationship between the cumulative regret values of joint actions and the cumulative regret values of individual actions. Furthermore, a novel decomposition method over cumulative regret values is proposed to guarantee the consistency relationship between the cumulative regret values. To implement the decomposition method, CFR-MIX employs a mixing layer that estimates cumulative regret values of joint actions as a non-linear combination of cumulative regret values of individual actions. To further improve the performance, the parameter sharing technique is applied among all agents in the team, which also reduces the network parameters dramatically. Finally, experimental results show that CFR-MIX outperforms state-of-the-art algorithms significantly on games in different domains.

2 Background

2.1 Extensive-Form Game

An Extensive-Form Game (EFG) [Shoham and Leyton-Brown, 2008] can be formulated by a tuple $(N, H, A, P, \mathcal{I}, u)$. $N = \{1, ..., n\}$ is a set of players. H is a set of histories (i.e., the possible action sequences). The empty sequence $\emptyset$ which is the root node of game tree is in H, and every prefix of a sequence in H is also in H. $Z \subset H$ is the set of the terminal histories. $A(h) = \{a : (h, a) \in H\}$ is the set of available actions at a non-terminal history $h \in H$. P is the player function. $P(h)$ is the player who takes an action at the history h, i.e., $P(h) \mapsto P \cup \{c\}$. c denotes the "chance player", which represents stochastic events outside of the players' control. If $P(h) = c$ then chance determines the action taken at history h. Information sets $\mathcal{I}_i$ form a partition over histories h where player $i \in \mathcal{N}$ takes action. Therefore, every information set $I_i \in \mathcal{I}_i$ corresponds to one decision point of player i which means that $P(h_1) = P(h_2)$ and $A(h_1) = A(h_2)$ for any $h_1, h_2 \in I_i$. For convenience, we use $A(I_i)$ to represent the set $A(h)$ and $P(I_i)$ to represent the player $P(h)$ for any $h \in I_i$. For each player $i \in N$, a utility function is a mapping $u_i : Z \to \mathbb{R}$.

A player's behavior strategy σ_i is a function mapping every information set of player i to a probability distribution over $A(I_i)$. A strategy profile σ consists of a strategy for each player $\sigma_1, \sigma_2, ...,$ with σ_{-i} referring to all the strategies in σ except σ_i. Let $\pi^\sigma(h)$ be the reaching probability of history h if players choose actions according to σ. Given a strategy profile σ, the overall value to player i is the expected payoff of the resulting terminal node, $u_i(\sigma) = \sum_{h \in Z} \pi^\sigma(h) u_i(h)$.

In this paper, we consider a type of imperfect-information extensive-form games, *Team-Adversary Game*, that consists of an adversary $\mathbb{V}$ and one team $\mathbb{T}$. The team is formed by multiple agents $R = \{1, ..., n\}$. Every agent i has its own available action set $A_{\mathbb{T}_i}$. Therefore, the team's action set includes all joint actions that are the combinations of all agents'

individual actions, i.e., $A_{\mathbb{T}} = \times_{i \in R} A_{\mathbb{T}_i}$. Here, we focus on zero-sum games, i.e., $u_{\mathbb{T}}(z) = -u_{\mathbb{V}}(z)$ for $\forall z \in Z$. Nash equilibrium [Nash, 1950] is adopted as our solution concept which is a strategy profile such that no player can get more reward by switching to a different strategy unilaterally. Formally, in Team-Adversary Games, a Nash Equilibrium (NE) is a strategy profile σ where

$$u_{\mathbb{V}(\mathbb{T})}(\sigma) \geq \max_{\sigma'_{\mathbb{V}(\mathbb{T})} \in \sum_{\mathbb{V}(\mathbb{T})}} u_{\mathbb{V}(\mathbb{T})}(\sigma'_{\mathbb{V}(\mathbb{T})}, \sigma_{\mathbb{T}(\mathbb{V})})$$

An approximation of a Nash Equilibrium or ϵ-Nash equilibrium is a strategy profile σ where

$$u_{\mathbb{V}(\mathbb{T})}(\sigma) + \epsilon \geq \max_{\sigma'_{\mathbb{V}(\mathbb{T})} \in \sum_{\mathbb{V}(\mathbb{T})}} u_{\mathbb{V}(\mathbb{T})}(\sigma'_{\mathbb{V}(\mathbb{T})}, \sigma_{\mathbb{T}(\mathbb{V})})$$

