QFuture: Learning Future Expectation Cognition in Multiagent Reinforcement Learning

Boyin Liu, Zhiqiang Pu, *Member, IEEE*, Yi Pan, Jianqiang Yi, *Senior Member, IEEE*, Min Chen, and Shijie Wang

***Abstract*—In multiagent reinforcement learning (MARL), agents must learn to cooperate by observing the environment and selecting actions that maximize their rewards. However, this learning process can be hampered by myopia, wherein agents' strategies fail to consider the long-term consequences of their actions. A primary reason for this problem is the inaccurate estimation of the long-term value of each action. Socially, humans derive future expectation cognition from available information to anticipate potential future outcomes and adjust their actions accordingly to avoid myopia. Motivated by these insights, this article proposes a novel framework called QFuture to address the myopia problem. Specifically, we first design a future expectation cognition module (FECM) in this framework to build future expectation cognition in the calculation of individual action-value (IAV) and joint action-value (JAV). We model future expectation cognition as random variables in FECM, which learn representation by maximizing mutual information with the future trajectory based on current information. Furthermore, a return-based regularizer is designed to reflect "expectation" and ensure informativeness in the future expectation representation module (FERM) which encodes the future trajectory. Experiments on StarCraft II micromanagement tasks and Google Research Football show that QFuture achieves significant state-of-the-art performance. Demonstrative videos are available at https://sites.google.com/view/qfuture.**

***Index Terms*—Cooperative multiagent games, Google Research Football, multiagent reinforcement learning, mutual information, StarCraft II.**

I. Introduction

REINFORCEMENT learning (RL) has become a popular research area in the field of machine learning due to its success in solving complex decision-making problems [1], [2], [3]. With the advancement of deep learning, RL models can now handle high-dimensional and continuous state-action spaces. This development led to the emergence of deep reinforcement learning (DRL), which utilizes deep neural networks to approximate value or policy functions. DRL's success in the game [4], optimization [5], and other significant domains have highlighted its potential for solving complex and challenging problems. Inspired by DRL's powerful perception and learning ability, researchers continuously attempt to apply DRL to multiagent reinforcement learning (MARL) to facilitate multiagent cooperative behaviors. Current research has made great strides in developing cooperative MARL algorithms and frameworks for multiagent systems (MASs), which have demonstrated promising results in several domains such as autonomous vehicle teams [6], swarm systems [7], and traffic management [8].

Moving from single-agent to MARL presents significant new challenges. The nonstationarity of the environment and partial observability are two crucial challenges [9]. This nonstationarity poses substantial challenges in learning effective policies for coordinating multiple agents. The agents must learn to adapt to constantly changing environments and generate robust policies to changes in the state-action space. The nonstationarity of the environment can lead to an increased focus on immediate rewards during the learning process, resulting in shortsightedness and difficulty in learning long-term strategies for agents. Furthermore, when the agents in MASs only have partial information about the current state of the environment, they may make myopic decisions that optimize their immediate individual rewards. These myopic decisions may need to pay more attention to the impact on future outcomes and disregard the global state of the environment, resulting in suboptimal joint policies and inefficient coordination among agents.

A pivotal insight to solving myopia is to make agents learn to think about the future. Since the birth of reinforcement learning, researchers have been devoted to leading the agent to learn a long-term strategy. Many practical single-agent RL solutions (Q-learning [10], state–action–reward–state–action (SARSA) [11], and actor-critic [12] methods) adopt temporal difference (TD) learning [13], where an n-step return, a combination of current reward and future reward, is used as an estimate of the value function by averaging bootstrapping from the nth state's value function estimate [14]. In addition, dynamics prediction models are also studied in model-based RL to learn the dynamics model of the environment for the synthetic experience generation or planning [15]. Dreamer [16] is proposed to learn one-step predictive environment models through training policies in the simulated environment.

Manuscript received 28 July 2023; revised 24 October 2023; accepted 18 December 2023. Date of publication 22 December 2023; date of current version 13 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62073323, in part by the Beijing Nova Program under Grants 20220484077 and 20230484435, and in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030204. *(Corresponding author: Zhiqiang Pu.)*

The authors are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: liuboyin2019@ia.ac.cn; zhiqiang.pu@ia.ac.cn; yi.pan@ia.ac.cn; jianqiang.yi@ia.ac.cn; chenmin2020@ia.ac.cn; wangshijie2020@ia.ac.cn).

Digital Object Identifier 10.1109/TCDS.2023.3345735

Multisteps and long-term futures are also modeled in [17] with recurrent variational dynamics models, after which actions are chosen through online planning. In recent studies, incorporating the future is one of the leading solutions currently used to address the myopia issue in RL. Incorporating the future by dynamically computing rollouts across many rollout lengths and using this to improve the policy [18] has been considered. In addition, building a latent-variable autoregressive model which is forced to carry future information through an auxiliary task has also shown meaningful improvements for long-term planning [19].

Due to the nonstationarity and partial observability of the environment, the issue of shortsightedness in MARL becomes more severe, and the solutions utilized in single-agent RL scenarios need to be revised to tackle this problem. However, existing works have leveraged the construction of future expectation representations to enable agents to learn longer-term strategies. In multiagent tasks, constructing a representation of the future is essential because it allows agents to anticipate the actions and behaviors of other agents and make more informed decisions about their actions [20]. A future expectation representation can also help agents to coordinate their actions and achieve a common goal more efficiently [21]. Additionally, building a representation of the future can enable agents to learn more quickly and adapt to changing environments [22]. Existing works typically construct future expectation representations for only one or a few steps ahead, which can only slightly alleviate the myopia issue for agents and needs to be revised for agents to learn efficient long-term strategies.

One of the key concepts in cognitive science is the idea of future expectation cognition—the ability to predict the future based on past experiences and current observations [23]. Specifically, when humans are obliged to make a decision, they will derive future expectation cognition based on the current observation by asking themselves: if we do that, what return would we obtain in the future? This procedure assists them in evaluating and improving decisions [24], [25]. When the future arrives, that future information will be used to enhance their future expectation cognition. Owing to future expectation cognition, humans can make more reasonable assessments of their behaviors and then emerge with elaborate cooperation skills, such as planning and tacit understanding [26]. Analogically, the emergence of future expectation cognition should also be essential for MARL which tends to focus on short-term optimization, often at the expense of long-term goals.

Inspired by cognitive science, we investigate MARL with future expectation cognition to address the myopia problem and learn long-term strategy. In our opinion, myopia is induced by the unreasonable evaluation of actions in multiagent Q-learning. For example, the action (or joint action) with a higher immediate reward is assigned with higher value, even if it leads to worse outcomes in the long-term. To address the myopia problem, it is necessary to introduce future expectation cognition into calculating individual action-value (IAV) and joint action-value (JAV) functions. In this article, we propose a novel MARL approach, called *future expectation cognition multiagent Q-learning* (QFuture), to learn future expectation cognition in MARL. Specifically, we design a future expectation cognition module (FECM), where the future expectation cognition is represented by stochastic latent variables conditioned on current observation (in IAV) or state (in JAV). We propose a novel information-theoretical objective to associate future expectation cognition with actual future trajectories by maximizing a mutual information (MI) objective. In addition, we design a future expectation representation module (FERM) and propose a regularizer to enable reasonable and effective representation of future trajectory. Finally, targeting the myopia issue in the existing action-value function calculation process, the output of FECM is used to generate the weight parameters to calculate the IAV and JAV.

In summary, our main contributions are fourfold.

1) We propose a future expectation cognition MARL framework for addressing the myopia problem and learning strategies with a long-term vision in MARL.
2) We design FECM to construct future expectation cognition, represented by stochastic latent variables conditioned on current observation (in IAV) or state (in JAV). FECM is first used in IAV to map future expectation cognition into the parameters that directly affect decisions to help agents make more proactive decisions. Then, FECM is combined with JAV to build a better estimate of expected future returns, which can effectively accelerate the learning process.
3) FERM is proposed to map any future trajectory to its latent representation to help construct future expectation cognition in FECM. FERM is forced to capture useful information by predicting future expected returns through supervised learning.
4) We conduct comprehensive experiments on StarCraft II micromanagement environments (SMAC) [27] and Google Research Football (GRF) [28]. The superior performance of our approach on challenging benchmarking tasks shows that our approach provides significantly higher coordination capacity compared with other well-known MARL algorithms.

This article is organized as follows. Section II describes the backgrounds. Section III gives a specific description of our method. In Section IV, representative simulations are carried out with two environments. Finally, conclusions are summarized in Section V.

II. Background

This section provides the context necessary to comprehend QFuture and its relationship to prior works.

