Signal Instructed Coordination in Cooperative Multi-agent Reinforcement Learning

Liheng Chen[1], Hongyi Guo[1], Yali Du[2], Fei Fang[3], Haifeng Zhang[4], Weinan Zhang[1(✉)], and Yong Yu[1]

[1] Shanghai Jiao Tong University, Shanghai, China
wnzhang@sjtu.edu.cn
[2] United Kingdom University College London, London, UK
[3] Carnegie Mellon University, Pittsburgh, USA
[4] Institute of Automation, Chinese Academy of Sciences, Beijing, China

Abstract. In many real-world problems, a team of agents need to collaborate to maximize the common reward. Although existing works formulate this problem into a centralized learning with decentralized execution framework, their decentralized execution paradigm limits the agents' capability to coordinate. Inspired by the concept of correlated equilibrium, we propose to introduce a *coordination signal* to address this limitation, and theoretically show that following mild conditions, decentralized agents with the signal can coordinate their individual policies as manipulated by a centralized controller. To encourage agents to learn to exploit the coordination signal, we propose *Signal Instructed Coordination* (SIC), a novel coordination module that can be integrated with most existing MARL frameworks. Our experiments show that SIC consistently improves performance in both matrix games and popular testbeds with high-dimensional strategy space.

Keywords: Multi-agent learning · Reinforcement learning · Correlated equilibrium

1 Introduction

Multi-agent interactions are common in real-world scenarios such as traffic control [24] and smartgrid management [27]. A straightforward approach to solve cooperative multi-agent environments is the *fully centralized* paradigm, where a centralized controller is used to make decisions for all agents, and its policy is learned by applying successful single-agent RL algorithms. However, the fully centralized method suffers from exponential growth of the size of the joint action space with the number of agents. Therefore, decentralized execution approaches are proposed, including the *fully decentralized* paradigm and the *centralized training with decentralized execution* (CTDE) [22,25] paradigm. The fully decentralized method models each participant as an individual agent with its own policy and critic

J. Chen et al. (Eds.): DAI 2021, LNAI 13170, pp. 185–205, 2022.
https://doi.org/10.1007/978-3-030-94662-3_12

conditioned on local information. This setting fails to solve the non-stationary environment problem [16,23], and is empirically deprecated by [8,20]. In CTDE framework, agents can leverage global information including the joint observations and actions of all agents in the training stage, e.g., through training a centralized critic, but the policy of an agent can only be dependent on the individual information and thus they can behave in the decentralized way in the execution stage. This training paradigm bypasses the non-stationary problem, and can lead to some coordination among the cooperative agents empirically [22].

Despite the merits of CTDE, the feasible joint policy space with distributed execution is much smaller than the joint policy space with a centralized controller, limiting the agents' capability to coordinate. For example, in a two-agent traffic system with agents A and B, whose individual action space is {go, stop}, we cannot find a joint policy that satisfies P(A goes & B stops) = P(A stops & B goes) = 0.5 if both agents are making decisions independently. Previous works [26,28] adopts peer-to-peer communication mechanism to facilitate coordination, but they require specially designed communication channels to exchange information and the agents' capability to coordinate is limited by the accessibility and the bandwidth of the communication channel.

Inspired by the *correlated equilibrium* (CE) [1,17] concept in game theory, we introduce a *coordination signal* to allow for more correlation of individual policies and to further facilitate coordination among cooperative agents in decentralized execution paradigms. The coordination signal is conceptually similar to the signal sent by a correlation device to induce CE. It is sampled from a distribution at the beginning of each episode of the game and carries no state-dependent information. After observing the same signal, different agents learn to take corresponding individual actions to formulate an optimal joint action. Such coordination signal is of practical importance. For example, the previous traffic system example can introduce a traffic policeman that sends a public signal via his pose to each agent. The type of the pose may be dependent on the current time (state-free) as a traffic light is, but agents can still coordinate their actions without any explicit communication among them. In addition, we prove that for a group of fully cooperative agents, if the signal's distribution satisfies some mild conditions, the joint policy space is equal to the centralized joint policy space. Therefore, the coordination signal expands the joint policy space while still maintains the decentralized execution setting, and is helpful to find a better joint policy.

To incentivize agents to make full use of the coordination signal, we propose *Signal Instructed Coordination* (SIC), a novel plug-in module for learning coordinated policies. In SIC, a continuous vector is sampled from a pre-defined normal distribution as the coordination signal, and every agent observes the vector as an extra input to its policy network. We introduce an information-theoretic regularization, which maximizes the mutual information between the signal and the resulting joint policy. We implement a centralized neural network to optimize the variational lower bound [2,5,18] of the mutual information. The effects of optimizing this regularization are three-fold: it (i) encourages each agent to align its individual policy with the coordination signal, (ii) decreases the uncertainty of policies of other agents to alleviate the difficulty to coordinate, and (iii) leads to a

more diverse joint policy. Besides, SIC can be easily incorporated with most models that follow the decentralized execution paradigm, such as MADDPG [22] and COMA [9].

To evaluate SIC, we first conduct insightful experiments on a multiplayer variant of matrix game *Rock-Paper-Scissors-Well* to demonstrate how SIC incentivize agents to coordinate in both one-step and multi-step scenarios. Then we conduct experiments on *Cooperative Navigation* and *Predator-Prey*, two classic games implemented in multi-agent particle worlds [22]. We empirically show that by adopting SIC, agents learn to coordinate by interpreting the signal differently and thus achieve better performance. Besides, the visualization of the distribution of collision positions in *Predator-Prey* provides evidence that SIC improves the diversity of policies. An additional parameter sensitivity analysis manifests that SIC introduces stable improvement.

2 Methods

2.1 Preliminaries

We consider a fully cooperative multi-agent game with N agents. The game can be described as a tuple as $\langle \mathcal{I}, \mathcal{S}, \mathcal{A}, T, r, \gamma, \rho_0 \rangle$. Let $\mathcal{I} = \{1, 2, \cdots, n\}$ denote the set of n agents. $\mathcal{A} = \langle \mathcal{A}_1, \cdots, \mathcal{A}_n \rangle$ is the joint action space of agents, and $\mathcal{S}$ is the global state space. At time step t, the group of agents takes the joint action $\mathbf{a}_t = \langle a_{1t}, a_{2t}, \cdots, a_{nt} \rangle$ with each $a_{it} \in \mathcal{A}_i$ indicating the action taken by the agent i. $T(s_{t+1}|s_t, \mathbf{a}_t) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition function. $r(s_t, \mathbf{a}_t) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ indicates the reward function from the environment. $\gamma \in [0, 1)$ is a discount factor and $\rho_0 : \mathcal{S} \to [0, 1]$ is the distribution of the initial state s_0.

Let $\pi_i(a_{it}|s_t) : \mathcal{S} \times \mathcal{A}_i \to [0, 1]$ be a stochastic policy for agent i, and denote the joint policy of agents as $\pi = \langle \pi_1, \cdots, \pi_n \rangle \in \Pi$ where Π is the joint policy space. Let $J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$ denots the expected discounted cumulative reward, where r_t is the reward received at time-step t following policy π. We aim to optimize the joint policy π to maximize $J(\pi)$.