2.2 Counterfactual Regret Minimization

CFR is a family of iterative algorithms that are the most popular approach to approximately solve large imperfect-information games [Zinkevich et al., 2008]. To define the concept, first consider repeatedly playing games. Let σ_i^t be the strategy used by player i on round t. We define $u_i(\sigma, h)$ as the expected utility of player i given that the history h is reached and then all players play according to strategy σ from that point on. Let's define $u_i(\sigma, h \cdot a)$ as the expected utility of player i given that the history h is reached and then all players play according to strategy σ except that player i selects action a in the history h. Formally, $u_i(\sigma, h) = \sum_{z \in Z} \pi^\sigma(h, z) u_i(z)$ and $u_i(\sigma, h \cdot a) = \sum_{z \in Z} \pi^\sigma(h \cdot a, z) u_i(z)$.

The *counterfactual value* $u_i^\sigma(I)$ is the expected value of an information set I given that player i tries to reach it. This value is the weighted average of the value of each history in an information set. The weight is proportional to the contribution of all players other than i to reach each history. Thus, $u_i^\sigma(I) = \sum_{h \in I} \pi_{-i}^\sigma(h) \sum_{z \in Z} \pi^\sigma(h, z) u_i(z)$.

For any action $a \in A(I)$, the counterfactual value of an action a is $u_i^\sigma(I, a) = \sum_{h \in I} \pi_{-i}^\sigma(h) \sum_{z \in Z} \pi^\sigma(h \cdot a, z) u_i(z)$. The *instantaneous regret* for action a in information set I on iteration t is $r^t(I, a) = u_{P(I)}^{\sigma^t}(I, a) - u_{P(I)}^{\sigma^t}(I)$. The cumulative regret for action a in I on iteration T is $R^T(I, a) = \sum_{t=1}^T r^t(I, a)$. In CFR, players use *Regret Matching* to pick a distribution over actions in an information set in proportion to the positive cumulative regret on those actions. Formally, on iteration $T+1$, player i selects actions $a \in A(I)$ according to probability

$$\sigma^{T+1}(I, a) = \begin{cases} \frac{R_+^T(I, a)}{\sum_{b \in A(I)} R_+^T(I, b)} & \text{if } \sum_{b \in A(I)} R_+^T(I, b) > 0 \\ \frac{1}{|A(I)|} & \text{otherwise} \end{cases}$$

where $R_+^T(I, a) = \max(R^T(I, a), 0)$ because we often mostly concern about cumulative regret when it is positive. If a player plays according to CFR in every iteration, then on iteration T, $R^T(I) \leq \Delta_i \sqrt{|A_i|}\sqrt{T}$ where $\Delta_i = \max_z u_i(z) - \min_z u_i(z)$ is the range of utility of player i. Moreover, $R_i^T \leq \sum_{I \in \mathcal{I}_i} R^T(I) \leq |\mathcal{I}_i| \Delta_i \sqrt{|A_i|}\sqrt{T}$. Therefore, as $T \to \infty$, $\frac{R_i^T}{T} \to 0$. In two-player zero-sum games, if both players' average regret $\frac{R_i^T}{T} \leq \epsilon$, then their average strategies $(\bar{\sigma}_1^T, \bar{\sigma}_2^T)$ form a 2ϵ-equilibrium [Waugh et al., 2009].

To solve large games effectively, some sampling-based CFR algorithms are proposed, such as external sampling, outcome sampling [Lanctot *et al.*, 2009], probe sampling [Gibson *et al.*, 2012] and other reduce variance sampling algorithms [Schmid *et al.*, 2019; Steinberger *et al.*, 2020]. External sampling algorithm has the lowest variance. However, it runs slowly because it traverses all actions of one player every iteration. Although outcome sampling algorithm runs faster than external sampling algorithm, it has large variance which influences the convergence rate. Compared to outcome sampling algorithm, probe sampling algorithm attempts to reduce variance by replacing "zeroed-out" counterfactual values of non-sampled actions with closer estimates of the true counterfactual values. In this paper, we adopt the probe sampling algorithm to traverse the game tree to collect regret value data.

3 Strategy Representation

In this paper, we focus on solving the Team-Adversary Games in which a team of agents plays against an adversary and all agents share one utility function. Note that the team's joint action space grows exponentially with the number of team players. Thus, it is impractical to solve this type of games with existing approaches due to the large action space. To address this problem, we propose a novel strategy representation, which significantly reduces the strategy space.