A. Decentralised Partially Observable Markov Decision Process (Dec-POMDP)

We consider a fully cooperative multiagent task as a Dec-POMDP [29], which can be defined as a tuple $M = < N, S, A, P, r, Z, O, \gamma >$, where N represents a finite set of agents and $s \in S$ the true state of the environment, $\gamma \in [0, 1)$ the discount factor. At each time step, each agent $i \in N$ receives his own observation $o_i \in O$ and then chooses an action $a_i \in A$ on a global state s, forming a joint action vector $\vec{a}$. It results in a joint reward $r(s, \vec{a})$ and causes a transition in the environment based on the transition function $P(s'|, s, \vec{a})$. Each agent has its own

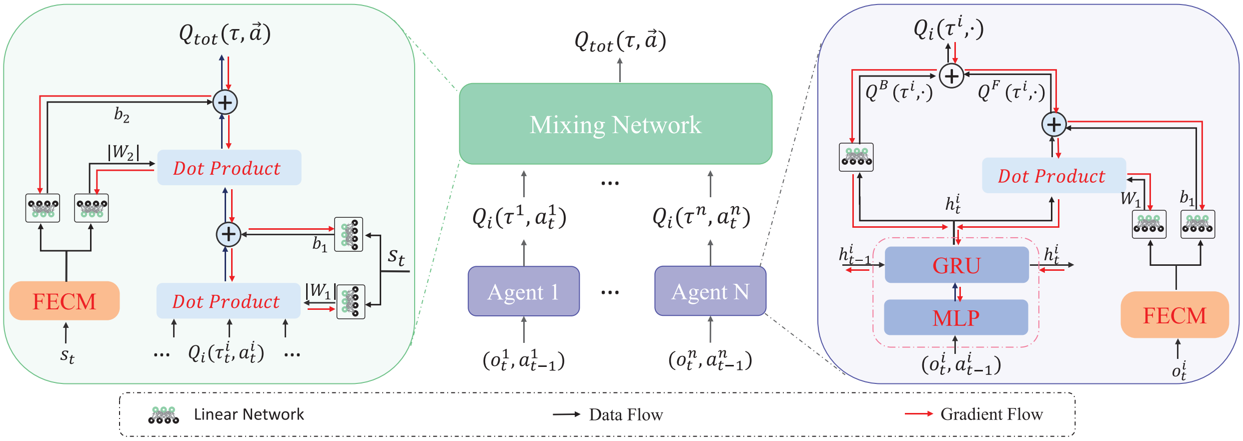


Fig. 1. Schematics of QFuture. FECM is plugged into both the utility network and the mixing network to calculate IAV and JAV, respectively. The left is the mixing network, and the right is the utility network of each agent.

action-observation history $\tau_i \in T_i \equiv (Z_i \times A)^*$, conditioned by a stochastic policy $\pi_i(a_i|\tau_i)$. The joint policy π then induces a JAV function: $Q^{\pi}_{\text{tot}}(s,\vec{a}) = \mathbb{E}_{s_{0:\infty},a_{0:\infty}}\left[G_t|\, s_0 = s, \boldsymbol{a_0} = \vec{a}, \pi\right]$, where $G_t = \sum_{t=0}^{\infty}\gamma^t r_{t+1}$ is the expected discounted return.

B. Centralized Training With Decentralized Execution (CTDE)

CTDE has been a major paradigm of cooperative multiagent deep reinforcement learning [30], [31], [32], [33] and can effectively deal with nonstationarity while learning decentralized policies for agents [34]. Agents are trained in a centralized way and have access to other agents' information or the global states during the centralized training process. Value function decomposition is of the central way to exploit the CTDE paradigm [30], [31], [32]. It learns a decentralized utility function for each agent and then adopts a mixing network to combine local utilities into a global action-value. *Individual-global-MAX* (IGM; [35]) is an essential principle to realize effective value-based CTDE which asserts that $\exists Q_i$, such that the following holds:

$$\begin{aligned}&\arg\max_{\vec{a}} Q^{\pi}_{\text{tot}}(s,\vec{a})\\&\quad = \left(\arg\max_{a_1} Q_1(\tau_1,a_1), \ldots, \arg\max_{a_N} Q_N(\tau_N,a_N)\right). \qquad (1)\end{aligned}$$

Value function factorization is the most popular method in value-based MARL under the CTDE paradigm. A value decomposition network (VDN) [36] proposes to decompose the value function of the team into agent-specific value functions by an additive factorization. QMIX [30] ameliorates the way of value function factorization by learning a mixing network, following the IGM principle [37]. Qatten [32] is a variant of VDN, which supplements global information through a multihead attention structure. DuPLEX dueling multiagent Q-learning (QPLEX) [31] employs a duplex dueling network architecture to estimate JAVs, achieving a full expressive power of IGM. However, increasing JAV's representative capability is insufficient to address the issue that future expected returns are difficult to estimate from current information. To solve this issue, we incorporate future expectation cognition into the IAV and JAV calculation procedures in this article.

C. Mutual Information

In information theory, MI is a measure of the mutual dependence between two random variables. It quantifies the amount of information that one random variable contains about another. In other words, it is a measure of how much knowing one variable reduces uncertainty about the other. Mathematically, it is defined as follows [38]:

$$\begin{aligned}\mathcal{I}(X;Y) &= \mathcal{H}(X) - \mathcal{H}(X \mid Y)\\&= \mathcal{H}(Y) - \mathcal{H}(Y \mid X) \qquad (2)\end{aligned}$$

where (X,Y) is a pair of random variables, $\mathcal{H}(X)$ and $\mathcal{H}(Y)$ are entropies, $\mathcal{H}(X \mid Y)$ and $\mathcal{H}(Y \mid X)$ represent the conditional entropies.

To emerge specific capability on agents, many MARL methods explicitly enhance the correlation of agents, where the correlations are typically quantified by the MI. For instance, to strengthen the exploration ability, a multiagent variational exploration network (MAVEN) [9] extracts the latent variables about joint policy information from the initial global state and maximizes the MI of future trajectories and the latent variables. Role-oriented multiagent (ROMA) framework [39] incorporates and cultivates role by optimizing the conditional MI between the individual trajectory and the role given the current observation. To learn diversity, CDS [40] constructs an information-theoretical regularization to maximize the MI between agents' identities and their trajectories. MI has demonstrated its advantage in guiding the agent to learn various capabilities.

III. Method

In this section, we will introduce the QFuture learning framework. The overall architectural sketch of QFuture is illustrated in Fig. 1. QFuture is a value-based MARL framework under the paradigm of CTDE. Over the course of training, neural networks are trained in a centralized manner where the agents are gathered to estimate the JAV and compute TD error for optimization. During decentralized execution, the mixing network will be removed, and each agent will use its

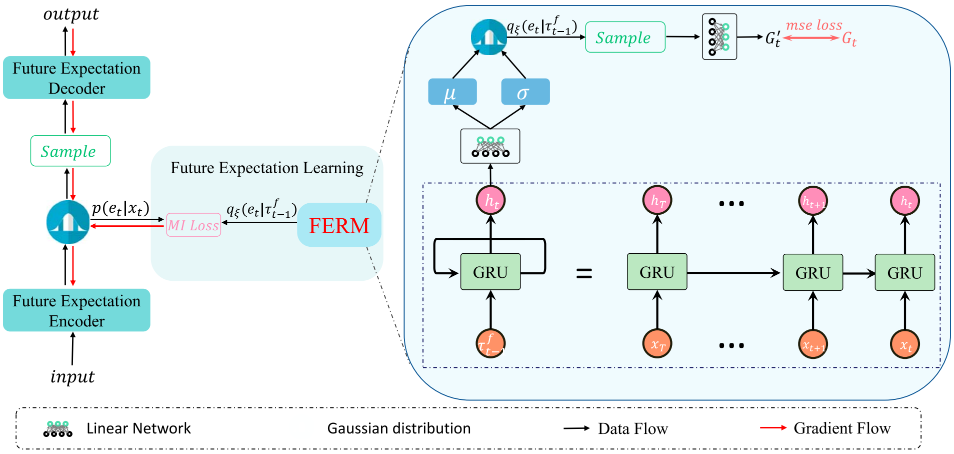


Fig. 2. Schematics of FECM. T is a final time step.

own IAV to take action with local observation. Specifically, in JAV, FECM derives the future expectation cognition from the current global state s_t and then uses them to generate the parameters to calculate JAV. In IAV, FECM uses local observation as input and generates the parameters to calculate IAV.

A. Future Expectation Cognition Module

In this article, we design a FECM, shown in Fig. 2, which generates the parameters for IAV or JAV. Since the future is uncertain given current information, future expectation cognition may not be deterministic but probabilistic. Here, we represent the future expectation cognition at time t (denoted as e_t) using a multivariate Gaussian distribution $\mathcal{N}(\mu_{e_t}, \sigma_{e_t})$, where mean μ_{e_t} and variance σ_{e_t} represents the expectation and the uncertainty of the future, respectively. Formally, at time t, its future expectation cognition is learned by

$$\begin{aligned}(\mu_{e_t}, \sigma_{e_t}) &= f_e\left(x_t; \theta_e\right)\\ e_t &= \mu_{e_t} + \sigma_{e_t} \odot \varepsilon_{e_t}, \quad \varepsilon_{e_t} \sim N(0,1)\end{aligned} \tag{3}$$

where x_t is the input of FECM and f_e denotes a trainable neural network parameterized by θ_e (future expectation cognition encoder). The sampled future expectation cognition is then fed into the future expectation cognition decoder to further generate the inputs for IAV or JAV.

1) Future Expectation Cognition Learning: We expect FECM to construct future expectation cognition based on current information. Intuitively, conditioning e_t on current information without a specific guide to achieve this goal is difficult. Under the CTDE paradigm, the whole episode's information in an episode is available in the training phase. Therefore, we propose an information-theoretic objective for maximizing the MI $\mathcal{I}(e_t; \tau_t^f \mid x_t)$ between future expectation cognition e_t and future trajectory τ_t^f given x_t, where $\tau_t^f = (x_{t+1}, x_{t+2}, \dots, x_T)$ is the future trajectory information at time t. The term $I(e_t; \tau_t^f \mid x_t)$ is known as

$$\begin{aligned}\mathcal{I}(e_t; \tau_t^f \mid x_t) &= \mathcal{H}(e_t \mid x_t) - \mathcal{H}(e_t \mid \tau_t^f, x_t)\\ &= \mathcal{H}(e_t \mid x_t) - \mathcal{H}(e_t \mid \tau_{t-1}^f)\end{aligned} \tag{4}$$

where $\mathcal{H}$ represents the entropy. Since $\mathcal{H}(e_t \mid \tau_{t-1}^f)$ is intractable for nontrivial mappings, the direct calculation of $\mathcal{I}(e_t; \tau_t^f \mid x_t)$ is currently unavailable. Therefore, we introduce a variational distribution $q_\xi(e_t \mid \tau_{t-1}^f)$ parameterized by ξ as a proxy for the posterior over e_t [9], [39], which derives a tractable lower bound for the MI objective

$$\begin{aligned}\mathcal{I}(e_t; \tau_t^f \mid x_t) &= \mathbb{E}_{e_t, \tau_{t-1}^f}\left[\log \frac{p(e_t \mid \tau_{t-1}^f)}{p(e_t \mid x_t)}\right]\\ &= \mathbb{E}_{e_t, \tau_{t-1}^f}\left[\log \frac{q_\xi(e_t \mid \tau_{t-1}^f)}{p(e_t \mid x_t)}\right]\\ &\quad + D_{\mathrm{KL}}(p(e_t \mid \tau_{t-1}^f) \| q_\xi(e_t \mid \tau_{t-1}^f))\\ &\geq \mathbb{E}_{e_t, \tau_{t-1}^f}\left[\log \frac{q_\xi(e_t \mid \tau_{t-1}^f)}{p(e_t \mid x_t)}\right]\\ &= \mathbb{E}_{e_t, \tau_{t-1}^f}\left[\log q_\xi(e_t \mid \tau_{t-1}^f)\right] + H(e_t \mid x_t).\end{aligned} \tag{5}$$

