QFUTURE: LEARNING FUTURE EXPECTATIONS IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Building accurate and robust value functions to estimate the expected future return from the current state is critical in Multi-Agent Reinforcement Learning. Previous works perform better estimation by strengthening the representation of the value function. However, due to the uncertain and unavailable future, directly estimating the future return from the current state is challenging and cannot be addressed by just promoting representation ability. Socially, humans will derive future expectations from current available information to help evaluate their behavior's long-term return. Motivated by this, we propose a novel framework, called *future expectations multi-agent Q-learning* (QFuture), for better estimating future expected returns. In this framework, we design a future expectation module (FEM) to build future expectations in the calculation process of the individual (IAV) and joint action-value (JAV). In FEM, the future expectations are modeled as random variables and perform representation learning by maximizing their mutual information (MI) with the future trajectory given current observation (in IAV) or state (in JAV). We design a future representation module (FRM) to encode the future trajectory, where a regularizer is designed to ensure informativeness. Experiments on StarCraft II micromanagement tasks and Google Research Football demonstrate that QFuture significantly achieves state-of-the-art performance.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) has shown widespread application in many real-world systems, such as autonomous vehicle teams (Xu et al. (2018)), swarm systems (Hüttenrauch et al. (2017)), and traffic management (Singh et al. (2020)). However, the emergence of effective coordination among agents is still challenging. The shortsightedness of agents is a major obstacle to efficient cooperation among agents.

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 (Watkins & Dayan (1992)), SARSA (Rummery & Niranjan (1994)) and Actor-Critic (Konda & Tsitsiklis (1999)) methods) adopt Temporal Difference (TD) Learning (Sutton (1988)), where a n-step return, the 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. TD learning is further extended to deep reinforcement learning (DRL), and MARL (Sutton et al. (1998)). However, a significant issue has been ignored. Unlike the simple value iteration in the Q-table, TD learning in DRL expects the agent to estimate the n-step return using only the current information. Since the current reward correlates directly to the current state, this term can be an easy estimate. However, the future is uncertain and unpredictable. Deriving an estimate of future reward from current information is extremely challenging in DRL, especially in MARL, due to the increasing future possibilities caused by exponentially enlarged action space, which leads to inefficient learning in MARL.

A natural concept that comes to mind is using future information to improve the value function estimate. However, since future information is not available when agents make decisions, direct usage of future information is impossible. Socially, when humans are obliged to make a decision, they will derive future expectations based on the current observation by asking themselves a question: if we do that, what return would we obtain in the future? This procedure assists them in evaluating and improving decisions (Engel et al. (2021); Guo et al. (2019)). When the future arrives, those

future information will be used to improve their future expectations. Owing to future expectations, humans can make more reasonable assessments of their behaviors and then emerge with elaborate cooperation skills, such as planning and tacit understanding (Wang & Jia (2019)). Analogically, the emergence of future expectations should also be essential for MARL to alleviate the estimated difficulty of future reward in the value function.

In this paper, we propose a novel MARL approach, called *future expectations multi-agent Q-learning* (QFuture), to learn future expectations in MARL. Specifically, we design a future expectation module (FEM), where the future expectations are 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 expectations with actual future trajectories by maximizing a mutual information (MI) objective. In addition, we design a future representation module (FRM) and propose a regularizer to enable reasonable and effective representation of future trajectory. Finally, the output of FEM is used to generate the weight parameters to calculate the individual action-values (IAV) and joint action-value (JAV).

We conduct experiments on two environments, i.e., StarCraft II micromanagement environments (Samvelyan et al. (2019)) and Google Research Football (GRF) (Kurach et al. (2019)). The superior performance of our approach on challenging benchmarking tasks shows that our approach provides significantly higher coordination capacity. Moreover, we further carry out an ablation study to evaluate the contribution of each key component in QFuture. Finally, the visualization of learned strategy and future expectations in GRF effectively demonstrates future expectations can promote collaboration among agents.

2 BACKGROUND

2.1 PRELIMINARIES

We consider a fully cooperative multi-agent task as a *decentralised partially observable Markov decision process* (Dec-POMDP) (Oliehoek & Amato (2016)), which can be defined as a tuple $M =< N, S, A, P, r, Z, O, n, \gamma >$, 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 on the environment based on the transition function $P(s'|, s, \vec{a})$. Each agent has its own 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 joint action-value function: $Q^{\pi}_{tot}(s, \vec{a}) = \mathbb{E}_{s_{0:\infty}, a_{0:\infty}} [G_t | \, s_0 = s, \boldsymbol{a_0} = \vec{a}, \pi]$, where $G_t = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$ is the expected discounted return.

2.2 RELATED WORK

Centralized Training with Decentralized Execution (CTDE) has been a major paradigm of cooperative multi-agent deep reinforcement learning (Rashid et al. (2018); Wang et al. (2020a); Yang et al. (2020); Rashid et al. (2020)) and can effectively deal with nonstationarity while learning decentralized policies for agents (Foerster et al. (2016)). 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 (Rashid et al. (2018); Wang et al. (2020a); Yang et al. (2020)). It learns a decentralized utility function for each agent and then adopts a mixing network to combine local utilities into a global action value. IGM (*Individual-Global-MAX*;Son et al. (2019)) is an essential principle to realize effective value-based CTDE which asserts that $\exists Q_i$, such that the following holds:

$$\arg\max_{\vec{a}} Q^{\pi}_{tot}(s, \vec{a}) \quad = \quad (\arg\max_{a_1} Q_1(\tau_1, a_1), \ldots, \arg\max_{a_N} Q_N(\tau_N, a_N)). \tag{1}$$

Value-Function Factorization: Value-function factorization is the most popular method in value-based MARL under the CTDE paradigm. VDN (Sunehag et al. (2017)) proposes to decompose the value function of the team into agent-specific value functions by an additive factorization. QMIX (Rashid et al. (2018)) ameliorates the way of value-function factorization by learning a mixing network, following the Individual-Global-Max (IGM) principle (Hostallero et al. (2019)) . Qatten

(Yang et al. (2020)) is a variant of VDN, which supplements global information through a multi-head attention structure. QPLEX (Wang et al. (2020a)) employs a duplex dueling network architecture to estimate joint action-values, 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 expectations into the IAV and JAV calculation procedures in this paper.