3.1 Individual Strategy Representation

In this section, we introduce a new strategy representation to denote the team's joint action strategy using the individual action strategies of all the team players. Different from the exponential combinatorial action space, the new individual action space of all the agents is linear with the number of agents. It is worth noting that all agents share the same history of the team $\mathbb{T}$, i.e., the player function can be extended as $P(h) = \mathbb{T} = \{1, 2, 3, ..., n\}$. In other words, all the agents share the same information sets of the team $\mathbb{T}$. Therefore, the new strategy representation of the team is defined as $f_{\mathbb{T}} = (\sigma_1, \sigma_2, ..., \sigma_n)$, where σ_i is agent i's strategy which maps each information set of player $\mathbb{T}$ to a probability distribution over agent i's action space $A_{\mathbb{T}_i}$. For clarity, we call this new representation as *individual strategy representation* and denote the old team strategy representation by *joint strategy representation*. Given a strategy profile $\sigma = (\sigma_{\mathbb{V}}, f_{\mathbb{T}}) = (\sigma_{\mathbb{V}}, \sigma_1, ..., \sigma_n)$, the expected payoff of every agents is the same one $u_i(\sigma) = \sum_{h \in Z} u_{\mathbb{T}}(h) \pi^{\sigma}(h)$. $\pi^{\sigma}(h)$ is the reaching probability of history h if players choose actions according to σ which means that each team agent i chooses its action according to its strategy σ_i independently.

3.2 Strategy Consistency

Despite that the individual strategy representation reduces the search space efficiently, it cannot represent all join action strategy space. To clarify the difference between these two strategy representations, we first propose a consistency relationship between the two strategy representations which can be defined as

$$\boldsymbol{\sigma}_{\mathbb{T}}(I, \boldsymbol{a}) = \sigma_1(I, a_1)\sigma_2(I, a_2)...\sigma_n(I, a_n), \quad (1)$$
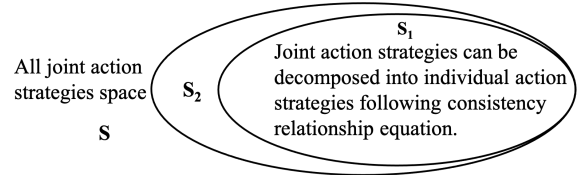

Figure 1: The relationship between joint action strategy space and individual action strategy space

where $\boldsymbol{\sigma}_{\mathbb{T}}(\cdot)$ and $\sigma_i(\cdot)$ are the probability of a joint action and the probability of an individual action, respectively. In short, the joint strategy representation can be decomposed into the individual strategy representation. Note that the consistency relationship divides the whole joint action space S into two parts S_1, S_2 (Fig. 1). We set the strategy space of adversary as S_V and (S, S_V) forms a two-player zero-sum game. Under our new strategy representation, the strategy space of the team becomes S_1, and (S_1, S_V) forms a two-player zero-sum game with the consistency relationship holds which is the game we focus on in this paper. The difference between these two games is that in game (S_1, S_V), each team's joint strategy can be decomposed into the agents' individual strategies. Obviously, some joint strategies cannot be decomposed (S_2). For example, a joint action strategy $\boldsymbol{\sigma} = (0.5, 0, 0, 0.5)$ is based on the joint action set $\{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$. We cannot get the corresponding individual strategy representation following Eq. (1). Therefore, if the team's NE strategy of (S, S_V) is in the strategy space S_1, the NE of (S_1, S_V) equals the NE of (S, S_V) (See Theorem 1). Otherwise, the team's NE strategy of (S_1, S_V) is a special team-maxmin equilibrium strategy that can be decomposed into individual strategy which is different from the NE of (S, S_V). In this case, the team may lose some utility after reducing strategy space since the utility of the team's joint strategy is always not lower than the utility of the team's strategy which can be decomposed. However, experimental results show that solving the game (S_1, S_V) can get good strategies faster than solving the large game (S, S_V). The reason could be that the exponential action space influences the computing process of the join action strategy significantly.

Theorem 1. *Under the new strategy representation, the Nash Equilibrium strategy profile between the adversary and the team keeps unchanged if Eq. (1) holds for the team's Nash equilibrium strategy.* [1]

4 CFR-MIX

Now we introduce a novel algorithm CFR-MIX to compute an NE for the Team-Adversary game with strategy consistency (S_1, S_V). Firstly, we introduce the consistency relationship between cumulative regret values which is transformed from Eq. (1). Secondly, a novel decomposition method over cumulative regret values is proposed based on the consistency relationship. Then, we introduce our algorithm CFR-MIX in which the decomposition method is implemented by a mixing layer. Finally, we provide the theoretical analysis.