Since the future expectation cognition encoder is conditioned on current information x_t, the distributions of future expectation cognition $p(e_t)$ are independent of the future trajectory τ_t^f, which follows:

$$\begin{aligned}&\mathbb{E}_{e_t, \tau_{t-1}^f}\left[\log q_\xi(e_t \mid \tau_{t-1}^f)\right]\\ &= \mathbb{E}_{\tau_{t-1}^f}\left[\int p(e_t \mid \tau_{t-1}^f) \log q_\xi(e_t \mid \tau_{t-1}^f) \mathrm{d}e_t\right]\\ &= \mathbb{E}_{\tau_{t-1}^f}\left[\int p(e_t \mid x_t) \log q_\xi(e_t \mid \tau_{t-1}^f) \mathrm{d}e_t\right]\\ &= -\mathbb{E}_{\tau_{t-1}^f}\left[\mathcal{CE}\left[p(e_t \mid x_t) \| q_\xi(e_t \mid \tau_{t-1}^f)\right]\right]\end{aligned} \tag{6}$$

where $\mathcal{CE}$ means cross entropy. Therefore, we have

$$\begin{aligned}\mathcal{I}(e_t; \tau_t^f \mid x_t) \geq -\mathbb{E}_{\tau_{t-1}^f}&\left[\mathcal{CE}\left[p(e_t \mid x_t) \| q_\xi(e_t \mid \tau_{t-1}^f)\right]\right]\\ &+ H(e_t \mid x_t).\end{aligned} \tag{7}$$

The lower bound matches the exact MI when the following equations hold:

$$D_{\mathrm{KL}}(p(e_t \mid \tau_{t-1}^f) \| q_\xi(e_t \mid \tau_{t-1}^f)) = 0. \tag{8}$$

On this basis, we can derive the following minimization objective:

$$L_{\mathrm{MI}}\left(\theta_e,\xi\right)=\mathbb{E}_{\tau_t^f\sim\mathcal{B}}\left[D_{\mathrm{KL}}\left[p\left(e_t\mid x_t\right)\|q_\xi(e_t\mid\tau_{t-1}^f)\right]\right] \tag{9}$$

where $\mathcal{B}$ is the replay buffer, and $D_{\mathrm{KL}}[\cdot\|\cdot]$ is the Kullback–Leibler divergence operator.

For accurate learning of q_ξ, we design a FERM to encode the future trajectory information. As shown in Fig. 2, GRU receives time series τ_{t-1}^f (combinations of τ_t^f and x_t) as input and then outputs the hidden future state h_t^f. It is noted that the time series is input in reverse chronological order. This is because future information sampled at larger time steps shares fewer environmental dynamics correlations than that sampled at time steps closer to t. In other words, the distant future is illusory, but the next few steps are foreseeable and more meaningful for the present.

Expectation is an important aspect of human cognition [23] that allows individuals to navigate their environment effectively and make informed decisions. In decision-making, individuals anticipate the potential consequences of their choices to guide their decision-making process. In MARL, the potential consequences are reflected in the expected return. Therefore, we design a return-based loss to construct expectation representation. As shown in Fig. 2, the embedding of FERM is sampled from the variational posterior distribution $q_\xi(e_t\mid\tau_{t-1}^f)$, and then fed into a linear neural network to estimate the expected discounted return and output G_t'. The reparameterization trick is applied to ensure the gradient is tractable for the sampling operation. Thus, the loss of FERM is

$$\mathcal{L}_{\mathrm{RB}}(\xi)=\mathbb{E}_{<G_t,G_t'>\sim\mathcal{B}}[(G_t'-G_t)^2]. \tag{10}$$

B. Future Expectation Cognition in IAV

Socially, future expectation cognition improves human behaviors by influencing their evaluation of strategies. As a result, we additionally propose a future expectation cognition Q-function Q^F. In teamwork, different members will show different levels of concentration on future expectation cognition, e.g., leaders will share more but followers less. Therefore, we let agents adaptively decide the focus level by decomposing Q_i as

$$Q(a_i\mid\tau_i)=Q^B(a_i\mid\tau_i)+Q^F(a_i\mid\tau_i) \tag{11}$$

where Q^B is the basic Q-function among agents.

Here, FECM uses its observation information with $x_t^i=(o_t^i)$ as input, and then future expectation cognition encoder generates an embedding distribution. We then sample a future expectation cognition vector e_t and input it to the decoder, which generates the parameters of the local utility network. In centralized training phase, $\tau_{t-1}^{f_i}=(x_T^i,x_{T-1}^i,\ldots,x_t^i)$ is fed into a GRU in reverse chronological order. After several steps of computation, FERM then offers the variational posterior distribution $q_{\xi_1}(e_t^i\mid\tau_{t-1}^{f_i})$. We maximize an MI objective $I(e_t^i;\tau_t^{f_i}\mid x_t^i)$ for each agent according to (9), derived an MI regularizer L_{IMI}. Since $I(e_t^i;\tau_t^{f_i}\mid x_t^i)$ is a low bound of $I(a_t^i;\tau_t^{f_i}\mid x_t^i)$ (see the proof in the Appendix), pursuing this objective also maximizes the MI $I(a_t^i;\tau_t^{f_i}\mid x_t^i)$ that can be formulated as

$$I(a_t^i;\tau_t^{f_i}\mid x_t^i)=H(a_t^i\mid x_t^i)-H(a_t^i\mid\tau_{t-1}^{f_i}). \tag{12}$$

Since a_t^i is deterministic given $\tau_{t-1}^{f_i}$, we have $H(a_t^i\mid\tau_{t-1}^{f_i})=0$. Therefore, we have

$$I(a_t^i;\tau_t^{f_i}\mid x_t^i)=H(a_t^i\mid x_t^i) \tag{13}$$

a $H(a_t^i\mid x_t^i)$ measures agent i' ability to explore various behaviors, which encourages the agent to show various behaviors, and therefore our method explores the environment better.

C. Future Expectation Cognition in JAV

The mixing network uses the IAVs of all agents as input and mixes them monotonically, providing the values of $Q_{\mathrm{tot}}(s,\vec{a})$ and optimizing the following TD loss:

$$\mathcal{L}_{\mathrm{TD}}(\theta)=\sum_{i=1}^{b}\left[(r+\gamma\max_{\vec{a}'}Q_{\mathrm{tot}}(s',\vec{a}';\theta^-)\right.$$
$$\left.-Q_{\mathrm{tot}}(s,\vec{a};\theta))^2\right] \tag{14}$$

where θ^- are the parameters of a periodically updated target network. The term $\gamma\max_{\vec{a}'}Q_{\mathrm{tot}}(s',\vec{a}';\theta^-)$ estimates expected future return.

Many value decomposition methods try to complicate the mixing network to strengthen the representation ability, such as QPLEX [31] and Qatten [32]. The ability of JAV to build better estimates of the JAV functions directly results in better policy estimates and faster learning. The estimation objective can be divided into two parts: immediate reward r and expected future return $\gamma\max_{\vec{a}'}Q_{\mathrm{tot}}(s',\vec{a}';\theta^-)$. The immediate reward can be estimated easily since there is a certain mapping mechanism from the current state to the reward. However, due to the uncertain and unknown future, deriving an accurate estimate of future return from the current state is challenging, especially in the early training phase. As a result, the essence limiting JAV's performance may not be its representation ability but rather its ability to estimate expected future returns. To alleviate this problem, we introduce future expectation cognition into the mixing network to provide faster and more reliable learning.

In JAV, FECM takes the global state $x_t=s_t$ as input and then generates future expectation cognition e_t, which are input into the decoder network and finally generates the final parameters to calculate $Q_{\mathrm{tot}}(\tau,\vec{a})$. Here is also an MI regularizer L_{JMI} according to (9) to learn future expectation cognition.