MI Learning in MARL: 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, MAVEN (Mahajan et al. (2019)) extracts the latent variables about joint policy information from the initial global state and maximizes the MI of future trajectories and the latent variables. To learn roles, ROMA (Wang et al. (2020b)) proposes to optimize the conditional MI between the individual trajectory and the role given the current observation. To learn diversity, CDS (Li et al. (2021)) constructs an information-theoretical regularization to maximize the MI between agents' identities and their trajectories. MI has demonstrated its advantage in guiding the agent learning various capabilities. In this paper, with the help of MI, we construct future expectations learning in IAV and JAV.

3 QFUTURE LEARNING FRAMEWORK

In this section, we will introduce the QFuture learning framework. The overall architectural sketch of QFuture is illustrated in Figure 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 own IAV to take action with local observation. Specifically, in JAV, FEM derives the future expectations from the current global state s_t and then uses them to generate the parameters to calculate JAV. In IAV, FEM uses local observation as input and generates the parameters to calculate IAV.

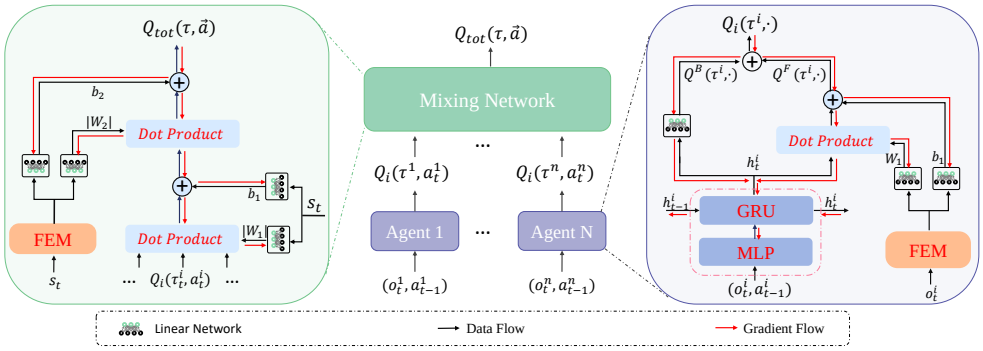


Figure 1: Schematics of QFuture. FEM is plugged into both the utility network and the mixing network to calculate IAV and JAV, respectively. The left is the mixing network, and the right is the utility network of each agent.

3.1 FUTURE EXPECTATION MODULE

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

$$\begin{aligned} (\mu_e, \sigma_e) &= f_e\left(x_t; \theta_e\right) \\ e &= \mu_e + \sigma_e \odot \varepsilon_e, \quad \varepsilon_e \sim N(0,1) \end{aligned} \tag{2}$$

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

Future Expectations Learning: We expect FEM to construct future expectations 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 $I(e_t; \tau_t^f \mid x_t)$ between future expectations e_t and future trajectory τ_t^f given x_t, where $\tau_t^f = (x_T, x_{T-1}, ..., x_{t+1})$ is the future trajectory information at time t.

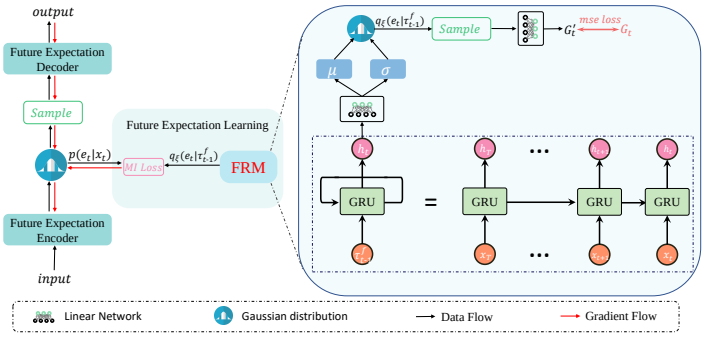


Figure 2: Schematics of FEM. T is a final time step.

However, since the estimate of MI is intractable, we can not directly optimize this objective. Motivated by the related works about variational inference (Alemi et al. (2016),Mahajan et al. (2019)), a variational estimator is introduced to derive a tractable lower bound for the MI objective (see the proof in Appendix A.1):

$$\begin{aligned} I(e_t; \tau_t^f \mid x_t) &= \mathbb{E}_{e_t, \tau_t^f}[\log \frac{p(e_t, \tau_t^f \mid x_t)}{p(e_t \mid x_t) p(\tau_t^f \mid x_t)}] \\ &\geq -\mathbb{E}_{\tau_t^f}[\mathcal{CE}[p(e_t \mid x_t) \| q_\xi(e_t \mid \tau_{t-1}^f)]] + \mathbb{E}_{\tau_t^f}[H(e_t \mid x_t)], \end{aligned} \tag{3}$$

where T is a final time step; q_ξ is the variational estimator parameterised with ξ and $H(\cdot)$ denotes the entropy. For q_ξ, we design a future representation module (FRM) to encode the future trajectory information. As shown in Figure 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. Meanwhile, the learned future representation should be informative. To this end, an auxiliary loss function is necessary. In MARL, we focus on the expected return. Therefore, we design a return-based loss to guide the learning process. As shown in Fig. 2, the future representation embedding 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 FRM is

$$\mathcal{L}_{RB}(\xi) = \mathbb{E}_{<G_t, G_t^{'}> \sim \mathcal{B}}[(G_t^{'} - G_t)^2]. \tag{4}$$