---

[1]The complete proof is in https://arxiv.org/pdf/2105.08440.pdf

4.1 Regret Consistency

For extensive-form games, one of the most popular algorithms to compute NEs is CFR-based algorithm which leverages regret-matching to compute strategy according to cumulative regret values. In other words, the probability of each action is related to the cumulative regret value of the action. Therefore, to compute the individual strategy for each agent, we need to know the cumulative regret values of its actions explicitly. However, all the agents share the same team utility function and we can only get the regret values of joint actions. To this end, we propose a decomposition method over cumulative regret values. Before describing the decomposition method, we first transform the consistency relationship between strategies into the consistency relationship between cumulative regret values which can be defined as follows:

$$\forall \boldsymbol{a}, \quad \boldsymbol{\pi}_{\mathbb{T}}(R_{tot}(I, \boldsymbol{a})) = \prod_{i=1}^{n} \pi_i(R_i(I, a_i)), \quad (2)$$

where $\pi_i(\cdot)$ and $\boldsymbol{\pi}_{\mathbb{T}}(\cdot)$ are the probabilities of individual action and joint action, i.e., $\pi_i(R(I, a)) = \sigma_i(I, a)$. $\boldsymbol{a} = (a_1, ..., a_n)$ is the joint action composed by every agent's action. R_{tot} and R_i are the cumulative regret values of a joint action and an individual action, respectively. Since these probabilities are computed using regret-matching based on cumulative regret value, we can reformulate this consistency relationship as

$$\frac{R_{tot}(I, \boldsymbol{a})_+}{\sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}')_+} = \prod_{i=1}^{n} \frac{R_i(I, a_i)_+}{\sum_{b \in A_i(I)} R_i(I, b)_+} \quad (3)$$

when $\sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}')_+ > 0$ and $\sum_{b \in A_i(I)} R_i(I, b)_+ > 0$. If $\sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}')_+ \leq 0$, the cumulative regret values of individual actions need to satisfy: $\sum_{b \in A_i(I)} R_i(I, b)_+ \leq 0$, $\forall i \in \{1, ..n\}$. The consistency relationship guarantees that the probability of a joint action equals the product of the probabilities of all individual actions selected by each player.

4.2 Product-Form Decomposition

To guarantee these consistency relationships (Eqs. (1-3)), we propose a product-form decomposition method over cumulative regret values which can be defined as follows:

$$\forall \boldsymbol{a}, \quad R_{tot}(I, \boldsymbol{a}) = \prod_{i=1}^{n} R_i(I, a_i). \quad (4)$$

Here, we set $R_{tot} = 0$ if $R_{tot} \leq 0$ following the setting of regret-matching+ [Tammelin *et al.*, 2015] which is a regret-minimizing algorithm that operates very similarly to regret-matching. Compare with the regret matching which ignores actions that have an cumulative negative regret, regret-matching+ actively resets any cumulative negative regret back to zero. In the remaining paper, we consider that all cumulative regret values are non-negative.

Theorem 2. *If product-form decomposition (Eq. (4)) holds, the consistency relationship between the joint action strategy and the individual strategy (Eq. (1)) can be guaranteed.*

Proof. When $\sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}') > 0$, we can get

$$\sigma(I, \boldsymbol{a}) = \frac{R_{tot}(I, \boldsymbol{a})}{\sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}')} = \frac{\prod_{i=1}^{n} R_i(I, a_i)}{\sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}')}, \quad (5)$$
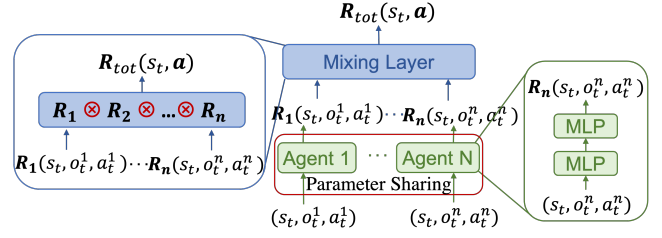

Figure 2: Architecture of team's cumulative regret neural network

$$\prod_{i=1}^{n} \sigma(I, a_i) = \prod_{i=1}^{n} \frac{R_i(I, a_i)}{\sum_{a_i^j \in A_i(I)} R_i(I, a_i^j)} \quad (6)$$

$$= \frac{\prod_{i=1}^{n} R_i(I, a_i)}{\prod_{i=1}^{n} \sum_{a_i^j \in A_i(I)} R_i(I, a_i^j)}.$$