D. Overall Optimization Objective

In this section, we present the training algorithm of QFuture. The details are shown in Algorithm 1. We have introduced optimization objectives for future expectation cognition learning in IAV and JAV. The final learning objective of QFuture is

$$\mathcal{L}(\theta)=\mathcal{L}_{\mathrm{TD}}(\theta)+\beta_I\mathcal{L}_{\mathrm{IMI}}(\theta_e^I,\xi^I)$$
$$+\beta_J\mathcal{L}_{\mathrm{JMI}}(\theta_e^J,\xi^J)+\mathcal{L}_{\mathrm{RB}}(\xi) \tag{15}$$

Algorithm 1 The training algorithm of QFuture

```
Input: batch size $B$, number of agents $n$, steps per episode $T$, episodes $K$;
Initialize: Replay buffer $\mathcal{B}$, the parameters of the mixing network $\theta$; the parameters of the target network $\theta^-$: $\theta^- = \theta$;
for $k = 1$ to $K$ do
    for $t = 0, \cdots, T$ do
        Collect the global state $s^t$
        for each agent $i$ do
            Collect the local observation $o_i^t$;
            Estimate the local action-value $Q_i^t(o_i^t, a_i^t)$ for each agent with the individual local action-value network and choose greedy action or random action with probability $\epsilon$;
        end for
        Execute the joint action $\boldsymbol{a}^t$ and collect the next joint observation $\boldsymbol{o}^{t+1}$, next state $s^{t+1}$, and reward $r^{t+1}$;
        Store $(\boldsymbol{o}^t, \boldsymbol{a}^t, r^{t+1}, \boldsymbol{o}^{t+1}, s^t, s^{t+1})$ into the buffer $\mathcal{B}$;
    end for
    sample a random mini-batch from $\mathcal{B}$;
    Calculate the loss of future expectation cognition in IAV $\mathcal{L}_{\text{IMI}}(\theta_e^I, \xi^I)$ and JAV $\mathcal{L}_{\text{JMI}}(\theta_e^J, \xi^J)$ with variational inference;
    Calculate return-based loss $\mathcal{L}_{\text{RB}}(\xi)$ in FRM;
    Estimate the joint action-value $Q_{\text{tot}}^{\pi}(s^t, \vec{a}^t)$, calculate the update target $y_{\text{total}}$, and calculate the TD loss.
    Update the parameters of the whole network by minimizing the total loss $\mathcal{L}(\theta)$ with gradient descent algorithm, $\mathcal{L}(\theta) = \mathcal{L}_{\text{TD}}(\theta) + \beta_I \mathcal{L}_{\text{IMI}}(\theta_e^I, \xi^I) + \beta_J \mathcal{L}_{\text{JMI}}(\theta_e^J, \xi^J) + \mathcal{L}_{\text{RB}}(\xi)$;
    Update the target network with fixed interval;
end for $= 0$
```

where $\theta = (\theta_e^I, \theta_e^J, \xi)$ are the parameters of the whole framework; θ_e^I and θ_e^J represent the parameters of future expectation cognition encoder in IAV and JAV, respectively; $\xi = (\xi^I, \xi^J)$ are the parameters of FERM; $\mathcal{L}_{\text{IMI}}$ and $\mathcal{L}_{\text{JMI}}$ represent the MI regularizers in IAV and JAV, respectively; ξ^I and ξ^J represent the parameters of FERM in IAV and JAV, respectively; β^I and β^J are scaling factors.

IV. Experimental

A. Experimental Setup

To evaluate the effectiveness of QFuture, as shown in Fig. 3, we conduct experiments with different scenarios on two challenging benchmarks, i.e., SMAC [27][1] and GRF [28]. In these tasks, QFuture is compared with Qtran [35], QMIX [30], QPLEX [31], and Qatten [32], all of which can be implemented on both SMAC and GRF. For evaluation, each method is conducted with four different seeds. Test-winning rates are chosen to better compare the effectiveness and superiority of different methods.

(a) (b)

Fig. 3. Snapshoots of (a) SMAC and (b) GRF.

TABLE I
SMAC Challenges

Task	Ally Units	Enemy Units
Corridor	6 Zealots	24 Zerglings
3s_vs_8z	3 Stalkers	8 Zealots
3s5z_vs_3s6z	3 Stalkers, 5 Zealots	3 Stalkers, 6 Zealots
3s5z_vs_3s7z	3 Stalkers, 5 Zealots	3 Stalkers, 7 Zealots
MMM2	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 2 Marauders, 8 Marines

SMAC is built on the prevalent real-time strategy game StarCraft II [41], which is a popular real-time strategy game that derives many micromanagement scenarios. In the micromanagement scenarios, the agents need to cooperate to eliminate the enemies. This benchmark consists of various maps classified as easy, hard, and super hard. We test our method on five super hard micromanagement tasks, i.e., *MMM2*, *corridor*, *3s8z*, *3s5z_vs_3s6z*, *3s5z_vs_3s7z*. Details of these maps are shown in Table I. There is a shaped reward based on the hit-point damage on the enemies and a special incentive for winning the battle (SMAC default dense reward setting). The detailed reward of each scenario is defined as follows:

$$r_t = \frac{\sum_{k=1}^{N} \Delta h_k^t + N_{\text{death}}^t \ \times r_{\text{kill}}}{\sum_{k=1}^{N} H_{\text{total}}^k \ + N \times r_{\text{kill}} \ + r_{\text{win}}} \tag{16}$$

where N represents the number of enemies. N_{death}^t represents the number of enemies died at step t. H_{total}^k total is the total health of enemy k. $\Delta h_k^t = h_k^t - h_k^{t-1}$ is the health difference of enemy k between two steps. r_{kill} and r_{win} are the special bonuses for killing the enemy and winning the battle, which are set as 10 and 200, respectively.

1) GRF Tasks: In GRF, agents are trained to play football in a physics-based 3-D simulator. GRF is a challenging task for its inner stochasticity and sparse reward. The agents must learn high-level cooperation skills such as passing, obstructing opponents for teammates et al., and then score a goal. We choose five academy tasks (three official and two hand-crafted) to evaluate our method, i.e., *run pass and shoot with keeper*, *3vs1 with keeper*, *counterattack hard*, *3v3,* and *3v4*.

The initial positions of players, opponents, and the ball are shown in Fig. 4. In these tasks, we control the left team, where each agent must choose an action from 19 available actions, including run, pass, dribble, shot, etc. All agents must cooperate well to organize offenses and seize fleeting opportunities. There

[1] We use SC2.4.10 version.

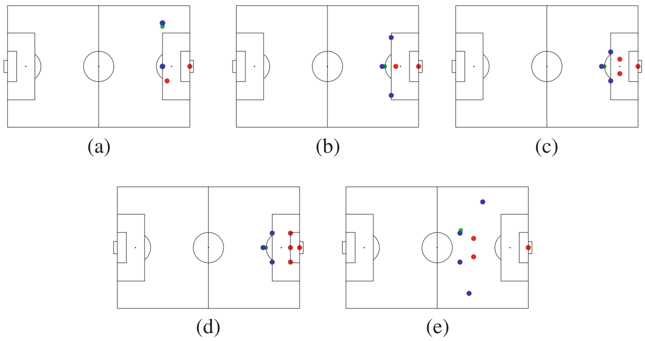


Fig. 4. Visualization of the initial position of each agent in five GRF tasks. Blue dots represent the agent. The red dots are opponents, and the green dot denotes the ball. (a) Run pass and shootwith keeper. (b) 3 versus 1 with keeper. (c) 3 versus 3. (d) 3 versus 4. (e) Counter attack hard.

are only two types of rewards: 1) a reward of +0.5 for the first time the team gets the ball; and 2) a reward +10 for the left team to score a goal. An episode will be terminated, reaching the following four situations: 1) the ball controlled by opponents; 2) the ball returning to the left half-court; 3) scoring a goal; and 4) the ball bouncing out of fields. The observation contains the positions and directions of the ego-agent, teammates, and the ball. The original observation data will induce explosive gradients in the agent utility network. To address this problem, we normalize all the observation data in the $[-1, 1]$.

2) Architecture, Hyperparameters, and Infrastructure: In this article, we compare our approach with five value-based methods. QFuture is developed based on the QMIX. QFuture and QMIX use the same code framework.[2] QFuture is also implemented based on this code framework. Except for the additional parameters in QFuture, all other parameters are set the same as QMIX, such as batch size, learning rate, parallel environments, etc. For QPLEX, Qatten, and Qtran, we use the code framework provided by PYMARL2,[3] and we use the default training settings in SMAC tasks. For GRF tasks, we ensure the same environmental settings as QFuture, including reward, observation, state settings, etc.

In QFuture, we introduce two important hyperparameters: β_I and β_J as in (15), correlated to the MI regularizers. For SMAC scenarios, we search the best hyperparameters on *MMM2*, and use $\beta_I, \beta_J = 0.01, 0.05$ on all five tasks. For GRF scenarios, we also search the best hyperparameters and use $\beta_I, \beta_J = 0.02, 0.05$ on *run pass and shoot with keeper* and *3vs1 with keeper*, $\beta_I, \beta_J = 0.1, 0.05$ on *counterattack hard*, *3v3,* and *3v4*. The MI regularizer L_{IMI} can promote exploration ability, so we increase the value of β_I in more challenging GRF tasks. The hyperparameters of FECM are shown in Table II.

B. Comparison Studies

In this section, we first carry out the experiments on SMAC, and the average results are shown in Fig. 5. These experimental results show that our method outperforms all alternative baselines with acceptable variance across random seeds on all maps. Our method is developed based on the QMIX and QFuture significantly and constantly improves the learning performance and outperforms QMIX. Specifically, in *Corridor*, *3s5z_vs_3s6z* and *3s5z_vs_3s7z*, QMIX all fails to learn effective strategy with 0% win rate. The baselines QPLEX and Qatten can achieve satisfactory performance on some tasks, such as MMM2 and Corridor. In *3s5z_vs_3s6z*, QPLEX learns effective strategies earlier than QFuture, but QFuture learns faster than QPLEX. Qtran fails to show progress on all tasks. On our handcrafted map *3s5z_vs_3s7z*, only QFuture can explore an effective strategy. Overall, the comparison studies on SMAC show the success of QFuture in learning performance improvements.

TABLE II
HYPERPARAMETERS OF FECM

Hyperparameter	Value
Output dimension of the encoder	64
Output dimension of the decoder	64
Number of layers of hidden layers	2
Dimension of the hidden layer	64
Activation function	ReLU
Optimizer	RMSprop
Learning rates	$5e-4$

To further evaluate the proposed method in cooperative tasks, we conduct experiments on five GRF tasks. Unlike SMAC tasks, GRF tasks highly test the exploration ability of the algorithm since agents cannot get effective feedback in the early training phase due to the sparse reward. Only the agent exploring goal strategy can then know their learning objective. As depicted in Fig. 6, QFuture achieves superior performance on all tasks, whose curves rise earliest and fastest. Among these baselines, QMIX delivers a relatively better strategy. Compared to QMIX, our method shows noticeable performance promotion. In *3vs4*, QMIX traps in optimum local strategy, whereas QFuture leaps out quickly. Although QPLEX and Qatten behave better than QMIX in SMAC tasks, they only show slightly effective performance here. Qtran only delivers meaningful learning in *run pass and shoot with keeper*, but fails on other tasks.