2.2 Joint Policy Space with Coordination Signal

In the fully centralized paradigm, a centralized controller is used to manipulate a group of agents. We denote Π^C as the policy space of the centralized controller and $\pi^C : \mathcal{S} \times \mathcal{A} \to [0, 1]$ as a joint policy in Π^C. In the decentralized execution paradigm, the agents make decisions independently according to their individual policies $\pi_i^D : \mathcal{S} \times \mathcal{A}_i \to [0, 1]$. We define the policy space of agent i as Π_i^D and the joint policy space as $\Pi^D = \Pi_1^D \times \cdots \times \Pi_n^D$, i.e. the Cartesian product of the policy spaces of each agent. For a joint policy $\pi^D \in \Pi^D$, we have $\pi^D(\boldsymbol{a}|s) = \pi_1^D(a_1|s) \cdots \pi_n^D(a_n|s)$, $\forall s \in \mathcal{S}$ and $\forall \boldsymbol{a} = \langle a_1, \cdots, a_n \rangle \in \mathcal{A}$. We conclude the relation between Π^C and Π^D as the following proposition:

Proposition 1. *Π^D is a subset of Π^C.*

This proposition is obvious and we provide a proof in the appendix. This proposition reveals one critical issue: as the objective of optimization is $J(\pi)$ instead of $J(\pi_1 \times \cdots \times \pi_N)$, it is possible that current CTDE methods can never perform as good as centralized methods when the optimal policy is in the complement space, i.e., $\Pi^C \backslash \Pi^D$. An intuitive method to solve this problem is to assume agents act sequentially and those who act later condition their policies on previous ones, e.g., $\pi^C(\boldsymbol{a}|s) = \pi_1(a_1|s)\pi_2(a_2|s, a_1) \cdots \pi_N(a_N|s, a_1, \cdots, a_{N-1})$. However, this method is not practical as it requires stable communication channel and large bandwidth to implement.

From a game-theoretic perspective, the decentralized agents try to reach a Nash equilibrium, with each individual policy as a best response to others' policies. Previous studies on computational game theory show that by following the signal provided by a *correlation device*, agents may reach a more general type of equilibrium, *correlated equilibrium* (CE) [17], which can potentially lead to better outcomes for all agents [1,7,13]. Inspired by CE, we propose a *signal instructed* framework. We introduce a coordination signal sent to every agent at the beginning of a game, which is conceptually close to the signal sent by the correlation device in CE. The usage of the signal changes Π^D to a different joint policy space, Π^S. Every agent observes the same signal $z \in \mathcal{Z}$ sampled from a conditional distribution P_z, where $\mathcal{Z}$ is the *signal space*, and learns an individual policy as $\pi_i^S(a_i|s, z)$. Therefore, the agents formulate a special joint policy, $\boldsymbol{\pi}^S$, which suffices that $\forall s \in \mathcal{S}$, $\forall \boldsymbol{a} = \langle a_1, \cdots, a_n \rangle \in \mathcal{A}$, and $\forall z \in \mathcal{Z}$, $\boldsymbol{\pi}^S(\boldsymbol{a}, z|s) = P_z(z|s) \cdot \pi_1^S(a_1|s, z) \cdots \pi_n^S(a_n|s, z)$. $\boldsymbol{\pi}^S$ is a conditional joint distribution of $\boldsymbol{a}$ and z, and differs from aforementioned types of joint policies. However, by regarding z as an extension of global state, we do not change the way we model the individual policy of each agent, which is still $\pi_i^S(a_i|s')$ with $s' = (s, z)$.

In signal instructed approach, all agents observe the same z, and we assume that every agent follows the instruction of z, i.e., takes only one specific corresponding action a_i^z. We denote the corresponding joint action as $\boldsymbol{a}^z = \langle a_1^z, \cdots, a_n^z \rangle$. Intuitively, this assumption is like that agents make an "agreement" on which joint action to take in current state when observing z, which is common in real-world scenarios. For example, the cars in a traffic junction can tell whether they should accelerate or stop from the same observed pose of a policeman, and they can be regarded as reaching a CE. Following the assumption still results in a stochastic joint policy, with the stochasticity conditioned on z now. With this assumption, we derive Proposition 2:

Proposition 2. *Π^S is equal to Π^C.*

We provide a proof in the appendix. This proposition shows that the signal instructed method enlarged the joint policy space to the same size as Π^C, while still maintaining a mostly decentralized framework. Therefore, agents can still act with their decentralized individual policies, while exploiting a larger joint policy space. We illustrates the relationship among different joint policy spaces in Fig. 1.

A practical concern is that according to the assumption, agents need to assign every joint action to a specific z, which means that the size of $\mathcal{Z}$ can be very

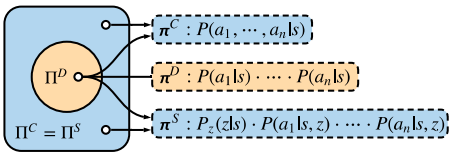


Fig. 1. The relationship among Π^C, Π^D and Π^S is $\Pi^C = \Pi^S \supseteq \Pi^D$. The white circle represents an element in the set.

large. Fortunately, the set of optimal joint actions is often small, and hence an optimal π^S needs only a small subspace of $\mathcal{Z}$ to instruct agents to take those optimal joint actions. Another concern is that agents might choose a non-optimal joint action when following the instruction of z, since the signal is drawn from a random distribution. Intuitively, a signal z serves as a "consensus", so that agents can infer others' actions and lean a good policy correspondingly. The signal itself does not explicitly tell each agent which action to take. Hence, when there is only one optimal joint action in current state, agents can learn to take the corresponding individual action whatever the signal is. In other words, the size of the signal space reduces to 1 in this case. In following sections, we show how this is achieved.

2.3 Signal Instructed Coordination

When a coordination signal is observed, how to incentivize agents to follow its instruction and coordinate is a critical issue. As a coordination signal is sampled from P_z, it is possible for agents to treat it as random noise and ignores it during the training process. Our idea is to facilitate the coordination signal to be entangled with agents' behaviors and thus encourage the coordination in execution. We name our method as *Signal Instructed Coordination (SIC)*. SIC introduces an information-theoretic regularization to ensure the signal makes an impact in agents' decision making. This regularization aims to maximize the mutual information between the signal z, and the joint policy π^S, given current state s, as

$$\begin{aligned} I(z; \pi^S) &= -H(\pi^S|z) + H(\pi^S) \\ &= -H(\pi_i^S|z) - H(\pi_{-i}^S|\pi_i^S, z) + H(\pi^S), \end{aligned} \tag{1}$$

where π^S, π_i^S and π_{-i}^S are abbreviations for $\pi^S(\boldsymbol{a}, z|s)$, $\pi_i^S(a_i, z|s)$ and $\pi_{-i}^S(\boldsymbol{a}_{-i}, z|s)$, and π_{-i}^S and $\boldsymbol{a}_{-i}$ are the joint policy and the joint action of all agents except agent i respectively. The decomposition of π^S into π_i^S and π_{-i}^S in the second line holds in our decentralized approach.