Then, the lower bound in Eq. 3 can be formalized as a loss function to be minimized:

$$L_{MI}(\theta_e, \xi) = \mathbb{E}_{\tau_t^f \sim \mathcal{B}}[D_{KL}[p(e_t \mid x_t) \| q_\xi(e_t \mid \tau_{t-1}^f)]], \tag{5}$$

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

3.2 Future Expectations in IAV

Socially, future expectations improve human behaviors by influencing their evaluation of strategies. As a result, we additionally propose a future expectation Q-function Q^F. In teamwork, different

members will show different levels of concentration on future expectations, e.g., leaders will share more but followers less. Therefore, we let agents adaptively decide the focus level by decomposing Q_i as

$$Q_i(a_i \mid \tau_i) = Q^B(a_i \mid \tau_i) + Q^F(a_i \mid \tau_i), \tag{6}$$

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

Here, FEM uses its observation information with $x_t^i = (o_t^i)$ as input, and then future expectations encoder generates an embedding distribution. We then sample a future expectations vector e_t and input it to the decoder, which generates the parameters of the local utility network. In centralized training phase, $\tau_{t-1}^{f_i} = (x_T^i, x_{T-1}^i, ..., x_t^i)$ is fed into a GRU in reverse chronological order. After several steps computation, FRM 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 Eq. 5, 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 Appendix A.2), pursuing this objective also maximizes the MI $I(e_t^i; \tau_t^{f_i} \mid x_t^i)$ that can be formulated as

$$I(a_t^i; \tau_t^{f_i} \mid x_t^i) = H(a_t^i \mid x_t^i) - H(a_t^i \mid \tau_{t-1}^{f_i}). \tag{7}$$

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

$$I(a_t^i; \tau_t^{f_i} \mid x_t^i) = H(a_t^i \mid x_t^i). \tag{8}$$

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

3.3 Future Expectations in JAV

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

$$\mathcal{L}_{TD}(\theta) = \sum_{i=1}^{b}[(r + \gamma \max_{\vec{a}'} Q_{tot}(s', \vec{a}'; \theta^-) - Q_{tot}(s, \vec{a}; \theta))^2], \tag{9}$$

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

Many value decomposition methods try to complicate the mixing network to strengthen the representation ability, such as QPLEX (Wang et al. (2020a)) and Qatten (Yang et al. (2020)). The ability of JAV to build better estimates of the joint action-value 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 $\gamma \max_{\vec{a}'} Q_{tot}(s', \vec{a}'; \theta^-)$. 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 expectations into the mixing network to provide faster and more reliable learning.

In JAV, FEM takes the global state $x_t = s_t$ as input and then generates future expectations e_t, which are input into the decoder network and finally generates the final parameters to calculate $Q_{tot}(\tau, \vec{a})$. Here is also a MI regularizer L_{JMI} according to Eq. 5 to learn future expectations.

3.4 Overall Optimization Objective

We have introduced optimization objectives for future expectations learning in IAV and JAV. The final learning objective of QFuture is:

$$\mathcal{L}(\theta) = \mathcal{L}_{TD}(\theta) + \beta_I \mathcal{L}_{IMI}(\theta_e^I, \xi^I) + \beta_J \mathcal{L}_{JMI}(\theta_e^J, \xi^J) + \mathcal{L}_{RB}(\xi) \tag{10}$$

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 expectations encoder in IAV and JAV respectively; $\xi = (\xi^I, \xi^J)$ are the parameters of FRM; $\mathcal{L}_{IMI}$ and $\mathcal{L}_{JMI}$ represent the MI regularizers in IAV and JAV respectively; ξ^I and ξ^J represent the parameters of FRM in IAV and JAV respectively; β^I and β^J are scaling factors.

4 EXPERIMENTAL RESULTS

4.1 EXPERIMENTAL SETUP

To evaluate the effectiveness of QFuture, we conduct experiments with different scenarios on two challenging benchmarks, i.e., StarCraft II micro-management challenge. (SMAC) (Samvelyan et al. (2019)) [1] and Google Research Football (GRF) (Kurach et al. (2019)). In these tasks, QFuture is compared with Qtran (Son et al. (2019)), QMIX (Rashid et al. (2018)), QPLEX (Wang et al. (2020a)) and Qatten (Yang et al. (2020)), all of which can be implemented on both Starcraft II 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.

The details of the architecture of our method, baselines, and task settings can be found in Appendix B. Additional experimental results can be found in Appendix C.

4.2 COMPARISON STUDIES

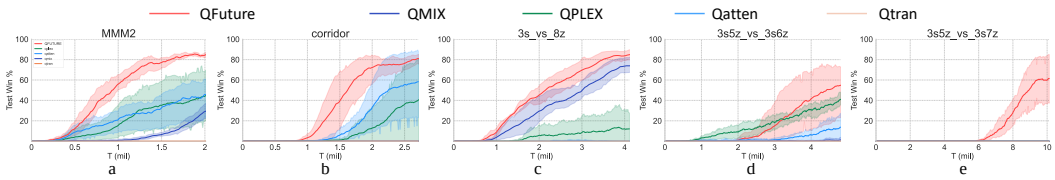


Figure 3: Comparison of our method against baseline methods on five super hard maps in StarCraft II with the evaluation index of test winning rate.