Additionally, it is crucial to acknowledge that the incorporation of FECM introduces an additional time overhead to our method. To evaluate the time complexity, we measured the runtime of our program during experiments. Specifically, we compared the average execution time of QFuture and QMIX by conducting four training runs over 2.5 million time steps in the scenario of *3 vs 1 with keeper*. The computational analysis reveals that the average time taken by QMIX is 7.9 hours, while QFuture increases to 8.7 hours, resulting in a 10.1% increase in time consumption. We consider this increase to be acceptable, considering the significant improvement it brings to the agents' performance.

C. Ablations

To thoroughly understand the superior performance of QFuture, we carry out ablation studies to show the contribution of future expectation cognition learning. In particular, we carry out

[2] https://github.com/starry-sky6688/MARL-Algorithms
[3] https://github.com/hijkzzz/pymarl2

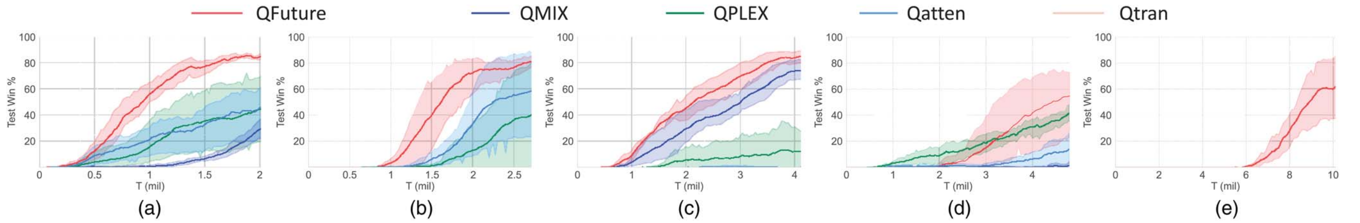


Fig. 5. Comparison of our method against baseline methods on five super hard maps in SMAC with the evaluation index of test-winning rate. Invisible algorithms mean that the results are consistently zero. (a) MMM2. (b) Corridor. (c) 3s_vs_8z. (d) 3s5z_vs_3s6z. (e) 3s5z_vs_3s7z.

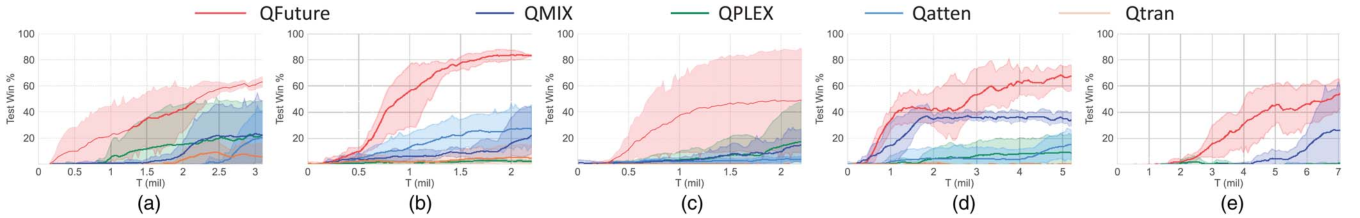


Fig. 6. Comparison of our method with baseline methods on five academy tasks in GRF with the evaluation index of test-winning rate. Invisible algorithms mean that the results are consistently zero. (a) Run pass and shoot with keeper. (b) 3 vs 1 with keeper. (c) 3 vs 3. (d) 3 vs 4. (e) Counter attack hard.

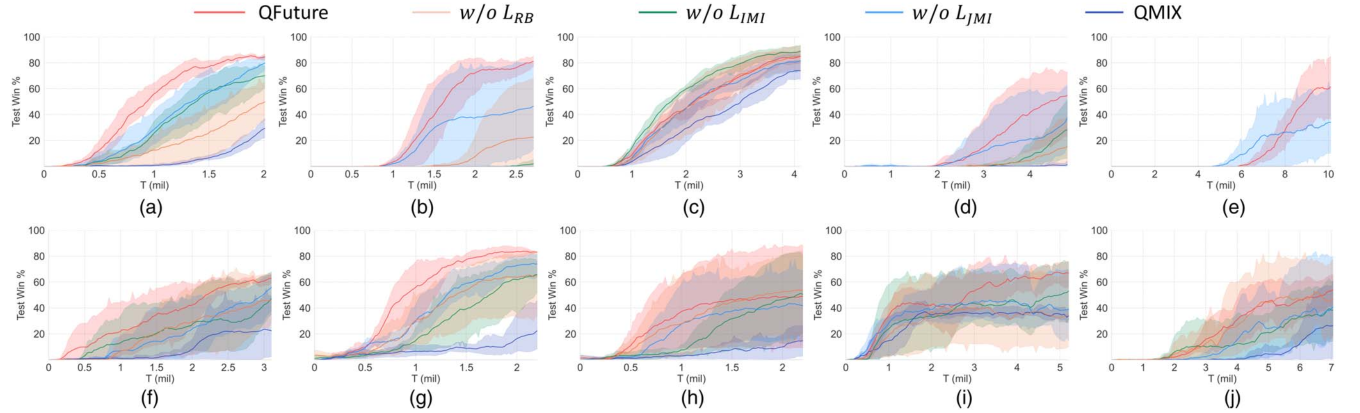


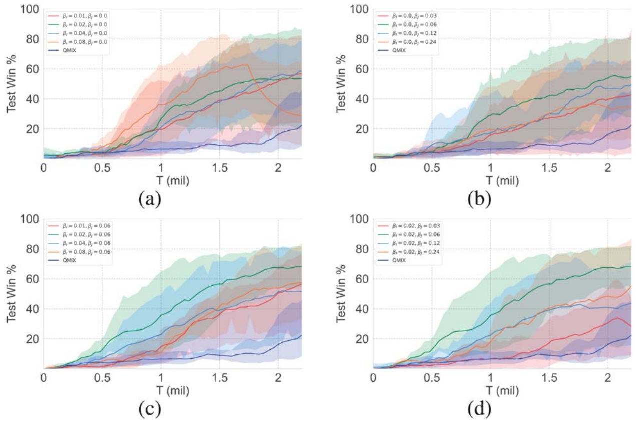
Fig. 7. Winning rate of ablation studies on all tasks. Invisible algorithms mean that the results are consistently zero. (a) MMM2. (b) Corridor. (c) 3s_vs_8z. (d) 3s5z_vs_3s6z. (e) 3s5z_vs_3s7z. (f) Run pass and shoot with keeper. (g) 3 vs 1 with keeper. (h) 3 vs 3. (i) 3 vs 4. (j) Counter attack hard.

the following ablation studies: 1) $w/o\ L_{\text{IMI}}$: the QFuture without (abbreviated as w/o) MI regularizer in IAV; 2) $w/o\ L_{\text{JMI}}$: the QFuture without MI regularizer in JAV; and 3) $w/o\ L_{\text{RB}}$: the QFuture without return-based regularizer in IAV and JAV's FERM.

As shown in Fig. 7, QFuture offers the best learning performance on nearly all tasks. In addition, deleting any regularizer can still lead to better learning performance than QMIX. Separation of any loss will yield degeneracy in learning quality. These results convincingly demonstrate the necessity of each designed regularizer.

Removing return-based loss L_{RB} in FERM will bring the most noticeable performance decadence in SMAC tasks. This detachment will induce adverse effects for FERM, which may fail to extract helpful information from the future trajectory and then influence the future expectation cognition of learning in both IAV and JAV. However, in GRF tasks, $w/o\ L_{\text{RB}}$ only shows slight performance degeneration compared to those in SMAC tasks. Since GRF is a sparse reward scenario, FERM cannot receive practical training due to the lack of feedback, especially at the early training stage, which hinders the learning of future expectation cognition. However, SMAC tasks provide dense rewards for each frame, which induce more effective future expectation representation learning and contribute to efficient learning in IAV and JAV. These performance gaps in different scenarios demonstrate the importance of return-based loss in FERM. In addition, we show how to improve the performance of QFuture in sparse reward tasks in Section IV-F.

In GRF tasks, due to the sparse reward, the exploration ability of the method plays a vital role in the final learning performance. In Section III-B, our analysis shows that maximization of $I(e_t^i; \tau_t^{f_i} \mid x_t^i)$ realizes the additional objective $I(e_t^i; \tau_t^{f_i} \mid x_t^i)$ that can promote exploration. Therefore, deleting L_{IMI} will



(a) (b) (c) (d)

Fig. 8. Training curves of different β_I and β_J. (a)–(d) 3 vs 1 with keeper.

weaken the exploration ability. As shown in Fig. 7(f)–7(j), compared to $w/o\ L_{\text{RB}}$ and $w/o\ L_{\text{JMI}}$, $w/o\ L_{\text{IMI}}$ shows the worst learning quality, particularly in the early stages, where $w/o\ L_{\text{IMI}}$ usually spends more training steps to explore a strategy with 20% win rate. Furthermore, as shown in Fig. 7(b), 7(d), and 7(e), $w/o\ L_{\text{IMI}}$ fails to learn effective strategy in these three maps, which are harder than *MMM2* and *3s_vs_8z*. Since exploration ability is essential in harder tasks, the performance gap between QFuture and $w/o\ L_{\text{IMI}}$ reveals the ability to explore the environment.