These equations are derived by the regret matching+ and product-form decomposition equation.

$$\prod_{i=1}^{n} \sum_{a_i^j \in A_i(I)} R_i(I, a_i^j) \quad (7)$$

$$= \prod_{i=1}^{n} [R_i(I, a_i^1) + R_i(I, a_i^2) + ... + R_i(I, a_i^{|A_i(I)|})]$$

$$= \prod_{i=1}^{n} R_i(I, a_i^1) + \prod_{i=1}^{n-1} R_i(I, a_i^1) R_{n-1}(I, a_{n-1}^2) +$$

$$... + \prod_{i=1}^{n} R_i(I, a_i^{|A_i(I)|})$$

$$= R_{tot}(I, \boldsymbol{a}_1) + ... + R_{tot}(I, \boldsymbol{a}_{|A_1(I)|...|A_n(I)|})$$

$$= \sum_{\forall \boldsymbol{a}'} R_{tot}(I, \boldsymbol{a}').$$

The above derivation is following the rules of polynomial multiplication and product-form decomposition equation. Therefore, the Eq. (1) holds. □

4.3 Mixing Layer

This section introduces our algorithm, CFR-MIX, which employs a mixing layer to implement the product-form decomposition under the Double Neural CFR framework. The reasons that we do not implement decomposition in tabular CFR framework are described as follows. If we use tabular CFR to compute every cumulative regret value of individual action according to Eq. (4), we will get a non-linear equation which is difficult to be solved efficiently. For example, in a Team-Adversary Game, there are two agents $R = \{1, 2\}$ in the team. At an information set $I_{\mathbb{T}}$, agent 1 has two actions $\{a_{11}, a_{12}\}$ and agent 2 has two actions $\{a_{21}, a_{22}\}$. We only know the cumulative regret values of all joint actions $R_{tot}(I, \boldsymbol{a})$, where $\boldsymbol{a} \in \{(a_{11}, a_{21}), (a_{11}, a_{22}), (a_{12}, a_{21}), (a_{12}, a_{22})\}$. Then we need to compute the cumulative regret value of each individual action following the equations:

$$\begin{cases} R_{tot}(I, (a_{11}, a_{21})) = & R_1(I, a_{11}) \cdot R_2(I, a_{21}) \\ R_{tot}(I, (a_{11}, a_{22})) = & R_1(I, a_{11}) \cdot R_2(I, a_{22}) \\ R_{tot}(I, (a_{12}, a_{21})) = & R_1(I, a_{12}) \cdot R_2(I, a_{21}) \\ R_{tot}(I, (a_{12}, a_{22})) = & R_1(I, a_{12}) \cdot R_2(I, a_{22}) \end{cases}$$

Solving the above equations is a resource intensive task, especially when the game has a large number of agents. Therefore, we adopt to combine CFR with deep neural networks and utilize the great capabilities of neural networks in function approximation properties (e.g., Double Neural CFR).

Our CFR-MIX algorithm is built on the Double Neural CFR framework, considering it can represent cumulative regret explicitly. Specifically, CFR-MIX uses the Probe sampling algorithm to traverse the game tree to collect training data (the data of cumulative regret values and strategies). In the sampling process, when it turns the team player to act, every agent plays according to its strategy and composes their strategies into the joint action strategy which is used to compute regret values only. Because all agents share one team utility function, it computes regret values for all joint actions based on the joint action strategy and joint action values. Then, CFR-MIX trains one cumulative regret value network and one average strategy network for each player.

The key to our method is the new cumulative regret network for the team which is shown in Fig. 2. To decompose the global cumulative regret values of joint actions, we design a mixing layer as a product operation on the cumulative regret values of all individual actions. The Agent network takes the information sets, agent i's observations and agent i's available actions as input and outputs the cumulative regret value of the agent i's actions. The mixing layer takes all the outputs of the Agent network as input and outputs the cumulative regret value of the corresponding joint action. In this way, the mixing layer guarantees that the consistency between cumulative regret values of individual action and joint action. Therefore, the mixing layer can also guarantee the strategy consistency relationship. When training this cumulative regret network, we use the cumulative regret values of joint actions which are the sum of: (1) the regret values obtained from the sampling process and (2) the estimated cumulative regret values obtained from the cumulative regret network in the last iteration.