In this section, we first carry out the experiments on StarCraft II with four random seeds, and the average results are shown in Fig. 3. 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 hand-crafted map *3s5z_vs_3s7z*, only QFuture can explore an effective strategy. Overall, the comparison studies on StarCraft II show the success of QFuture in learning performance improvements.

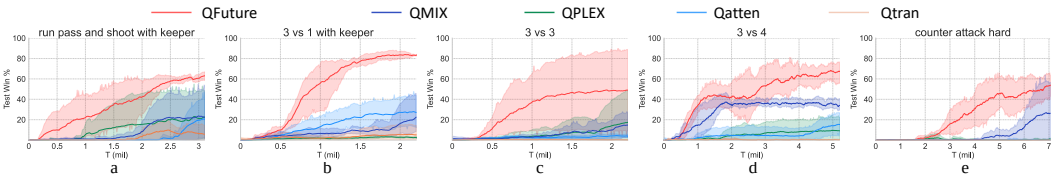


Figure 4: Comparison of our method with baseline methods on five academy tasks in GRF with the evaluation index of test winning rate.

To further evaluate the proposed method in cooperative tasks, we conduct experiments on five GRF tasks. Unlike StarCraft II 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. 4, QFuture achieves superior performance on all tasks, whose curves rise earliest and fastest. Among these

[1] We use SC2.4.10 version.

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 StarCraft II 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.

4.3 Ablations

To thoroughly understand the superior performance of QFuture, we carry out ablation studies to show the contribution of future expectations learning. In particular, we carry out the following ablation studies: (1) $w/o\ L_{IMI}$: the QFuture without (abbreviated as w/o) MI regularizer in IAV; (2) $w/o\ L_{JMI}$: the QFuture without MI regularizer in JAV; (3) $w/o\ L_{RB}$: the QFuture without return-based regularizer in both IAV and JAV's FRM.

As shown in Fig.5, QFuture offers the best learning performance on nearly all tasks. In addition, the deletion of any regularizer can still lead to evidently 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.

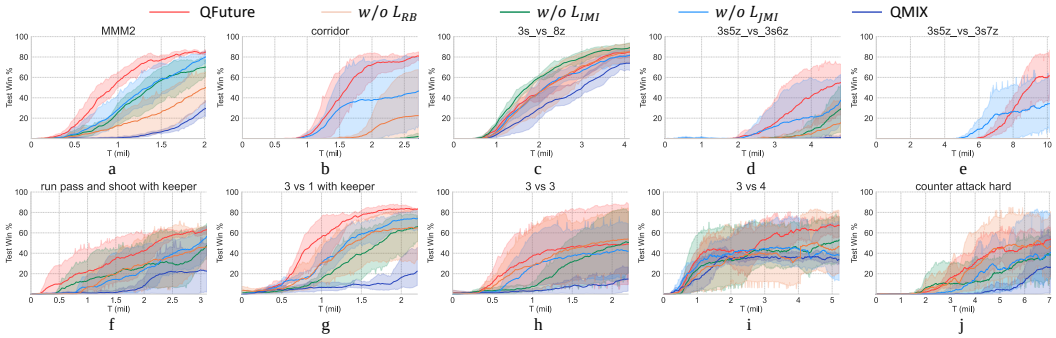


Figure 5: The Winning rate of ablation studies on all tasks.

In Starcraft II tasks, removing return-based loss L_{RB} in FRM will bring the most noticeable performance decadence. This detachment will induce adverse effects for FRM, which may fail to extract helpful information from the future trajectory, and then influence the future expectations of learning in both IAV and JAV. However, in GRF tasks, $w/o\ L_{RB}$ only shows slight performance degeneration compared to those in Starcraft II tasks. Since GRF is a sparse reward scenario, FRM cannot receive effective training due to the lack of feedback, especially at the early training stage, which hinders the learning of future expectations. However, Starcraft II tasks provide dense rewards for each frame, which then induce more effective future 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 FRM. In addition, we show how to improve the performance of QFuture in sparse reward tasks in Appendix C.1.

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 3.2, 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 weaken the exploration ability. As shown in Figs. 5(f $\sim$ j), compared to $w/o\ L_{RB}$ and $w/o\ L_{JMI}$, $w/o\ L_{IMI}$ shows the worst learning quality, particularly in the early stages, where $w/o\ L_{IMI}$ usually spends more training steps to explore a strategy with 20% win rate. Furthermore, as shown in Figs. 5(b,d,e), $w/o\ L_{IMI}$ 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 $w/o\ L_{IMI}$ reveals the ability to explore the environment.

The learning curves of $w/o\ L_{JMI}$ show outstanding performance in most tasks. The comparison of QFuture performance and $w/o\ L_{JMI}$ proves that the regularizer L_{JMI} effectively reduces performance variance between different random seeds, particularly in Starcraft II tasks. Furthermore, as shown in Figs. 5(b,d,e), QFuture offers faster promotion of the strategy among agents after explor-

ing an effective strategy. This may be because the mixing network with future expectations provides more reasonable credit assignments.

4.4 VISUALIZATION OF FUTURE EXPECTATIONS

In this section, we visualize the learned strategy and its future expectations, shown in Fig. 6. 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 Figs. 6(a, 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 (Fernandez & Bornn (2018)) at the final step, with 1 for entirely left team dominance and 0 for the entire right team, whereas the length of the vector represents the player's velocity at the final time step. To show the learned strategy delicately, we provide more pitch control snapshots in the Appendix C.3 and provide the learned strategy's video in supplement materials. In Figs. 6(b,c,e,f), dots with the same color denote the same phase in an episode. The number close to the dot is the time step located in the strategy.