The learning curves of $w/o\ L_{\text{JMI}}$ show outstanding performance in most tasks. The comparison of QFuture performance and $w/o\ L_{\text{JMI}}$ proves that the regularizer L_{JMI} effectively reduces performance variance between different random seeds, particularly in SMAC tasks. Furthermore, as shown in Fig. 7(b), 7(d), and 7(e), QFuture offers faster promotion of the strategy among agents after exploring an effective strategy. This may be because the mixing network with future expectation cognition provides more reasonable credit assignments.

D. Sensitivity Analysis of β_I and β_J

In order to further analyze the influence of weight parameters β_I and β_J on the performance of the proposed method, simulation experiments were conducted on the *3 vs 1 with keeper* scenario with different β_I and β_J. The curve of the average winning rate is shown in Fig. 8. Fig. 8 indicates that when $\beta_J = 0$, there is no significant change in performance when β_I ranges from 0.01 to 0.02 to 0.04. However, when β_I increases from 0.04 to 0.08, although the early stage performance improves significantly, the later-stage performance decreases notably. This may be because excessively large β_I causes the agent's strategy to excessively focus on the future, leading to the failure of the agent's strategy due to future uncertainties. Fig. 8 shows that when $\beta_I = 0$, the performance increases with the increase of β_J from 0.03 to 0.06. This is because the increase of β_J enables the mixing network to pay more attention to the future when estimating the joint Q-values Q_{tot}, resulting in a more accurate estimation of future cumulative returns. However, too large β_J may cause the mixing network to lose focus on the current information and result in performance degradation. Therefore, we observe that as β_J increases from 0.06 to 0.12 to 0.24, the policy performance gradually decreases. In Fig. 8, we conducted experiments by changing β_I under the setting of the best $\beta_J = 0.06$. The performance is significantly improved when β_I ranges from 0.01 to 0.02, while it decreases significantly when β_I increases from 0.02 to 0.04 to 0.08. In Fig. 8, we conducted experiments by changing β_J under the setting of $\beta_I = 0.02$, the results of which are consistent with those in Fig. 8. In conclusion, the performance of the proposed method is influenced to a certain extent by β_I and β_J, and the influence of the two parameters is relatively balanced.

Additionally, as shown in Fig. 8, we have tried fifteen different parameter combinations for β_I and β_J. We can observe that QFuture demonstrates excellent learning performance across all these parameter settings, achieving significantly better results than QMIX. This indicates that our method exhibits robustness and does not overly rely on parameter adjustments. Even under inappropriate parameter settings, it can still perform well.

E. Visualization of Future Expectation Cognition

In this section, we visualize the learned strategy and its future expectation cognition, shown in Fig. 9. We choose *counter attack hard*, the most challenging GRF task, where we control four left team players, i.e., $p7, p8, p9$, and $p10$, with others controlled by the built-in rule. As shown in Fig. 9(a) and 9(d), the green and red trajectories represent the left and right team's whole episode movement, respectively. The blue points denote the ball's location. We show the pitch control heat map [42], [43] at the final step, with 1 for entirely left team dominance at a given position in the pitch and 0 for the entire right team dominance, whereas the length of the vector represents the player's velocity at the final time step. To show the learned strategy delicately, we provide the learned strategy's video in https://sites.google.com/view/qfuture. In Fig. 9(b), 9(c), 9(e), and 9(f), dots with the same color denote the same event phase in an episode. The number close to the dot is the time step located in the strategy.

1) Strategy Learned by QFuture: As shown in Fig. 9(a), this strategy can be divided into five event phases (see more details in videos): 1) From steps 0 to 9, the player $p8$ gets ball possession and then dribbles the ball to make space for $p9$. 2) From steps 10 to 17, $p8$ gives a ground pass to $p9$ after successfully distracting two defensive players, and $p9$ runs to the expected pass position. 3) From steps 18 to 30, $p9$ gives a one-touch pass directly to the penalty box in step 18, while $p7$ dashes forward to the expected ball impact point. 4) From steps 32 to 35, the ball reaches the penalty box at step 32, and $p7$ keeps running, trying to give a quick attack. 5) From steps 36 to 39, p7 gives a one-touch shot facing the block of a goalkeeper and finally scores a goal, spending four steps from shot to goal.

2) Strategy Learned by $w/o\ L_{IMI}$: As illustrated in Fig. 9(d), this strategy can be decomposed into six event phases (see more details in videos): 1) From step 0 to 3, player $p8$ gets ball possession. 2) From steps 4 to 15, $p8$ waits motionlessly while $p7$ sprints to the penalty box. 3) From step 16 to 32, $p8$ dribbles

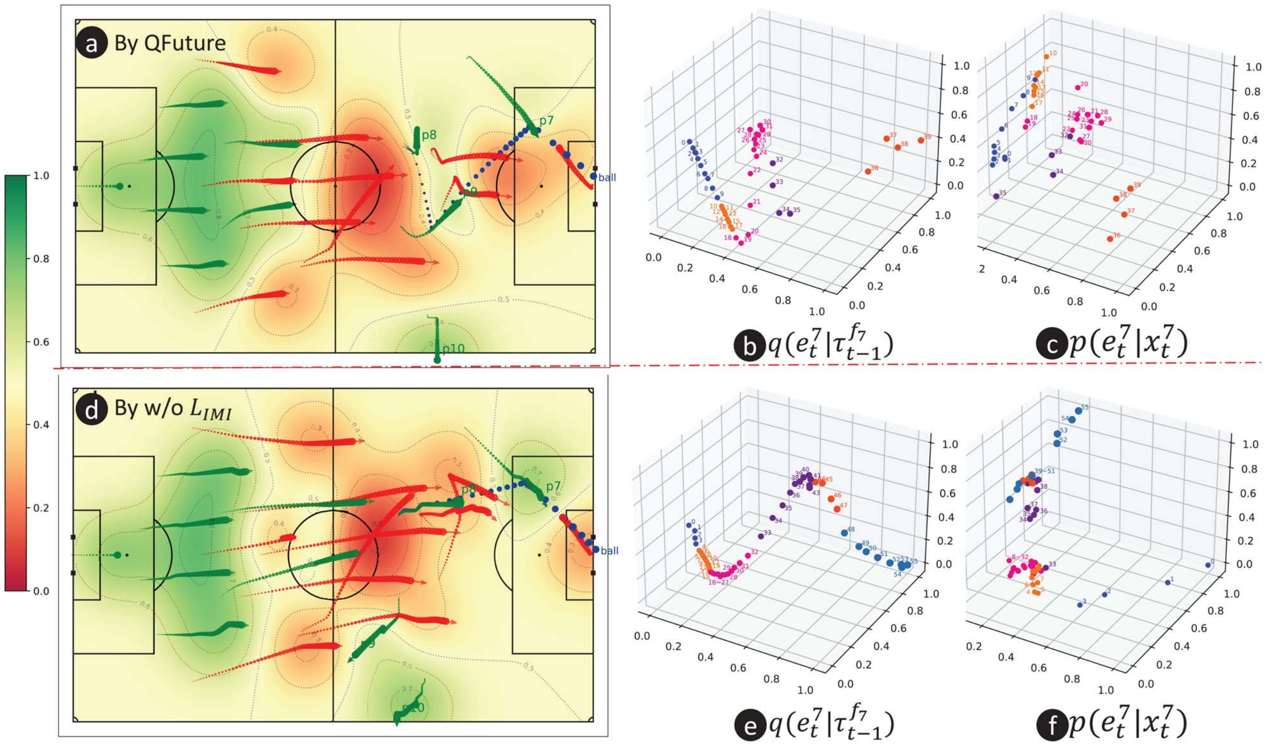


Fig. 9. Visualization of the learned strategies, the future expectation representation distributions $q(e_t^7|\tau_{t-1}^{f_7})$ and future expectation cognition $p(e_t^7|x_t^7)$. Number 7 represents player $p7$. (a) and (d) Strategies learned by QFuture and w/o L_{IMI}, respectively. (b), (c), (e), and (f) Principal components of the sampling vector at each step after linear PCA, corresponding to the strategy in (a) and (d), respectively.

the ball and waiting for $p7$ running to appropriate location. 4) From steps 33 to 43, $p7$ stands still, and $p8$ gives him a ground pass. 5) From steps 44 to 47, $p7$ receives the ball at step 44 and dribbles the ball to the goal area. 6) From steps 48 to 55, $p7$ shots and then gets a score, spend eight steps from shot to goal.

3) Whether Learned Future Expectations Representation Meaningful: In this article, FERM models future expectations based on future trajectories, and the future expectations we hope to learn are intentionally in line with the laws of human future expectations. Socially, there are four laws of human future expectations in events.

1) Future expectations progress and tend to be consistent at different times when considering the same event.
2) Future expectations can be inconsistent when different events are considered.
3) Future expectations encounter a notable turning point when important events happen.

In Fig. 9(b), we visualize the learned future expectation representation distribution $q(e_t^7|\tau_{t-1}^f)$ of player 7, which is the leading player of this score. The location of the dots represents the contents of future expectations. Above mentioned three laws can be concluded from this figure. First, the positions of dots in the same event phase are nearly in line with time, meaning future expectations is consistent. Second, the dots in different event phases will not overlap, corresponding with law 2. Third, the dots in the junction of two event phases will show the discontinuity or turning points (transitions). There are mainly three transitions, i.e., (18,19,20), (30,31,32), and (34,35,36). Transitions happen when $p7$'s situation changes. He observes the ball passed into his region in the first transition, prepares to receive the ball in the second transition, and completes a shot in the third transition. These transitions accord with our understanding of football that a progressive pass could change the situation on the field and demonstrate law 3.

In Fig. 9(e), we also visualize the learned future expectation representation distribution $(q(e_t^7|\tau_t^{f_7}))$. Clearly, the orbit of these dots continues to follow the aforementioned three laws. Particularly, there are two transitions, i.e., (16,17,18) and (43,44,45).