Through decomposing the regularization term in Eq. (1), one can find that the effects for maximizing the mutual information between signal and policy are threefold. Minimizing the first term increases consistency between the coordination signal and the individual policy to suffice the assumption of Proposition 2. Minimizing the second term ensures low uncertainty of other agents' policies, which is beneficial to establish coordination among agents. Maximizing the third

term encourages the joint policy to be diverse, which prohibits the opponents from inferring our policy in competition. These effects in combination improves the performance of the joint policy. However, directly optimizing Eq. (1) is intractable. Considering the symmetry property of mutual information, we aim to maximize

$$\begin{aligned} I(z;\pi^S(\boldsymbol{a},z|s)) &= -H(z|\pi^S(\boldsymbol{a},z|s)) + H(z) \\ &= \mathbb{E}_{z\sim P_z(\cdot|s),\boldsymbol{a}\sim\pi(\cdot|s,z)}[\mathbb{E}_{z'\sim P(\cdot|s,\boldsymbol{a})}\log P(z'|s,\boldsymbol{a})] + H(z) \quad (2) \\ &\geq \mathbb{E}_{z,\boldsymbol{a}}[\log U(z|s,\boldsymbol{a})] \quad (3) \end{aligned}$$

where $U(z|s,\boldsymbol{a})$ is an approximation of $P(z|s,\boldsymbol{a})$. A detailed derivation from Eq. (2) to Eq. (3) can be found in the appendix. The inequality sign holds due to $D_{KL}(\cdot) \geq 0$ and $H(\cdot) \geq 0$, and the equality sign in the last line holds as proved in Lemma 5.1 of [5]. Considering the visiting probability of s in sampled trajectories τ following π, we derive a mutual information loss (MI loss) from Eq. (3) as

$$L_I(\pi,U) = -\mathbb{E}_{s\sim\tau,z\sim P_z(\cdot|s),\boldsymbol{a}\sim\pi(\cdot|s,z)}[\log U(z|s,\boldsymbol{a})]. \quad (4)$$

Minimizing Eq. (4) facilitates agents to follow the instruction of the coordination signal.

2.4 Implementation Details

In the proposed signal instructed coordination method, agents optimize their joint policy to maximize expected returns and minimize the mutual information loss. To integrate SIC with existing models, the first problem is how to model P_z. Instead of using discrete signal space, we propose to adopt a continuous signal space, and approximate P_z in a Monte Carlo way. In detail, we sample a D_z-dimension continuous vector v from a normal distribution $\mathcal{N}(\mathbf{0},I)$, and distribute it to all agents. v can be viewed as a proxy of the variable z. Agents learn to divide the $\mathbb{R}^{D_z}$ space into several subspaces, with each corresponding to one signal and hence one optimal joint action. The probability of sampling a specific z, $P_z(z|s)$, is approximated by the probability of sampling a vector v that belongs to the corresponding subspace. Therefore, we can replace z in Eq. (4) with v and change to compute $U(v|s,\boldsymbol{a})$.

To compute $U(v|s,\boldsymbol{a})$, we use a centralized multi-layer feed-forward network, named as U-Net, as a parameterized function, f_U. U-Net inputs s and $\boldsymbol{a}$, and outputs a continuous vector with the same dimension as v as the reconstructed value of the signal, $v' = f_U(s,\boldsymbol{a})$. Intuitively, we hope $v' = v$, which means that agents follow the instruction so well that we can infer what they see from their behaviors. Therefore, $U(v|s,\boldsymbol{a})$ is measured by the mean squared error between v' and v and minimized during training. One obvious advantage of SIC is that it can be easily integrated with most existing models with policy networks, as shown in Fig. 2. To pass gradients even when stochastic policy is used, we use a simplified approximation $U(z|s,\boldsymbol{h})$, where $\boldsymbol{h}$ is the concatenation of last layers of

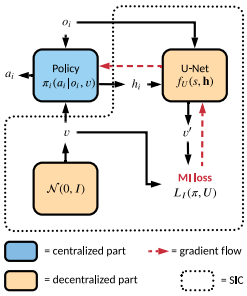


Fig. 2. Illustration of SIC. Black arrows indicate how variables are passed between components. Note that in U-Net, we use the concatenation of last hidden vectors in policy networks $\boldsymbol{h}$, instead of $\boldsymbol{a}$, to enable gradient flow when adopting stochastic policies. Besides, in practice we sample a continuous vector v from a fixed normal distribution $\mathcal{N}(0, I)$ as a proxy of z.

hidden vectors in policy networks. Parameters of the centralized U-net, ω, and parameters of decentralized policies, $\theta = \langle \theta_1, \cdots, \theta_n \rangle$, are jointly optimized as:

$$\max_{\omega,\theta} \mathbb{E}_{\boldsymbol{o}\sim\tau,\boldsymbol{a}\sim\pi_\theta,v\sim\mathcal{N}(0,I)}[Q_i(\boldsymbol{o},\boldsymbol{a}) - \alpha L_I(\pi, U_\omega)], \tag{5}$$

where $Q_i(\boldsymbol{o}, \boldsymbol{a}|z)$ is the centralized critic function, and $\alpha > 0$ is the hyperparameter for information maximization term. By optimizing this objective, we aim to find a trade-off between *maximizing long-term returns* and *reaching consensus*. $Q_i(\boldsymbol{o}, \boldsymbol{a}|z)$ can also be substituted with the advantage function used by COMA. We do not share parameters among agents. Note that o_i may be a partial observation of agent i, and additional communication mechanism can be introduced to ensure theoretical correctness of Eq. (4). However, we empirically show that in some partially observable environments, e.g., particle worlds [22], where the agent can infer global state from its local observation, SIC can still work with o_i. When applied to Multi-agent Actor-Critic frameworks [22], the objective of updating critic remains unchanged.

3 Experiments

3.1 Rock-Paper-Scissors-Well (RPSW)

We use a 2 vs 2 variant of the matrix game, *Rock-Paper-Scissors-Well (RPSW)*. Each team consists of two independent agents, and the available actions of each agent are $Access(A)$ and $Yield(Y)$. The joint action space of each team consists of *Rock* ($\langle Y, Y \rangle$), *Paper* ($\langle Y, A \rangle$), *Scissors* ($\langle A, Y \rangle$), and *Well* ($\langle A, A \rangle$). The former three actions play as in the traditional *Rock-Paper-Scissors (RPS)* game, while *Well* wins only against *Paper* and is defeated by *Rock* and *Scissors*. We present the payoff matrix in Table 1. This game can reflect agents' capability to coordinate. Assume both teams are controlled by centralized controllers, the

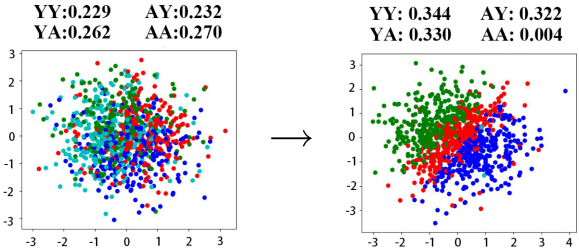
Table 1. Payoff matrix of the 2 vs 2 RPSW game (M_4). Both row and column players consist of two agents, who coordinate with individual actions as Y or A to play a joint action, e.g., *Paper* ($\langle Y, A\rangle$), and receive shared rewards from (r_{row}, r_{col}).