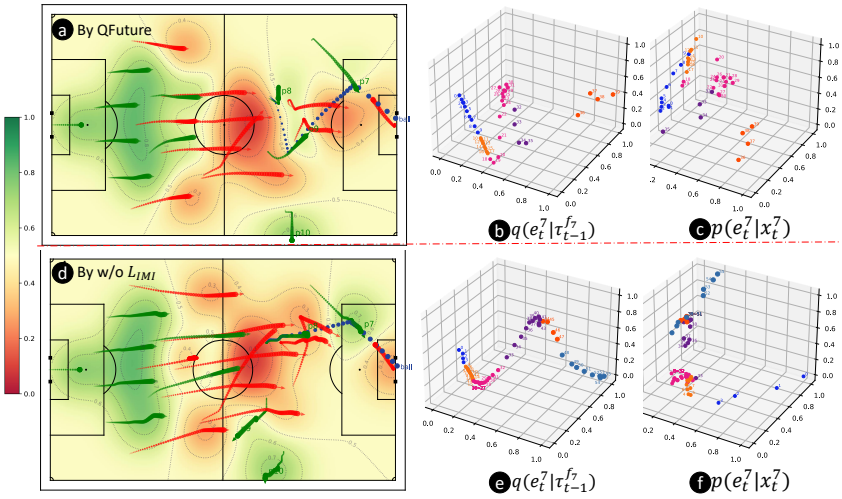


Figure 6: The visualization of the learned strategies, the future representation distributions $q(e_t^7|\tau_{t-1}^{f_7})$ and future expectations $p(e_t^7|x_t^7)$. The number 7 represents the player $p7$. Fig.a and Fig.d are strategies learned by QFuture and $w/o\ L_{IMI}$, respectively. Figs.(bc) and Figs.(ef) are principal components of sampling vector at each step after Linear PCA, corresponding to the strategy in Fig. a and Fig.d, respectively.

Strategy learned by QFuture: As shown in Fig. 6 (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.

In Fig. 6 (b), we visualize the learned future representation distribution $q(e_t^7|\tau_{t-1}^{f})$, who is the leading role of this score. Overall, three laws can be concluded from this figure. First, the positions of dots in the same event phase are nearly in line with time. Second, the dots in different event phases will not overlap. Third, 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 at the second transition, and completes a shot at the third transition. These transitions accord with our understanding of football that a progressive pass could

change the situation on the field. Socially, future expectations should change as time progresses and show a turning point when critical events happen. These two laws correspond to our cognition of future expectations. Therefore, by comparing the learned strategy with Fig. 6(b), it can be concluded that the FRM successfully encodes future trajectories for agents.

Strategy learned by $w/o\ L_{IMI}$: As illustrated in Fig. 6(d), this strategy can be decomposed into 6 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 the ball and waiting for $p7$ running to appropriate location. (4) From steps 33 to 43, $p7$ stand still and $p8$ give 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 step 48 to 55, $p7$ shots and then gets a score, spend 8 step from shot to goal.

In Fig. 6(e), we also visualize the learned future representation distribution ($q(e_t^7|\tau_t^{f_7})$). Clearly, the orbit of these dots continues to follow the aforementioned three laws. Particlularly, there two transitions, i.e. (16,17,18) and (43,44,45). These results further demonstrate the effectiveness of our learned future representation.

How MI regularizer help build future expectations: As shown in Fig. 6(c), although the learned future expectations 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. Specifically, e_t also captures three transitions in this strategy. As shown in Fig. 6(f), the majority 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. Furthermore, two transitions in Fig. 6(e) are difficult to capture here. Hence, the future expectations fail to be build without L_{IMI} regularizer. The difference between Fig. 6 (e) and (f) clearly describes the necessity and effectiveness of the proposed MI regularizer to establish future expectations.

How future expectations benefits cooperation: By comparing two strategies, QFuture learns a more excellent strategy. It is noted that the strategy learned by QFuture consumes same training steps with $w/o\ L_{IMI}$. With future expectations, 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 $w/o\ L_{IMI}$, $p8$ 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 expectations, i.e., running to receive a pass and one-touch pass (shot). With future expectations, 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. 6 (d) fails to perform a one-touch pass or shot since future expectations are hard to construct without a specific guide. In general, future expectations help players achieve closer cooperation with tacit understanding and contribute to a quick attack strategy in Fig. 6(a).

5 Conclusion and Future Work

In this paper, we have introduced the concept of future expectations into deep multi-agent reinforcement learning by maximizing a MI objective $I(e_t, \tau_t^f \mid x_t)$. Future expectations are used to generate parameters to estimate individual action-values and joint action-value. Experimental results demonstrate the superior effectiveness of our method.

To our best knowledge, it is the first paper using all future step's information at each step's training. 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 sing-agent RL and multi-agent RL fields.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Christoph Engel, Sebastian Kube, and Michael Kurschilgen. Managing expectations: How selective information affects cooperation and punishment in social dilemma games. *Journal of Economic Behavior & Organization*, 187:111–136, 2021.

Javier Fernandez and Luke Bornn. Wide open spaces: A statistical technique for measuring space creation in professional soccer. In *Sloan sports analytics conference*, volume 2018, 2018.

Jakob N Foerster, Yannis M Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

Tian Guo, Mi Guo, Yan Zhang, and Shuanglu Liang. The effect of aspiration on the evolution of cooperation in spatial multigame. *Physica A: Statistical Mechanics and its Applications*, 525: 27–32, 2019.