These results demonstrate that learned future expectation is reasonable and meaningful.

4) How MI Regularizer Helps Build Future Expectation Cognition: As shown in Fig. 9(c), although the learned future expectation cognition e_t^7 is derived from $p7$'s local observation, there are still similar dots distribution with $q(e_t^7|\tau_{t-1}^{f_7})$, which are indicative of three laws. These dots are chronically consistent, reflecting law 1. Specifically, e_t also captures three transitions in this strategy. As shown in Fig. 9(f), most of the dots mass as a cluster rather than being spread over a line when the MI regularizer is removed from IAV. Dots in different event phases overlap, conflicting with law 2. Furthermore, two transitions in Fig. 9(e) are difficult to capture in Fig. 9(e). These results show that the future expectation cognition is difficult to be built without L_{IMI} regularizer. The difference between

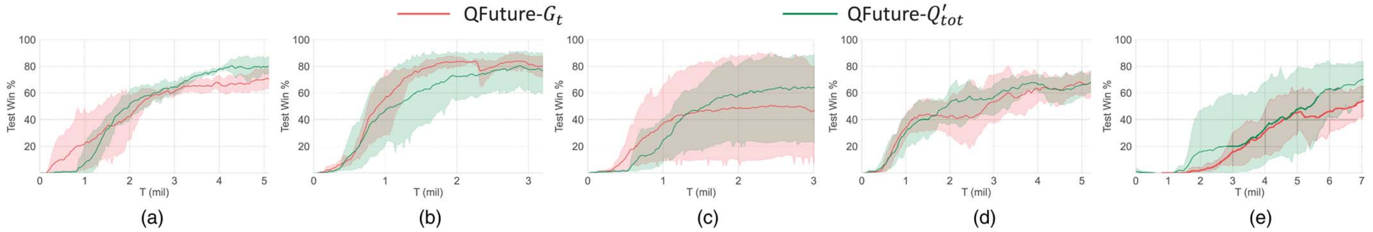


Fig. 10. Comparison of QFuture-G_t against QFuture-Q'_{tot} on five academy tasks in GRF with the evaluation index of test-winning rate. (a) Run pass and shoot with keeper. (b) 3 vs 1 with keeper. (c) 3 vs 3. (d) 3 vs 4. (e) Counter attack hard.

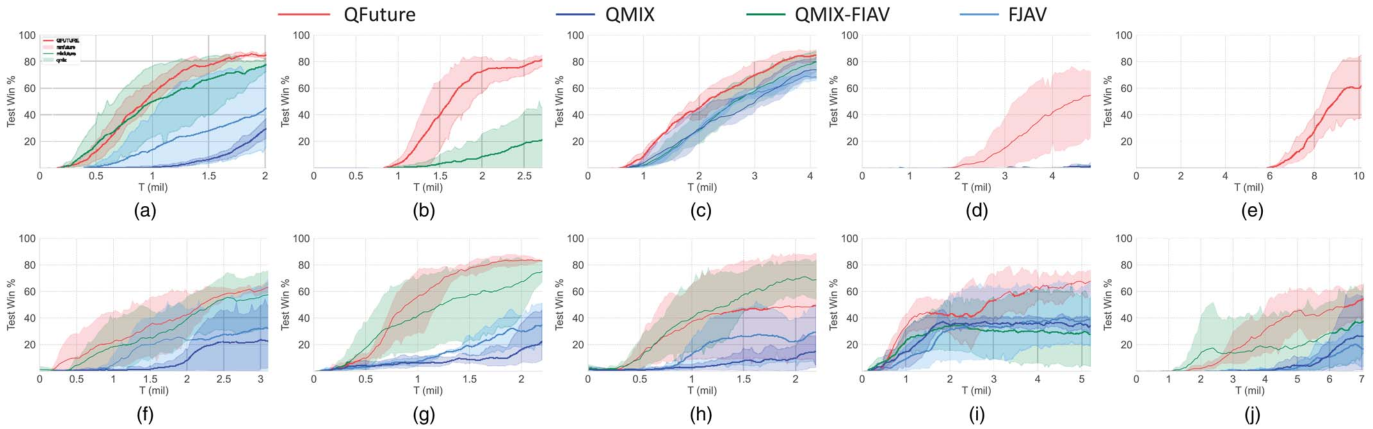


Fig. 11. Winning rate of QFuture, QMIX, QMIX-FIAV, and FJAV on all tasks. Invisible algorithms mean that the results are consistently zero. (a) MMM2. (b) Corridor. (c) 3s_vs_8z. (d) 3s5z_vs_3s6z. (e) 3s5z_vs_3s7z. (f) Run pass and shoot with keeper. (g) 3 vs 1 with keeper. (h) 3 vs 3. (i) 3 vs 4. (j) Counter attack hard.

Fig. 9(e) and 9(f) clearly describes the necessity and effectiveness of the proposed MI regularizer to establish future expectation cognition.

5) How Future Expectation Cognition Benefits Cooperation: By comparing two strategies, QFuture learns a more excellent strategy. It is noted that the strategy learned by QFuture consumes the same training steps with $w/o\ L_{\text{IMI}}$. With future expectation cognition, agents know what to expect next and then how to do best. Both methods learn to pass the ball to $p7$ when he reaches the appropriate position. However, in $w/o\ L_{\text{IMI}}$, $p8$ shows excessive dribbling in event phase (3) before a pass because of waiting for $p7$ to run to the penalty box and stand still. Instead, in QFuture, $p9$ runs to the expected ball landing and gives a one-touch pass directly to the penalty box even if $p7$ does not reach the ball landing position at this time, which releases a one-touch shot finally. Here, players $p7$, $p8$, and $p9$ show two advanced cooperation skills owing to future expectation cognition, i.e., running to receive a pass and a one-touch pass (shot). With future expectation cognition, the passer can know where his teammates can intercept the ball promptly, and the receiver is qualified to infer the ball's landing and then plan its velocity. To realize a one-touch shot or pass, the agent must plan the destination of the pass before touching the ball (action lag in GRF tasks). Therefore, we observe that the strategy in Fig. 9(d) fails to perform a one-touch pass or shot since future expectation cognition is hard to construct without a specific guide. In general, future expectation cognition helps players achieve closer cooperation with tacit understanding and contribute to a quick attack strategy in Fig. 9(a).

F. Improving QFuture in Sparse Reward Tasks

In ablation studies, our experimental results and analysis indicate that the sparse reward problem will degrade the performance of QFuture by hindering the learning of $q(e_t|\tau_t^f)$. In sparse reward tasks, G_t can only provide effective feedback in successful episodes' training, unfavorable to future expectation cognition learning. To address this problem, we slightly modify QFuture for sparse reward tasks.

In sparse reward tasks, we can change the predicted target in the FERM of the agent network from G_t (denoted QFuture-G_t) to $\max_{\vec{a}'} Q_{\text{tot}}(s', \vec{a}'; \theta^-)$ (denoted QFuture-Q'_{tot}). Once agents can succeed with a specified probability, $\max_{\vec{a}'} Q_{\text{tot}}(s', \vec{a}'; \theta^-)$ can evaluate these failed episodes with the help of successful episode experience. As shown in Fig. 10, QFuture-Q'_{tot} show evident performance promotion in GRF tasks. In Fig. 10(a), 10(c), and 10(e), QFuture-Q'_{tot} performs an increase of over ten percent winning rate at the end. In Fig. 10(b) and 10(d), the winning rate is the same at the end. It is worth noting that QFuture-Q'_{tot} is always worse than QFuture-G_t in the early training phase, but it learns faster after it reaches the winning rate 40%, corresponding to our above analysis.

In addition, in dense reward tasks, G_t can provide more accurate feedback than $\max_{\vec{a}'} Q_{\text{tot}}(s', \vec{a}'; \theta^-)$ on both successful and failed episodes. We also perform experiments on SMAC, QFuture-G_t performs better than QFuture-Q'_{tot} in these tasks.

G. Applying Parts of QFuture to QMIX

In this article, we propose a novel method to introduce future expectation cognition into the calculation of IAV and JAV. We denote the IAV and JAV in QFuture as FIAV and FJAV, respectively. FIAV can be combined with many value-based MARL algorithms, such as QMIX, QPLEX, Qatten, etc. FJAV provides a way of value function decomposition by learning a mixing network. To show the scalability of FIAV, we apply it to QMIX, denoted as QMIX-FIAV. To show the effectiveness of FJAV, we replaced the mixing network in QMIX with FJAV, denoted as FJAV.

As shown in Fig. 11, QMIX-FIAV shows significant improvements to QMIX, demonstrating the scalability of our method. FJAV also shows better performance than QMIX. The splendid performance on QFuture indicates that the combination of FIAV and FJAV will present an advantageous performance over those used separately.

V. Conclusion and Future Work

In this article, we have introduced the concept of future expectation cognition into deep MARL by maximizing a MI objective $I(e_t, \tau_t^f \mid x_t)$. Future expectation cognition is used to generate parameters to estimate individual action-values and JAV. Experimental results demonstrate the superior effectiveness of our method.

To our best knowledge, it is the first article using all future steps information at each step's training in MARL. This utilization can accelerate training and promote collaboration, but it is easily stuck in chronological logical traps. We believe there are other ways to fully advantage of future information to improve learning performance and sample efficiency. Our work may motivate researchers in both the single-agent RL and multiagent RL fields.