	Rock $\langle Y, Y\rangle$	Paper $\langle Y, A\rangle$	Scissors $\langle A, Y\rangle$	Well $\langle A, A\rangle$
Rock $\langle Y, Y\rangle$	(0, 0)	(−1, 1)	(1, −1)	(1, −1)
Paper $\langle Y, A\rangle$	(1, −1)	(0, 0)	(−1, 1)	(−1, 1)
Scissors $\langle A, Y\rangle$	(−1, 1)	(1, −1)	(0, 0)	(1, −1)
Well $\langle A, A\rangle$	(−1, 1)	(1, −1)	(−1, 1)	(0, 0)

best π^C is $P(\langle Y, Y\rangle) = P(\langle Y, A\rangle) = P(\langle A, Y\rangle) = \frac{1}{3}$ and $P(\langle A, A\rangle) = 0$, since it is better to take $\langle A, Y\rangle$ instead of $\langle A, A\rangle$. To achieve this, agents within the same team need to coordinate to avoid the disadvantaged joint action $\langle A, A\rangle$, and choose others uniformly randomly. From a probabilistic perspective, the coordination requires high correlation between teammates, otherwise either $\langle A, A\rangle$ is inevitable to appear as long as $\pi_1(A) \cdot \pi_2(A) > 0$, or the joint action space degenerates to $\{\langle Y, Y\rangle, \langle Y, A\rangle\}$ or $\{\langle Y, Y\rangle, \langle A, Y\rangle\}$, and their joint policy may be easily exploited by the opponent.

We denote the matrix game in Table 1 as M_4, since the fourth joint action is undesired. By exchanging the fourth row with the i-th row, and the fourth column with the i-th column sequentially, we turn the i-th joint action to *Well* and denote the new matrix as M_i. In this way we obtain a set of matrices $\mathcal{M} = \{M_1, M_2, M_3, M_4\}$. We design a multi-step matrix game, where two teams play according to a random payoff matrix drawn from $\mathcal{M}$ in each step. Each agent can only observe the ID $i \in \{1, 2, 3, 4\}$ of the current matrix and the coordination signal. To simulate sparse rewards, we only give agents the sum of rewards on each step after an episode of game is finished, and train them with discounted returns. We use REINFORCE algorithm with fully independent agents as the baseline model, which we denote as **IND-RE**. We apply our SIC module to REINFORCE algorithm, and denote it as **SIC-RE**. The team-shared coordination signal $\boldsymbol{z} \in \mathbb{R}^2$ is sampled from $\mathcal{N}(\mathbf{0}, I)$. Note that each agent takes a stochastic policy, and signals received by the two teams are different. We conduct experiments and observe that SIC-RE defeats IND-RE with averaged rewards close to 0.4, while both IND-RE vs IND-RE and SIC-RE vs SIC-RE settings gradually converge to a tie with averaged rewards equal to 0. We can see that although IND-RE also reaches an equilibrium with a game value as 0 in IND-RE vs IND-RE, its ability to coordinate is limited, as it can only formulate joint policy in Π^D. Therefore, in direct competition as SIC-RE vs IND-RE, IND-RE is outperformed and stuck in a disadvantaged equilibrium with a negative averaged reward.

To better study which kind of equilibrium agents have reached, we test how agents respond to 5000 randomly sampled signals before and after training on M_4 with SIC-MA vs SIC-MA, and plot results of row players in Fig. 3. Note that results of column players are similar to Fig. 3. Before training (a), the frequency



(a) Row player (0 ep). (b) Row player (100k ep).

Fig. 3. Correlations between signal distribution and joint actions of row players. Each point represents a 2-dim signal, and red, green, blue, and cyan represent the corresponding joint action as $\langle Y, Y\rangle$, $\langle Y, A\rangle$, $\langle A, Y\rangle$, and $\langle A, A\rangle$ respectively. The frequency of different actions in these 5000 points is shown above each sub-figure.

of each joint action is roughly 0.25. each agent takes a random individual policy. In addition, the distribution of signals triggering different joint actions are quite spreading. After training (b), the signal space is roughly divided into three "zones", with each zone representing one joint action. The "area" of each zone, i.e., the probability of sampling one signal belonging to the zone, is roughly 1/3, which indicates that the result is close to the best performance a centralized controller can achieve.

3.2 Particle Worlds

In this section, we evaluate SIC on two particle world game, **Cooperative Navigation** and **Predator-Prey**, following the implementation in [22]. In cooperative navigation, $N = 3$ agents and $L = 3$ landmarks are randomly placed in a two-dimensional world. In each timestep, agents are rewarded with a common reward, which is the sum of the negative Euclidean distance between a landmark and the nearest agent. In addition, two agents are penalized simultaneously if they collide with each other. In Predator-Prey, M slow *predators* and M fast *preys* are randomly placed in a two-dimensional world with $L = 2$ large landmarks impeding the way, and predators need to collaborate to collide with another team of agents, preys. Note that when collisions happen, the predators are rewarded simultaneously, while the preys are penalized independently, which is different from the original setting in [22].

We use three popular models as our baselines: **MADDPG** [22], **COMA** [9] and **MAAC** [12]. All three models follow the CTDE framework, and MADDPG and COMA especially model only individual policies. Therefore, we integrate SIC with MADDPG and COMA and denote them as **SIC-MA** and **SIC-COMA**. We use hyper-parameters of neural networks in the original paper, and inherit them in SIC variants. For comparison, we also implement fully-decentralized and fully-centralized actor-critic methods based on DDPG, and denote them as **Dec-AC**

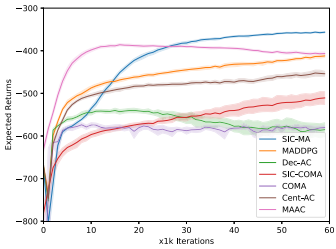


Fig. 4. Results of different models on Cooperative Navigation game.

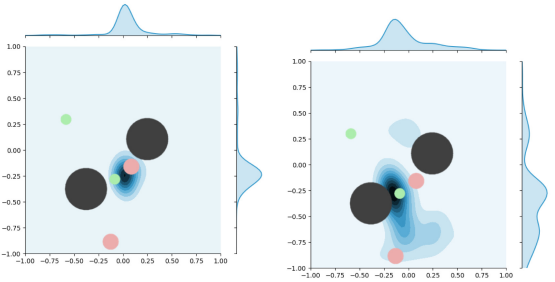
Table 2. Results of different models on Predator-Prey game when playing against MADDPG preys. Results are reported in terms of predator scores with 95% confidence intervals in 30 repeated games.

Predator model	2 vs 2	4 vs 4
Dec-AC	6.70 ± 0.78	31.97 ± 2.08
COMA	1.50 ± 0.15	20.05 ± 2.46
MADDPG	12.11 ± 1.39	46.80 ± 3.29
MAAC	11.48 ± 2.01	50.33 ± 3.65
SIC-COMA	2.05 ± 0.12	26.33 ± 2.64
SIC-MA	**16.56** ± 1.50	**59.85** ± 2.89
Cent-AC	17.31 ± 1.14	40.33 ± 1.51

and **Cent-AC**. Note that to evaluate the performance of different models, we compare them with the same opponent prey model (MADDPG) and report predator scores on predator-prey games. We report results of Cooperative Navigation in Fig. 4 and Predator-Prey (2 vs 2 and 4 vs 4) in Table 2. We can see that

1. Although SIC requires additional learning budgets and slows down the learning speed in the initial stage, it stably improves performance when combined with a baseline model. In addition, SIC-MA significantly outperforms all other decentralized execution models including the complicated SOTA baseline (MAAC).
2. On 2 vs 2 Predator-Prey game, the performance of SIC-MA is close to that of the fully-centralized method. On 4 vs 4 Predator-Prey and Cooperative Navigation games, where the training of the centralized method suffers, SIC-MA learns better joint policies through decentralized paradigm. This shows that even in high-dimensional space, coordination signal can still facilitate coordination among decentralized agents.