Wan Ju Kang David Earl Hostallero, Kyunghwan Son, Daewoo Kim, and Yung Yi Qtran. Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 31st International Conference on Machine Learning, Proceedings of Machine Learning Research. PMLR*, 2019.

Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011*, 2017.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. *arXiv preprint arXiv:1907.11180*, 2019.

Chenghao Li, Chengjie Wu, Tonghan Wang, Jun Yang, Qianchuan Zhao, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *arXiv preprint arXiv:2106.02195*, 2021.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020.

Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. Citeseer, 1994.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Arambam James Singh, Akshat Kumar, and Hoong Chuin Lau. Hierarchical multiagent reinforcement learning for maritime traffic management. 2020.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020a.

Qiuling Wang and Danyang Jia. Expectation driven by update willingness promotes cooperation in the spatial prisoner's dilemma game. *Applied Mathematics and Computation*, 352:174–179, 2019.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020b.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Zhao Xu, Yang Lyu, Quan Pan, Jinwen Hu, Chunhui Zhao, and Shuai Liu. Multi-vehicle flocking control with deep deterministic policy gradient method. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pp. 306–311. IEEE, 2018.

Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020.

A Mathematical Derivation

A.1 Future Expectations Learning

In this paper, we propose to learn future expectations which is implemented by maximizing:

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

Since future expectations encoder is conditioned on current information x_t, the distributions of future expectations $p(e_t)$ is independent from the future trajectory τ_t^f. Therefore, we have

$$
I(e_t; \tau_t^f \mid x_t) \geq -\mathbb{E}_{\tau_t^f, x_t}\left[\mathcal{CE}\left[p(e_t \mid x_t) \| q_\xi(e_t \mid \tau_{t-1}^f)\right]\right] + \mathbb{E}_{x_t}\left[H(e_t \mid x_t)\right], \tag{11}
$$

We use a replay buffer $\mathcal{B}$ in practice. We can derive the following minimization objective:

$$
L_{MI}(\theta_e, \xi) = \mathbb{E}_{\tau_t^f \sim \mathcal{B}}\left[D_{KL}\left[p\left(e_t \mid x_t\right) \| q_\xi(e_t \mid \tau_{t-1}^f)\right]\right] \tag{12}
$$

Above proofs refer to the derivation in MAVEN (Mahajan et al. (2019)) and ROMA (Wang et al. (2020b)).

A.2 $I(a_t^i; \tau_t^{f_i} \mid x_t^i)$ 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:

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

$$= I(\tau_t^{f_i}; a_t^i \mid x_t^i) + I(\tau_t^{f_i}; e_t^i \mid a_t^i, x_t^i). \tag{14}$$

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

$$I(\tau_t^{f_i}; a_t^i \mid x_t^i) \geq I(\tau_t^{f_i}; e_t^i \mid x_t^i). \tag{15}$$

Thus, the proof is accomplished.

B EXPERIMENT DETAILS

B.1 QFUTURE AND BASELINES

In this paper, we compare our approach with five value-based methods. QFuture is developed based on the QMIX. For QMIX, we use the code framework in `https://github.com/starry-sky6688/MARL-Algorithms`. 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 provided by PYMARL2 (`https://github.com/hijkzzz/pymarl2`), and we use the default training settings in StarCraft II 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, correlated to the MI regularizers. For Starcraft II 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.

Experiments are carried out on NVIDIA GTX3090 GPU.

B.2 STARCRAFT II

StarCraft II is a popular real-time strategy game, which 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 1.

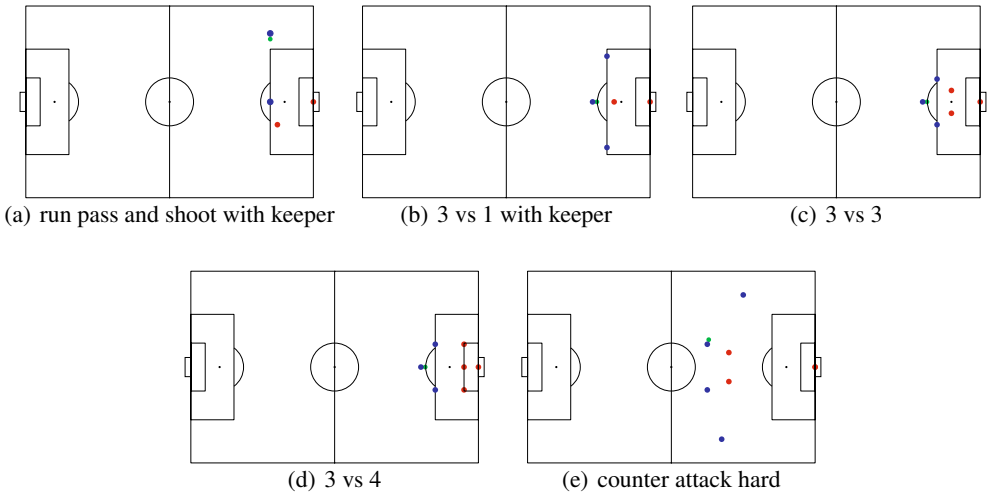
Table 1: Starcraft II challenges.

Task	Ally Units	Enemy Units	Type	Challenge
Corridor	6 Zealots	24 Zerglings	Asymmetric, Homogeneous	Kite enemy
3s_vs_8z	3 Stalkers	8 Zealots	Asymmetric, Homogeneous	Kite enemy
3s5z_vs_3s6z	3 Stalkers, 5 Zealots	3 Stalkers , 6 Zealots	Asymmetric, Heterogeneous	Medivac absorbs fire
3s5z_vs_3s7z	3 Stalkers, 5 Zealots	3 Stalkers , 7 Zealots	Asymmetric, Heterogeneous	Medivac absorbs fire
MMM2	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 2 Marauders , 8 Marines	Asymmetric, Heterogeneous	Circuitous tactics

B.3 GRF TASKS

In GRF, agents are trained to play football in a physics-based 3D 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 (3 official and 2 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. 7. 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 are only two types of rewards: (1) a reward $+0.5$ for the first time the team gets the ball. (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 left half-court, (3) scoring a goal (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 range $[-1, 1]$.