To get the individual strategy of each agent, we only use the Agent network to estimate the cumulative regret value for each agent's action and compute the agent's strategy using regret-matching+ based on the estimated cumulative regret value. Therefore, CFR-MIX only needs to traverse the action spaces of all the agents which are linear with the number of agents. On the contrary, to get the joint strategy, Double Neural CFR needs to traverse the whole joint action space which is exponential with the number of agents. Furthermore, in CFR-MIX, we adopt the parameter sharing technique which means that all agents share the same Agent network. The parameter sharing can reduce the parameters and improve the performance significantly. For the average strategy network, we also only use one network for all agents. The training data of the strategy network are all agents' strategies. However, in the Double Neural CFR algorithm, the training data of the average strategy network are the team strategies defined over all the joint actions.

4.4 Convergence Analysis

Recall that we focus on a subset of joint action strategy space which can be decomposed into individual action strategies.

Therefore, we compute a new equilibrium of game (S_1, S_V) with proposed CFR-MIX and provide a bound of regret under mild conditions based on Deep CFR [Brown et al., 2019].

Theorem 3. *Let T denote the number of CFR-MIX iterations, $|A|$ the maximum number of actions at any infoset and K the number of traversals per iteration. Let $L_{\mathcal{R}}^t$ be the average MSE loss for $\mathcal{R}_p(I, a|\theta^t)$ on a sample in $M_{r,p}$ at iteration t, and let $L_{\mathcal{R}*}^t$ be the minimum loss achievable for any function $\mathcal{R}$. Let $L_{\mathcal{R}}^t - L_{\mathcal{R}*}^t \le \epsilon_L$. If the value memories are sufficiently large and $\;$ **Eq. (4) holds**, then with probability $1 - \rho$ total regret of player p at time T is bounded by*

$$R_p^T \le (1 + \frac{\sqrt{2}}{\sqrt{\rho K}})\Delta|I_p|\sqrt{|A|}\sqrt{T} + 4T|I_p|\sqrt{|A|\Delta\epsilon_L}$$

As $T \to \infty$, the average regret $\frac{R_p^T}{T}$ is bounded by $4|\mathcal{I}_p|\sqrt{|A|\Delta\epsilon_L}$ with high probability.

Proof. Compared with the proof of Deep CFR, the only difference is that the agent in the team plays the individual strategy instead of the joint action strategy. The establishment of Eq. (4) guarantees the consistency relationship of strategies (Eq. (1)) which means that under the new strategy representation, the Nash Equilibrium keeps unchanged (Theorem 1). Therefore, the regret bound for the new representation is same as the joint strategy representation.[2] □

5 Evaluation

To verify the effectiveness of CFR-MIX, we deploy Double Neural CFR and Deep CFR as baselines and compare these baselines with CFR-MIX and Deep CFR with mixing layer. We also compare CFR-MIX with the algorithm that every agent learns its strategy independently (Individual Deep CFR). Furthermore, to verify the effectiveness of the parameter sharing technique, we evaluate the performance of CFR-MIX with and without parameter sharing. Without loss of generality, we select two different domain games. Experiments are performed on a server with a 10 core 3.3GHz Intel i9-9820X CPU and a NVIDIA RTX 2080 Ti GPU.

5.1 Goofspiel

Goofspiel [Ross, 1971] is a bidding card game where players have a hand of cards numbered 1 to K, and take turns secretly bidding on the top point-valued card in a point card stack using the cards in their hands (Fig. 3(a)). In each bidding process, the player with the higher point of the card will get the point of the top point-valued card. Finally, the player with the highest total points wins. Here, we use an imperfect informational version [Lisỳ et al., 2015; Brown and Sandholm, 2019] which is widely used: players can only observe the results of each bid while cannot know the cards used to bid. Meanwhile, we consider the special two-player case in which several players form a team to play against an opponent. If one of the team players wins, the whole team wins. Here, as shown in Fig. 3, we use different number of cards (6C, 10C and 13C) and team players (4P and 6P). Note that Double Neural CFR + MIX refers to CFR-
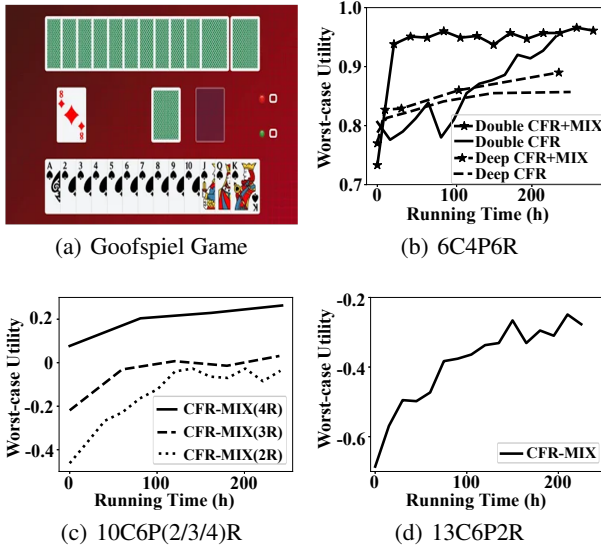
---