To better understand how SIC improves performance, we conduct case study on the mixed cooperative-competitive environment, 2 vs 2 Predator-Prey game. We visualize the distribution of collision positions in Fig. 5. We repeat MADDPG vs MADDPG, and SIC-MA vs MADDPG for 10000 games with different



(a) MADDPG vs MADDPG (1002 collisions)

(b) SIC-MA vs MADDPG (1637 collisions)

Fig. 5. The density and marginal distribution of positions of collision, (x, y), in 10000 repeated games initialized from the same environment. The orange, green and black circles representing predators, preys and landmarks at the start of each game. The result shows that compared to MADDPG, SIC-MA presents more diverse strategy and better performance.

seeds. We reset each game to the same state in each episode, and collect positions of collisions in the total 250000 steps. We visualize the positions in Fig. 5, where data points (x, y) with higher frequency are colored darker in the plane. In Fig. 5-a, most collisions happen around the lower prey, which reflects that both predators only collaborate to capture the lower prey. When SIC-MA plays predators as in Fig. 5-b, collisions appears in more diverse positions, and we observe two strategies: the two predators either chase the lower prey to the bottom part of the map, or move upward together to catch both preys. In this case, it is hard for preys to exploit the opponent strategies, but the two predators need high level of coordination to conduct such strategies, otherwise the lower prey may flee through the interval between them. Specifically, SIC-MA captures more preys (1637) than MADDPG does (1002). This evidences that SIC-MA learns better policies compared to MADDPG.

4 Related Works

Recent works [4] on MARL focus on complex scenarios with high dimensional state and action spaces like particle worlds [22] and StarCraft II [30]. Among different approaches to model the controlling of agents, *centralized training with decentralized execution* [22,25] outperforms others for circumventing the exponential growth of joint action space and the non-stationary environment problem [19]. Emergent communication [21] is proposed to enhance coordination and training stability, which allows agents to pass messages between agents and "share" their observations via communication vectors. [26,28] design special architectures to share information among all agents. The noisy channel problem arises when all other agents use the same communication channel to send information simultaneously, and the agent needs to distinguish useful information from useless or

irrelevant noise. To alleviate this problem, [6,12,14] propose to introduce the attention mechanism to control the bandwidth of different agents dynamically. However, communication requires large bandwidth to exchange information, and the effectiveness of communication is under question as discussed by [21].

The coordination problem [3], or the Pareto-Selection problem [23], has been discussed by a series of works in fully cooperative environments. The solution to the coordination problem requires strong coordination among agents, i.e., all agents act as if in a fully centralized way. In the game theory domain, it can also be viewed as pursuing *Correlated equilibrium* (CE) [1,17], where agents make decisions following instructions from a correlation device. It is desired that agents in the system can establish correlation protocols through adaptive learning method instead of constructing a correlation device manually for specific tasks [11] proposes to replace the value function in Q-learning with a new one reflecting agents' rewards according to some CE. [31] maintains coordination sets and select coordinated actions within these sets.

A similar concept to our coordination signal is *common knowledge*, which refers to common information, e.g., representations of states, among partially observable agents. Common knowledge is used to enhance coordination [10,29] and combined with communication [15]. Among them, [10] proposes MACKRL which introduces a random seed as part of common knowledge to guide a hierarchical policy tree. To avoid exponential growth of model complexity, MACKRL restricts correlation to pre-defined patterns, e.g., a pairwise one, which is too rigid for complex tasks.

5 Conclusions

We propose a signal instructed paradigm to improve the popular decentralized execution framework, which theoretically manipulates decentralized agents as a centralized controller. Accordingly, we design a novel module named Signal Instructed Coordination (SIC) to enhance coordination of agents' policies by introducing a mutual information regularization. Our experiments show the performance improvement of popular centralized-training-decentralized-execution algorithms with the help of SIC.

Acknowledgements. Haifeng thanks the support from Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDA27030401. Co-author Fang is supported in part by a research grant from Lockheed Martin and NSF grant IIS-2046640. The corresponding author Weinan Zhang is supported by "New Generation of AI 2030" Major Project (2018AAA0100900), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (62076161, 61632017).

A Algorithm

For completeness, we provide the SIC-MA algorithm as an example of application of SIC in Algorithm 1. The main body of SIC-MA is the similar to MADDPG, and the main change includes:

1. a common signal is sampled and every agent observes it before taking actions, and
2. a mutual information loss is computed to update parameters of U-Net and policy networks.

We can see that SIC is easy to implement with existing actor-critic-based algorithms.

B Proof of Proposition 1

Proof. $\forall \boldsymbol{\pi}^D \in \Pi^D$, we can construct $\boldsymbol{\pi}^C \in \Pi^C$, which suffices that for $\forall s \in \mathcal{S}$, $\forall \boldsymbol{a} = \langle a_1, \cdots, a_n \rangle \in \mathcal{A}$,

$$\boldsymbol{\pi}^C(\boldsymbol{a}|s) = \pi_1^D(a_1|s) \cdot \cdots \pi_n^D(a_n|s) = \boldsymbol{\pi}^D(\boldsymbol{a}|s).$$

Thus, we have every $\boldsymbol{\pi}^D = \boldsymbol{\pi}^C \in \Pi^C$ and $\Pi^D \subseteq \Pi^C$.

We use a counterexample to show that not every joint policy in Π^C is an element of Π^D. In a two-agent system where each agent has two actions x and y, $\forall s \in \mathcal{S}$, $\exists \boldsymbol{\pi}^C \in \Pi^C$ that suffices $\boldsymbol{\pi}^C(\langle a_1 = x, a_2 = x \rangle|s) = \boldsymbol{\pi}^C(\langle a_1 = y, a_2 = y \rangle|s) = 0.5$, but $\boldsymbol{\pi}^C \notin \Pi^D$, because there is no valid solution for $\pi_1(a_1 = x|s) \cdot \pi_2(a_2 = x|s) = (1 - \pi_1(a_1 = x|s)) \cdot (1 - \pi_2(a_2 = x|s)) = 0.5$. Since Π^C includes Π^D, the best joint policy in Π^C is superior or equal to the best joint policy in Π^D. However, due to computational complexity concerns, the decentralized execution paradigm is more practical in large-scale environments. Thus, we have motivation to propose a new framework which has centralized policy space Π^C and is executed in a decentralized way.