Appendix
Mathematical Derivation

A. $I(a_t^i; \tau_t^{f_i} \mid x_t^i)$ and $I(e_t^i; \tau_t^{f_i} \mid x_t^i)$

Given x_t^i, then e_t^i and $\tau_t^{f_i}$ are conditionally independent given a_t^i, since e_t^i can only influence $\tau_t^{f_i}$ through a_t^i. Considering the MI term $I(\tau_t^{f_i}; e_t^i, a_t^i \mid x_t^i)$ which can be decomposed as

$$\begin{aligned} I(\tau_t^{f_i}; a_t^i, e_t^i \mid x_t^i) &= I(\tau_t^{f_i}; e_t^i \mid x_t^i) + I(\tau_t^{f_i}; a_t^i \mid e_t^i, x_t^i) && (17) \\ &= I(\tau_t^{f_i}; a_t^i \mid x_t^i) + I(\tau_t^{f_i}; e_t^i \mid a_t^i, x_t^i). && (18) \end{aligned}$$

Since e_t^i and $\tau_t^{f_i}$ are conditionally independent given a_t^i and x_t^i, we have $I(\tau_t^{f_i}; e_t^i \mid a_t^i, x_t^i) = 0$. Since $I(\tau_t^{f_i}; a_t^i \mid e_t^i, x_t^i) \geq 0$, we have

$$I(\tau_t^{f_i}; a_t^i \mid x_t^i) \geq I(\tau_t^{f_i}; e_t^i \mid x_t^i). \quad (19)$$

Thus, the proof is accomplished.

References

[1] F. Khemakhem, H. Ellouzi, H. Ltifi, and M. B. Ayed, "Agent-based intelligent decision support systems: A systematic review," *IEEE Trans. Cogn. Develop. Syst.*, vol. 14, no. 1, pp. 20–34, Mar. 2022.

[2] Y. Wang, L. Zhang, L. Wang, and Z. Wang, "Multitask learning for object localization with deep reinforcement learning," *IEEE Trans. Cogn. Develop. Syst.*, vol. 11, no. 4, pp. 573–580, Dec. 2019.

[3] S. Na et al., "Federated reinforcement learning for collective navigation of robotic swarms," *IEEE Trans. Cogn. Develop. Syst.*, vol. 15, no. 4, pp. 2122–2131, Dec. 2023.

[4] G. Lample and D. S. Chaplot, "Playing FPS games with deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, pp. 1–7, 2017.

[5] W. Yi, R. Qu, L. Jiao, and B. Niu, "Automated design of metaheuristics using reinforcement learning within a novel general search framework," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 1072–1084, Aug. 2023.

[6] J. Yang, J. Zhang, M. Xi, Y. Lei, and Y. Sun, "A deep reinforcement learning algorithm suitable for autonomous vehicles: Double bootstrapped soft-actor-critic-discrete," *IEEE Trans. Cogn. Develop. Syst.*, vol. 15, no. 4, pp. 2041–2052, Dec. 2023.

[7] M. Hüttenrauch, A. Sosić, and G. Neumann, "Guided deep reinforcement learning for swarm systems," 2017, *arXiv:1709.06011*.

[8] A. J. Singh, A. Kumar, and H. C. Lau, "Hierarchical multiagent reinforcement learning for maritime traffic management," in *Adaptive Agents and Multi-Agent Systems*, 2020.

[9] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, "Maven: Multi-agent variational exploration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[10] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992.

[11] G. A. Rummery and M. Niranjan, *On-Line Q-Learning Using Connectionist Systems*, Eng. Depart., Cambridge Univ., Tech. Rep., 1994.

[12] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 1–7.

[13] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[14] R. S. Sutton et al., "Introduction to reinforcement learning," vol. 2, no. 4, Cambridge, MA, USA: MIT Press, 1998.

[15] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proc. ICML*, vol. 97, 1997, pp. 12–20.

[16] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," 2019, *arXiv:1912.01603*.

[17] D. Hafner et al., "Learning latent dynamics for planning from pixels," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 2555–2565.

[18] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[19] N. R. Ke et al., "Learning dynamics model in reinforcement learning by incorporating the long term future," 2019, *arXiv:1903.01599*.

[20] A. Xie, D. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning latent representations to influence multi-agent interaction," in *Proc. Conf. Robot Learn.*, PMLR, 2021, pp. 575–588.

[21] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[22] R. Valiente, M. Razzaghpour, B. Toghi, G. Shah, and Y. P. Fallah, "Prediction-aware and reinforcement learning based altruistic cooperative driving," 2022, *arXiv:2211.10585*.

[23] M. M. Botvinick and Z. B. Rosen, "Anticipation of cognitive demand during decision-making," *Psychol. Res. PRPF*, vol. 73, pp. 835–842, 2009.

[24] C. Engel, S. Kube, and M. Kurschilgen, "Managing expectations: How selective information affects cooperation and punishment in social dilemma games," *J. Econ. Behav. Org.*, vol. 187, pp. 111–136, 2021.

[25] T. Guo, M. Guo, Y. Zhang, and S. Liang, "The effect of aspiration on the evolution of cooperation in spatial multigame," *Physica A, Statist. Mech. Appl.*, vol. 525, pp. 27–32, 2019.

[26] Q. Wang and D. Jia, "Expectation driven by update willingness promotes cooperation in the spatial prisoner's dilemma game," *Appl. Math. Comput.*, vol. 352, pp. 174–179, 2019.

[27] M. Samvelyan et al., "The starcraft multi-agent challenge," 2019, *arXiv:1902.04043*.

[28] K. Kurach et al., "Google research football: A novel reinforcement learning environment," 2019, *arXiv:1907.11180*.

[29] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*. vol. 1, Cham, Switzerland: Springer-Verlag, 2016.

[30] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4295–4304.

[31] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, “QPLEX: Duplex dueling multi-agent q-learning,” 2020, *arXiv:2008.01062*.

[32] Y. Yang et al., “Qatten: A general framework for cooperative multiagent reinforcement learning,” , 2020, *arXiv:2002.03939*.

[33] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, “Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 10199–10210, 2020.

[34] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson, “Learning to communicate with deep multi-agent reinforcement learning,” 2016, *arXiv:1605.06676*.

[35] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, “Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5887–5896.

[36] P. Sunehag et al., “Value-decomposition networks for cooperative multi-agent learning,” 2017, *arXiv:1706.05296*.

[37] W. J. K. D. E. Hostallero, K. Son, D. Kim, and Y. Y. Qtran, “Learning to factorize with transformation for cooperative multi-agent reinforcement learning,” in *Proc. 31st Int. Conf. Mach. Learn.,* 2019, pp. 5887–5896.

[38] M. C. Thomas and A. T. Joy, *Elements of Information Theory*, vol. 3, New York, NY, USA: Wiley, 1991.

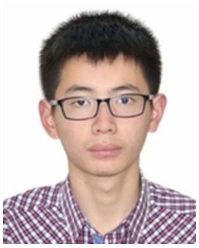
[39] T. Wang, H. Dong, V. Lesser, and C. Zhang, “ROMA: Multi-agent reinforcement learning with emergent roles,” 2020, *arXiv:2003.08039*.

[40] C. Li, C. Wu, T. Wang, J. Yang, Q. Zhao, and C. Zhang, “Celebrating diversity in shared multi-agent reinforcement learning,” 2021, *arXiv:2106.02195*.

[41] M. Samvelyan et al., “The StarCraft multi-agent challenge,” 2019, *arXiv:1902.04043*.

[42] J. Fernandez and L. Bornn, “Wide open spaces: A statistical technique for measuring space creation in professional soccer,” in *Sloan Sports Anal. Conf.*, vol. 2018, pp. 1–19, 2018.

[43] B. Liu, Z. Pu, T. Zhang, H. Wang, J. Yi, and J. Mi, “Learning to play football from sports domain perspective: A knowledge-embedded deep reinforcement learning framework,” *IEEE Trans. Games*, vol. 15, no. 4, pp. 648–657, Dec. 2023.



Boyin Liu received the B.Eng. degree in electrical engineering and its automation from Southeast University, Nanjing, China, in 2019. He is currently working toward the Ph.D. degree in computer application technology with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China.

His current research interests include multiagent reinforcement learning and graph neural networks.

Zhiqiang Pu (Member, IEEE) received the B.Eng. degree in automation from Wuhan University, Wuhan, China, in 2009, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently a Full Professor with the Institute of Automation, Chinese Academy of Sciences. His research interests include decision intelligence, collective intelligence, the intersection of sports science and artificial intelligence, and also applications of collective intelligence in unmanned systems.

Yi Pan received the Ph.D. degree in operations research and cybernetics from the University of Chinese Academy of Sciences, Beijing, China.

She is an Assistant Researcher with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her scientific research focuses on multiagent reinforcement learning and intelligent decision-making in soccer.

Jianqiang Yi (Senior Member, IEEE) received the B.Eng. degree in mechanical engineering from Beijing Institute of Technology, Beijing, China, in 1985, and the M.Eng. and Ph.D. degrees in automatic control from Kyushu Institute of Technology, Kitakyushu, Japan, in 1989 and 1992, respectively.

From 1992 to 1994, he worked as a Research Fellow with the Computer Software Development Company, Tokyo, Japan. From 1994 to 2001, he was a Chief Engineer with MYCOM, Inc., Kyoto, Japan. Since 2001, he has been a Full Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He has authored or co-authored over 100 international journal papers and 240 international conference papers. He holds more than 40 issued domestic patents. His research interests include theories and engineering applications of intelligent control, adaptive control, aerospace systems, and intelligent robotics.

Prof. Yi was an Associate Editor of IEEE *Computational Intelligence Magazine*, from 2008 to 2009, and is an Associate Editor for *Journal of Advanced Computational Intelligence and Intelligent Informatics* and *Journal of Innovative Computing, Information and Control*.

Min Chen received the B.Sc. degree in physics from the University of Chinese Academy of Sciences, Beijing, China, in 2020, and the M.Eng. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2023.

He is currently an Assistant Engineer with the Institute of Automation, Chinese Academy of Sciences. His current research interests include status assessment of soccer match and decision intelligence for collective systems.

Shijie Wang received the B.Eng. degree in control science and engineering from Shandong University, Jinan, China, in 2020. She is currently working toward the Ph.D. degree in control science and engineering with the Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Her current research interests include multiagent reinforcement learning, opponent modeling, and causal inference.