[2]The complete proof is in https://arxiv.org/pdf/2105.08440.pdf

(a) Goofspiel Game

(b) 6C4P6R

(c) 10C6P(2/3/4)R

(d) 13C6P2R

Figure 3: Goofspiel Games (C: card, P: team player, R: round)

(a) 3*3 grid, 1 vs 2

(b) 5*5 grid, 1 vs 4

(c) 15*15 grid, 1 vs 4

(d) 15*15 grid, 1 vs 8

Figure 4: NEST Games

MIX and Deep CFR + MIX denotes the Deep CFR algorithm with mixing layer. We can see that the algorithms with mixing layer outperform the others in 6C4P6R game. In 10C6P and 13C6P games, the joint action space is very large at some information sets (about 10^6 and 13^6), and only CFR-MIX can get satisfactory strategies in a limited time.

5.2 Pursuit-Evasion game

Pursuit-evasion games appear in many scenarios in robotics and security domains, where a team of pursuers aims to capture the evader, while the evader aims for the opposite [Horák and Bošanský, 2016]. Here we consider a more realistic version of the pursuit-evasion game, *NEtwork purSuiT game (NEST)* [Zhang *et al.*, 2017; Zhang *et al.*, 2019], which takes the tracking devices into consideration. Thus, pursuers can get the real-time location of the evader. We set a limited number of steps in which if the evader cannot escape to the exits or at least one pursuer catches the evader, the game ends and the defender gets a unit reward. Otherwise, the defender gets a zero payoff. Our experiments are conducted on grid networks in which each node connects to its neighbors. We choose some nodes on the edges as exits. The initial locations of the evader and all pursuers are chosen randomly. As shown in Fig. 4(a-c), the algorithms with the mixing layer outperform other algorithms. In Fig. 4(a), we also compare our algorithm with the algorithm that every pursuer uses Double Neural CFR independently. The result shows that Individual Double Neural CFR cannot converge to NE in a limited time. To verify the effectiveness of parameter sharing, we conduct an ablation experiment in which every pursuer uses different Agent neural networks. The results in Fig. 4(b) indicate that CFR-MIX with parameter sharing performs better than that without parameter sharing (PS). In Fig. 4(c), we only compare CFR-MIX with Double Neural CFR because Deep CFR is time-consuming and needs large memory. We can see that CFR-MIX still exhibits better performance. In
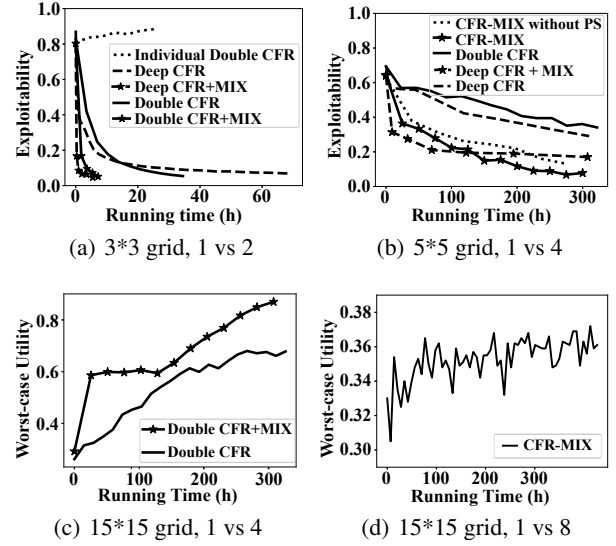
complicated games with large number of pursuers (Fig. 4(d), although CFR-MIX does not converge in a limited time, it still shows the tendency of growth which means that CFR-MIX get a better strategy than random strategy (initial strategy). In conclusion, for different domains games, CFR-MIX outperforms other algorithms significantly. In large games with many players, CFR-MIX can still get some satisfactory results within a limited time while other algorithms cannot solve these large games due to the combinatorial action space.