C Proof of Proposition 2

Proof. We prove this proposition in two steps:

1. $\Pi^C \subseteq \Pi^S$: $\forall \boldsymbol{\pi}^C \in \Pi^C$, we can construct $\boldsymbol{\pi}^S \in \Pi^S$, which suffices that $\forall s \in \mathcal{S}$, $\forall \boldsymbol{a} \in \mathcal{A}$, we assign a signal $z \in \mathcal{Z}$ to $\boldsymbol{a}$ with $P_z(z|s) = \boldsymbol{\pi}^C(\boldsymbol{a}|s)$, s.t.

$$\begin{aligned} \boldsymbol{\pi}^S(z, \boldsymbol{a}|s) &= P_z(z|s)[\pi_1^S(a_1|s, z) \cdot \cdots \pi_n^S(a_n|s, z)] \\ &= P_z(z|s)[1 \cdot \cdots 1] = \boldsymbol{\pi}^C(\boldsymbol{a}|s) \end{aligned}$$

   Thus, we have every $\boldsymbol{\pi}^C = \boldsymbol{\pi}^S \in \Pi^S$ and $\Pi^C \subseteq \Pi^S$.
2. $\Pi^S \subseteq \Pi^C$: $\forall \boldsymbol{\pi}^S \in \Pi^S$, we can construct $\boldsymbol{\pi}^C \in \Pi^C$, which suffices that $\forall s \in \mathcal{S}$, $\forall \boldsymbol{a} \in \mathcal{A}$, $\forall z \in \mathcal{Z}$,

$$\boldsymbol{\pi}(\boldsymbol{a}|s) = \begin{cases} P_z(z|s) & \boldsymbol{a} = \boldsymbol{a}^z \\ 0 & \text{otherwise.} \end{cases}$$

   Thus, we have every $\boldsymbol{\pi}^S = \boldsymbol{\pi}^C \in \Pi^C$ and $\Pi^S \subseteq \Pi^C$.

Given $\Pi^S \subseteq \Pi^C$ and $\Pi^S \supseteq \Pi^C$, we have $\Pi^S = \Pi^C$.

Algorithm 1. SIC-MADDPG Algorithm

for episode = 1 to M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial state $\mathbf{x}$
  Generate random signal z according to pre-defined distribution.
  for $t = 1$ to max-episode-length **do**
    for each agent i, sample action a_i according to $\pi_i(a_i|o_i, z)$
    Execute actions $a = (a_1, \ldots, a_N)$ and observe reward r and new state $\mathbf{x}'$
    Store $(\mathbf{x}, a, r, \mathbf{x}', z^j)$ in replay buffer $\mathcal{D}$
    $\mathbf{x} \leftarrow \mathbf{x}'$
    for $k = 1$ to 2 **do**
      Sample a random mini-batch of S samples $(\mathbf{x}^j, a^j, r^j, \mathbf{x}'^j, z^j)$ from $\mathcal{D}$
      for agent i in team k **do**
        Calculate h_k^j by concatenating inputs to the last layer of policy networks of all cooperative agents in team k
        Calculate $U_k^j = U(z_k^j|\mathbf{x}_k^j, h_k^j)$
        Set $y^j = r_i^j + \gamma Q_i^{\pi'}(\mathbf{x}'^j, \pi_k)|_{a'_k = \pi'_k(\sigma_k^j)}$
        Update critic by minimizing the loss

$$\mathcal{L}(\theta_i) = \frac{1}{S}\sum_j (y^j - Q_i^{\pi}(\mathbf{x}^j, \pi_k^j))^2$$

        Update actor using the sampled policy gradient

$$\nabla_{\theta_i} J \approx \frac{1}{S}\sum_j \nabla_{\theta_i}\pi_i(o_i^j)\nabla_{a_i}[Q_i^{\pi}(\mathbf{x}^j, \pi^j) + \beta L_I(\pi, U)]|_{a_i = \pi_i(\sigma_i^j)}$$

        Update U-Net by minimizing

$$\mathcal{L}(w) = \frac{1}{S}\sum_j L_I(\pi, U)]|_{a_i = \pi_i(\sigma_i^j)}$$

      end for
    end for
    Update target network parameters for each agent i:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$$

  end for
end for

D Derivation of the Lower Bound

The derivation of the lower bound mostly follows similar techniques used in variational inference domain [2,5,18]. Firstly, we have

$$\begin{aligned} I(z; \boldsymbol{\pi}^S(\boldsymbol{a}, z|s)) &= -H(z|\boldsymbol{\pi}^S(\boldsymbol{a}, z|s)) + H(z) \\ &= \mathbb{E}_{z\sim P_z(\cdot|s), \boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|s,z)}[\mathbb{E}_{z'\sim P(\cdot|s,\boldsymbol{a})} \\ &\quad \log P(z'|s, \boldsymbol{a})] + H(z) \end{aligned} \tag{6}$$

where $\boldsymbol{\pi}(\cdot|s, z)$ is the Cartesian product of individual policies after observing a specific z, and $P(\cdot|s, \boldsymbol{a})$ is the posterior distribution estimating the probability of a specific signal z' after seeing the state s and the joint action $\boldsymbol{a}$. Note that $P(\cdot|s, \boldsymbol{a})$ is not the same as $P_z(\cdot|s)$. Since we have no knowledge of the posterior distribution, we circumvent it by introducing a variational lower bound [2,5,18] which defines an auxiliary distribution $U(\cdot|s, \boldsymbol{a})$ as:

$$\begin{aligned} \text{Eq. (6)} &= \mathbb{E}_{z\sim P_z(\cdot|s), \boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|s,z)}[D_{KL}(P(\cdot|s, \boldsymbol{a})||U(\cdot|s, \boldsymbol{a})) \\ &\quad + \mathbb{E}_{z'\sim P(\cdot|s,\boldsymbol{a})} \log U(z'|s, \boldsymbol{a})] + H(z) \\ &\geq \mathbb{E}_{z\sim P_z(\cdot|s), \boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|s,Tz)}[\mathbb{E}_{z'\sim P(\cdot|s,\boldsymbol{a})} \log U(z'|s, \boldsymbol{a})], \\ &= \mathbb{E}_{z\sim P_z(\cdot|s), \boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|s,z)}[\log U(z|s, \boldsymbol{a})] \end{aligned} \tag{7}$$

E Experiment Details

E.1 Matrix Game Experiment

We conduct three multi-step matrix game experiments: SIC-RE vs SIC-RE, SIC-RE vs IND-RE, IND-RE vs IND-RE. For both SIC-RE and IND-RE models, we use the Adam optimizer with a learning rate of 0.0001. The policy network is parameterized by a one-layer ReLU MLP with 8 hidden units. We use a batch size of 100000. For SIC-RE models, we use, a two-layer ReLU MLP with 8 hidden units as U-Net, and set the coefficient of MI loss to be 0.01.

E.2 Particle World Experiment

We adopt 5 different models: MADDPG, SIC-MADDPG (SIC-MA), COMA, SIC-COMA, and MAAC. We provide details of our setups here:

1. **MADDPG** We use the Adam optimizer with a learning rate of 0.001 and $\delta = 0.01$ (has the same meaning as in original MADDPG) for updating the target network. Both Actor and Critic are parameterized by a two-layer ReLU MLP with 64 units per layer. γ is set to be 0.95. We use a batch size of 1024 before making an update.
2. **SIC-MADDPG** We use the Adam optimizer with a learning rate of 0.0005 and $\delta = 0.01$ for updating the target network. Gradient clipping is set to be 0.5. Both Actor and Critic are parameterized by a two-layer ReLU MLP with 64 units per layer. γ is set to be 0.95. We use a batch size of 1024 before making an update. The dimension of signals is 20 and the coefficient of information-theory regularization is 0.0001.