(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

Figure 7: Visualization of the initial position of each agent in five GRF tasks. Blue dots represent the agent. Red dots are opponents, and the green dot denotes the ball.

C MORE EXPERIMENTAL RESULTS

C.1 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 learning. To address this problem, we slightly modify QFuture for sparse reward tasks.

In sparse reward tasks, we can change the predict target in the FRM of the agent network from G_t (denoted QFuture-G_t) to $\max_{\vec{a}'} Q_{tot}(s', \vec{a}'; \theta^-)$ (denoted QFuture-Q'_{tot}). Once agents can succeed with a specified probability, $\max_{\vec{a}'} Q_{tot}(s', \vec{a}'; \theta^-)$ can evaluate these failed episodes with the help of successful episode experience. As shown in Fig. 8, QFuture-Q'_{tot} show evident performance promotion in GRF tasks. In Fig. 8(a,c,e), QFuture-Q'_{tot} performs an increase of over ten percent winning rate at the end. In Fig. 8(b,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.

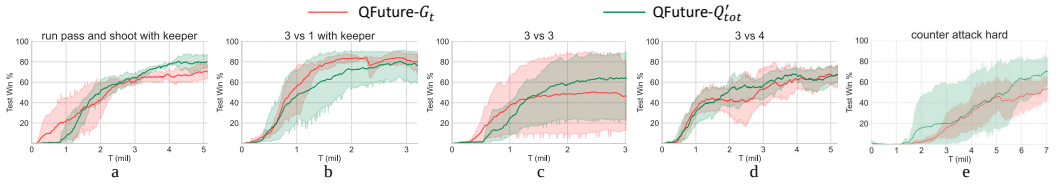


Figure 8: Comparison of QFuture-G_t against QFuture-Q'_{tot} on five academy tasks in GRF with the evaluation index of test winning rate.

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

C.2 Applying Parts of QFuture to QMIX

In this paper, we propose a novel method to introduce future expectations 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. 9, QMIX-FIAV shows significant improvements to QMIX, demonstrating the scalability of our method. FJAV also show better performance than QMIX. The splendid performance on QFuture indicates that the combination of FIAV and FJAV will present an advantageous performance than used them separately.

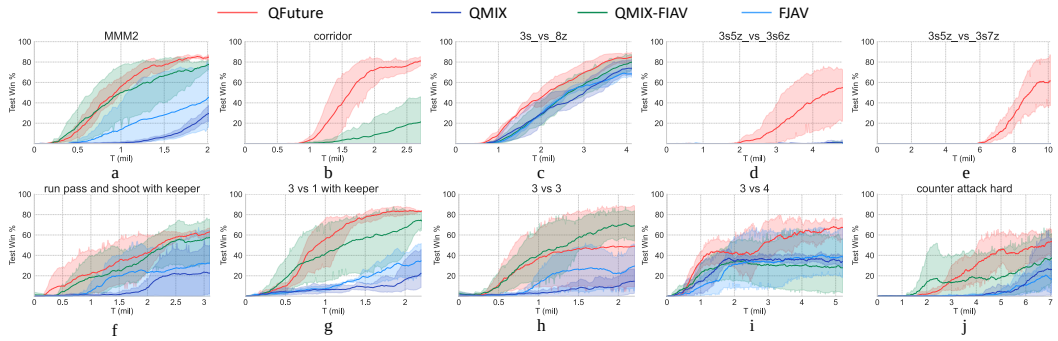


Figure 9: The Winning rate of QFuture, QMIX, QMIX-FIAV and FJAV on all tasks. .

C.3 Learned Strategy in GRF

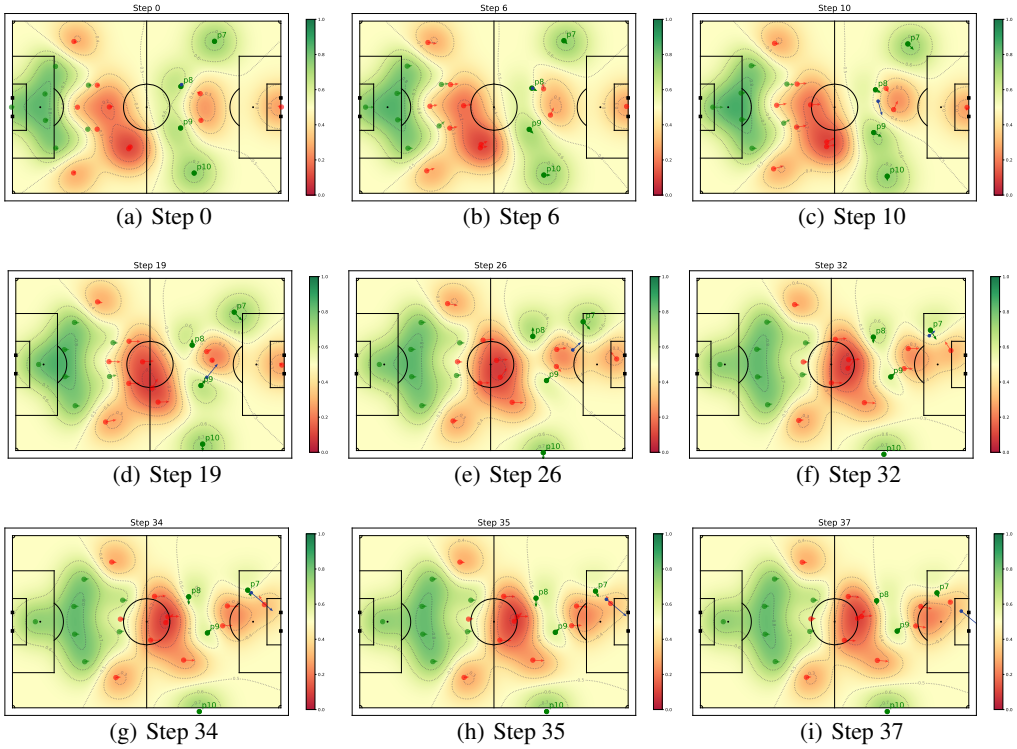
Football is a game about space. Here, we use the pitch control model (Fernandez & Bornn (2018)) to visualize the learned strategy.