6 Conclusion

In this paper, we propose a novel framework to solve the Team-Adversary Games with the combinatorial action space. Instead of using the joint action strategy representation, we represent the team's strategy with a novel individual strategy representation to reduce the strategy space. To maintain the cooperation among team players, we define a consistency relationship between these two strategy representations. Moreover, to compute the equilibrium with new strategy representation, we transform the consistency relationship between strategies to the consistency relationship between cumulative regret values. Finally, under the guarantee of these consistency, we propose the CFR-MIX framework which employs a mixing layer to implement the novel decomposition method over cumulative regret values. Experimental results show that our algorithm significantly outperforms other state-of-the-art CFR-based algorithms.

References

[Abou Risk *et al.*, 2010] Nicholas Abou Risk, Duane Szafron, et al. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *AAMAS*, pages 159–166, 2010.

[Basilico *et al.*, 2017] Nicola Basilico, Andrea Celli, Giuseppe De Nittis, and Nicola Gatti. Coordinating multiple defensive resources in patrolling games with alarm systems. In *AAMAS*, pages 678–686, 2017.

[Bowling *et al.*, 2015] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit Hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

[Brown and Sandholm, 2019] Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *AAAI*, pages 1829–1836, 2019.

[Brown *et al.*, 2017] Noam Brown, Tuomas Sandholm, and Strategic Machine. Libratus: The superhuman AI for no-limit poker. In *IJCAI*, pages 5226–5228, 2017.

[Brown *et al.*, 2019] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *ICML*, pages 793–802, 2019.

[Gibson *et al.*, 2012] Richard Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. Generalized sampling and variance in counterfactual regret minimization. In *AAAI*, pages 1355–1361, 2012.

[Horák and Bošanskỳ, 2016] Karel Horák and Branislav Bošanskỳ. Dynamic programming for one-sided partially observable pursuit-evasion games. *arXiv preprint arXiv:1606.06271*, 2016.

[Jain *et al.*, 2011] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *AAMAS*, pages 327–334, 2011.

[Lanctot *et al.*, 2009] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. *NeurIPS*, pages 1078–1086, 2009.

[Li *et al.*, 2019] Hui Li, Kailiang Hu, Shaohua Zhang, Yuan Qi, and Le Song. Double neural counterfactual regret minimization. In *ICLR*, 2019.

[Lisỳ *et al.*, 2015] Viliam Lisỳ, Marc Lanctot, and Michael Bowling. Online monte carlo counterfactual regret minimization for search in imperfect information games. In *AAMAS*, pages 27–36, 2015.

[McMahan *et al.*, 2003] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *ICML*, pages 536–543, 2003.

[Moravčík *et al.*, 2017] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[Nash, 1950] John F Nash. Equilibrium points in n-person games. *PNAS*, 36(1):48–49, 1950.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*, pages 4295–4304. PMLR, 2018.

[Ross, 1971] Sheldon M. Ross. Goofspiel — the game of pure strategy. *Journal of Applied Probability*, 8(3):621–625, 1971.

[Schmid *et al.*, 2019] Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, and Michael Bowling. Variance reduction in monte carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In *AAAI*, volume 33, pages 2157–2164, 2019.

[Shoham and Leyton-Brown, 2008] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[Steinberger *et al.*, 2020] Eric Steinberger, Adam Lerer, and Noam Brown. Dream: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410*, 2020.

[Steinberger, 2019] Eric Steinberger. Single deep counterfactual regret minimization. *arXiv preprint arXiv:1901.07621*, 2019.

[Tammelin *et al.*, 2015] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit Texas hold'em. In *IJCAI*, pages 645–652, 2015.

[Waugh *et al.*, 2009] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *AAMAS*, pages 781–788, 2009.

[Zhang *et al.*, 2017] Youzhi Zhang, Bo An, Long Tran-Thanh, Zhen Wang, Jiarui Gan, and Nicholas R Jennings. Optimal escape interdiction on transportation networks. In *IJCAI*, pages 3936–3944, 2017.

[Zhang *et al.*, 2019] Youzhi Zhang, Qingyu Guo, Bo An, Long Tran-Thanh, and Nicholas R Jennings. Optimal interdiction of urban criminals with the aid of real-time information. In *AAAI*, pages 1262–1269, 2019.

[Zinkevich *et al.*, 2008] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *NeurIPS*, pages 1729–1736, 2008.