3. **COMA** We use the Adam optimizer with a learning rate of 0.00005 and $\delta = 0.01$ for updating the target network. Gradient clipping is set to be 0.1. Both Actor and Critic are parameterized by a two-layer ReLU MLP with 64 units per layer. γ is set to be 0.99 and λ is set to 0.8. We use a batch size of 1000 before making an update.
4. **SIC-COMA** We use the Adam optimizer with a learning rate of 0.00005 and $\delta = 0.01$ for updating the target network. Gradient clipping is set to be 0.1. Both Actor and Critic are parameterized by a two-layer ReLU MLP with 64 units per layer. γ is set to be 0.99 and λ is set to 0.8. We use a batch size of 1000 before making an update. The dimension of signals is 20.
5. **MAAC** We use the Adam optimizer and set the learning rate for policy and critic networks as 0.001 and 0.01 respectively. Both policy and critic networks adopt two-layer Leaky ReLU MLP with 128 units per layer. The number of attention head is set to 4. We use a batch size of 100, and set the reward rescaling factor to be 100 as in the original paper.

Except aforementioned models, we also implement two variants of MADDPG: a fully-decentralized actor-critic and a fully-centralized actor-critic. The former one can be viewed as MADDPG with a decentralized critic $Q_i(o_i, a_i)$, and the latter one as MADDPG with a centralized agent that takes joint actions directly $\boldsymbol{a} = \pi(\boldsymbol{o})$.

We report results of games with 95% confidence intervals in 30 repeated games.

F Visualization for Joint Policy of Multi-step Matrix Game

We plot the curves of joint policies of both row players and column players in multi-step matrix games in Fig. 6, 7, and 8.

G Parameter Sensitivity

We conduct a parameter sensitivity analysis in 2 vs 2 Predator-Prey game on two crucial parameters of SIC, the coefficient of mutual information loss α and the dimension of signal D_z. We test SIC-MA vs MADDPG with different values of parameters in the 2 vs 2 Predator-Prey game, We find that when adopting signal without training U-Net ($\alpha = 0$), the performance of SIC-MA is close to MADDPG. Therefore, enforcing the mutual information constraint properly ($\alpha = 1e-4$) is important in achieving better results. Besides, SIC-MA presents a stable improvement over MADDPG ($D_z = 0$) and, most importantly, approximation through neural networks can compress $\mathcal{Z}$ and ensure good performance.

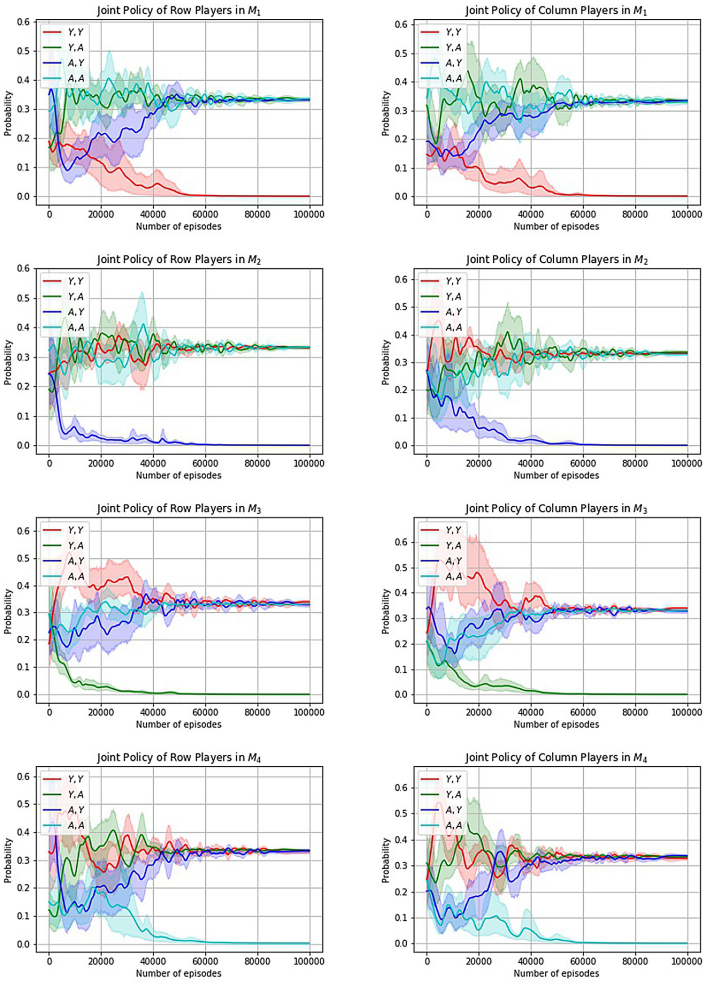


Fig. 6. Joint Policy of SIC-RE vs SIC-RE. During training, the i-th joint action in M_i is deprecated gradually, and all other joint actions are sampled uniformly randomly.

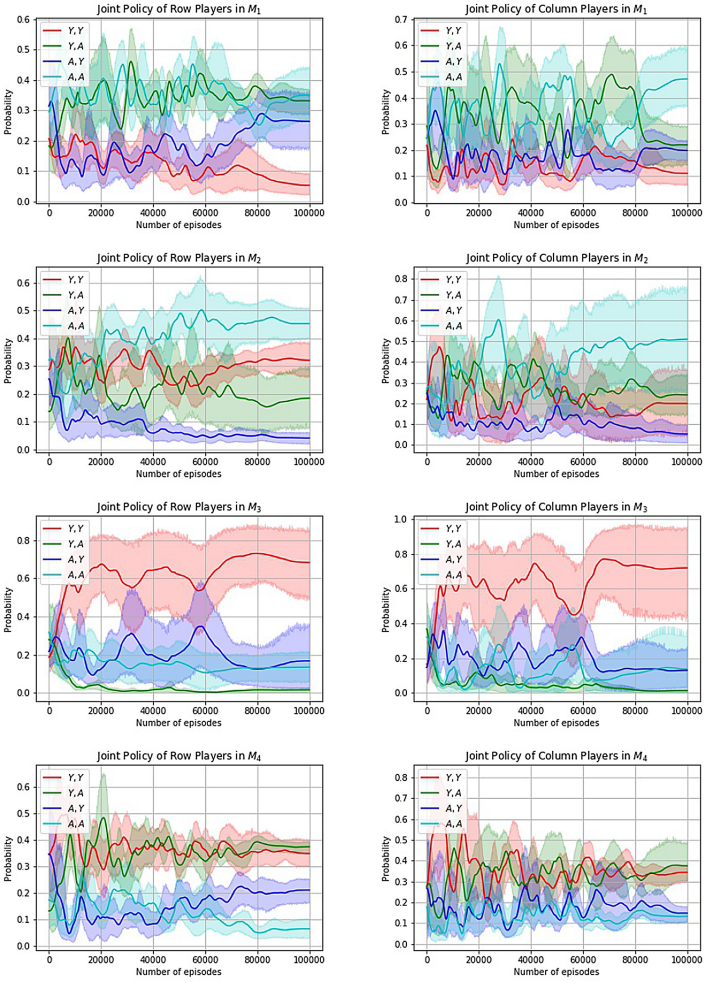


Fig. 7. Joint Policy of SIC-RE vs IND-RE. SIC-RE adjusts its joint policy to counter that of IND-RE, and achieves a positive game value.