QFuture: Comparing Fig. 10(a) and Fig. 10(c), $p8$'s dribble in Fig. 10(b) creates more valuable space for $p9$. As shown in Fig. 10(d), $p9$ shares a broad safe space to pass the ball. Comparing Fig. 10(d) and Fig. 10(f), the land point of the oblique pass is in the penalty box, where the receiver $p7$ does not reach this position when the pass starts. With the help of future expectations, the passer $p9$ knows $p7$ will reach this position at the correct time, and the receiver $p7$ knows how to plan his velocity to receive the ball. After the ball is passed to $p7$, $p8$ and $p9$ still occupy valuable upfield space for the team.

w/oL_{IMI}**:** As shown in Figs. 11(a,b,c), $p8$ gets the ball and stands still waiting for $p7$ running to appropriate positions. However, at this interval, more right team defenders cross the half line and occupy more valuable space in their half. As shown in Fig. 11(d), $p8$ is surrounded by defenders.

When $p7$ reaches the correct position and stands still, the ball is passed from $p8$ to $p7$. When $p7$ receives the ball, the majority of valuable space upfield has been occupied by the defenders.

Comparison: With future expectations, QFuture evidently learn higher level skills, where cooperation in $p7$, $p8$ and $p9$ show tacit understandings. In the whole episode, the strategy learned by QFuture occupies more valuable space in the opposition half than w/oL_{IMI}. In QFuture, agents learn one-touch pass and shot, which can pass the ball quickly and be powerful. Comparing Fig. 10(g) and Fig. 10(h), the ball shot by the agent $p7$ performs at higher speeds (the length of the vector represents the velocity). Overall, the agent with future expectations learns more effective cooperation.

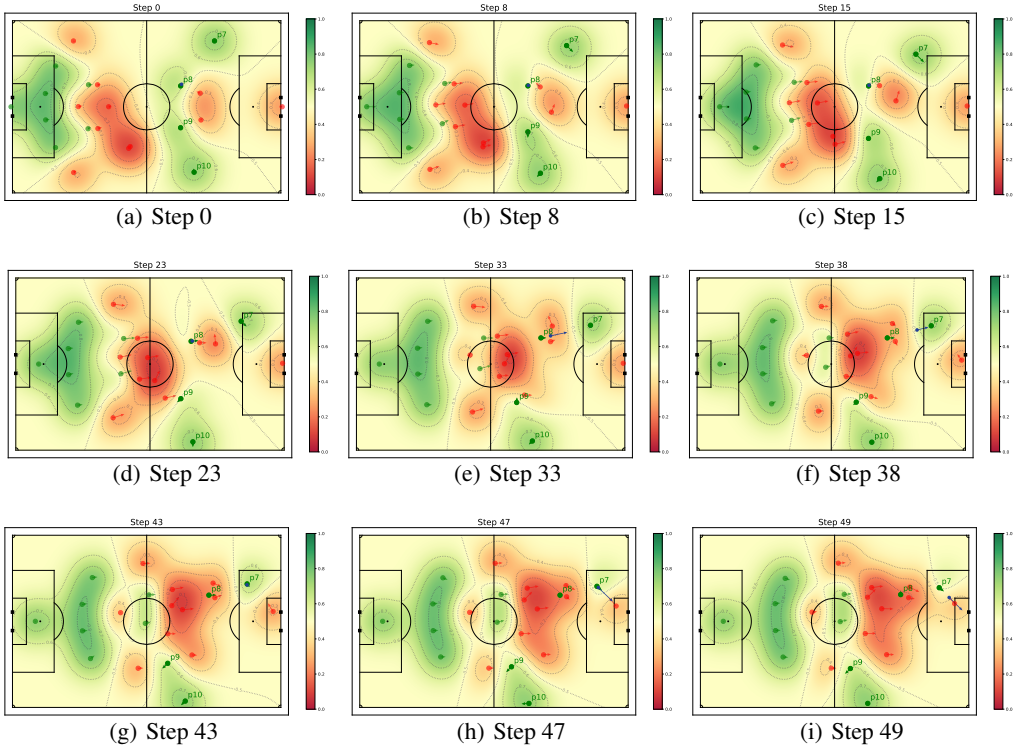


(a) Step 0 (b) Step 6 (c) Step 10

(d) Step 19 (e) Step 26 (f) Step 32

(g) Step 34 (h) Step 35 (i) Step 37

Figure 10: Snapshots of the learned strategy by QFuture.



(a) Step 0 (b) Step 8 (c) Step 15

(d) Step 23 (e) Step 33 (f) Step 38

(g) Step 43 (h) Step 47 (i) Step 49

Figure 11: Snapshots of the learned strategy by $w/o\ L_{IMI}$.