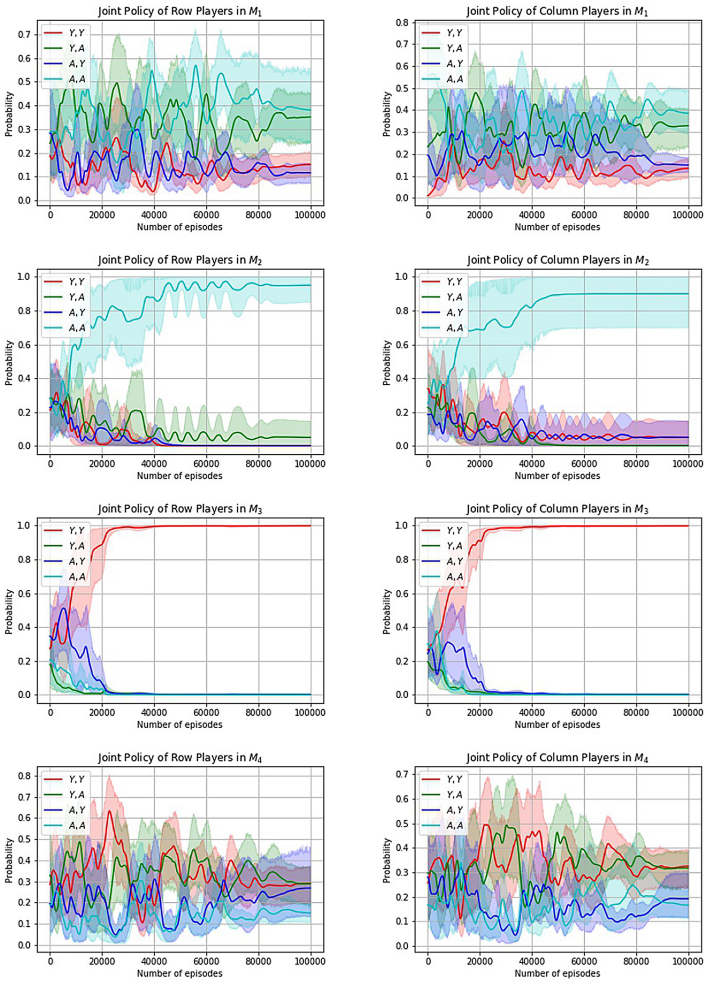


Fig. 8. Joint Policy of IND-RE vs IND-RE. IND-RE only finds worse joint policy in the team-policy space, and in some cases (M_2 and M_3), players play only one kind of joint action.

References

1. Aumann, R.J.: Subjectivity and correlation in randomized strategies. J. Math. Econ. **1**(1), 67–96 (1974)
2. Barber, D., Agakov, F.V.: The IM algorithm: a variational approach to information maximization. In: NIPS. p. None (2003)
3. Boutilier, C.: Sequential optimality and coordination in multiagent systems. In: IJCAI, vol. 99, pp. 478–485 (1999)
4. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. IEEE SMC-Part C Appl. Rev. **38**(2), 2008 (2008)

5. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS, pp. 2172–2180 (2016)
6. Das, A., et al.: Tarmac: Targeted multi-agent communication. arXiv preprint arXiv:1810.11187 (2018)
7. Farina, G., Ling, C.K., Fang, F., Sandholm, T.: Correlation in extensive-form games: saddle-point formulation and benchmarks. arXiv preprint arXiv:1905.12564 (2019)
8. Foerster, J.N., Assael, Y.M., de Freitas, N., Whiteson, S.: Learning to communicate to solve riddles with deep distributed recurrent Q-networks. arXiv preprint arXiv:1602.02672 (2016)
9. Foerster, J.N., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
10. Foerster, J.N., de Witt, C.A.S., Farquhar, G., Torr, P.H., Boehmer, W., Whiteson, S.: Multi-agent common knowledge reinforcement learning. arXiv preprint arXiv:1810.11702 (2018)
11. Greenwald, A., Hall, K., Serrano, R.: Correlated Q-learning. In: ICML, vol. 3, pp. 242–249 (2003)
12. Iqbal, S., Sha, F.: Actor-attention-critic for multi-agent reinforcement learning. arXiv preprint arXiv:1810.02912 (2018)
13. Jiang, A.X., Leyton-Brown, K.: Polynomial-time computation of exact correlated equilibrium in compact games. In: Proceedings of the 12th ACM Conference on Electronic Commerce, pp. 119–126. ACM (2011)
14. Jiang, J., Lu, Z.: Learning attentional communication for multi-agent cooperation. In: NIPS, pp. 7254–7264 (2018)
15. Korkmaz, G., Kuhlman, C.J., Marathe, A., Marathe, M.V., Vega-Redondo, F.: Collective action through common knowledge using a Facebook model. In: Proceedings of the 2014 AAMAS, pp. 253–260. IFAAMAS (2014)
16. Lanctot, M., et al.: A unified game-theoretic approach to multiagent reinforcement learning. In: NIPS, pp. 4190–4203 (2017)
17. Leyton-Brown, K., Shoham, Y.: Essentials of game theory: a concise multidisciplinary introduction. Synth. Lect. Artif. Intell. Mach. Learn. **2**(1), 1–88 (2008)
18. Li, Y., Song, J., Ermon, S.: Infogail: interpretable imitation learning from visual demonstrations. In: NIPS, pp. 3812–3822 (2017)
19. Li, Y.: Deep reinforcement learning: an overview. arXiv preprint arXiv:1701.07274 (2017)
20. Li, Y.: Deep reinforcement learning. arXiv preprint arXiv:1810.06339 (2018)
21. Lowe, R., Foerster, J., Boureau, Y.L., Pineau, J., Dauphin, Y.: On the pitfalls of measuring emergent communication. In: Proceedings of the 18th AAMAS, pp. 693–701. IFAAMAS (2019)
22. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O.P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: NIPS, pp. 6379–6390 (2017)
23. Matignon, L., Laurent, G.J., Le Fort-Piat, N.: Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems. Knowl. Eng. Rev. **27**(1), 1–31 (2012)
24. Nunes, L., Oliveira, E.: Learning from multiple sources. In: Proceedings of the 3rd AAMAS. AAMAS 2004, pp. 1106–1113. IEEE Computer Society, Washington, DC, USA (2004). http://dl.acm.org/citation.cfm?id=1018411.1018879

25. Oliehoek, F.A., Spaan, M.T., Vlassis, N.: Optimal and approximate Q-value functions for decentralized POMDPs. J. Artif. Intell. Res. **32**, 289–353 (2008)
26. Peng, P., et al.: Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games. arXiv preprint arXiv:1703.10069 (2017)
27. Schneider, J.G., Wong, W.K., Moore, A.W., Riedmiller, M.A.: Distributed value functions. In: Proceedings of the 16th ICML. ICML 1999, pp. 371–378. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999). http://dl.acm.org/citation.cfm?id=645528.657645
28. Sukhbaatar, S., Fergus, R., et al.: Learning multiagent communication with backpropagation. In: NIPS, pp. 2244–2252 (2016)
29. Thomas, K.A., DeScioli, P., Haque, O.S., Pinker, S.: The psychology of coordination and common knowledge. J. Pers. Soc. Psychol. **107**(4), 657 (2014)
30. Vinyals, O., et al.: StarCraft II: a new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782 (2017)
31. Zhang, C., Lesser, V.: Coordinating multi-agent reinforcement learning with limited communication. In: Proceedings of the 2013 AAMAS, pp. 1101–1108. IFAAMAS (2013)