Learning Generalizable Multi-Agent Cooperation Policy under Risk Conditions

Journal:	*IEEE Transactions on Neural Networks and Learning Systems*
Manuscript ID	TNNLS-2025-P-43634
Manuscript Type:	Regular Paper
Keywords:	Multi task generalization, Multi-agent reinforcement learning, Risk knowledge, Risk conditions

Learning Generalizable Multi-Agent Cooperation Policy under Risk Conditions

Anonymous

***Abstract*—Multi-agent reinforcement learning (MARL) often faces the challenge of agent number difference between the target and source tasks in multi-task generalization. Current multi-task MARL methods attempt to learn cooperation skills as the basis for generalization, but these skills are quite sensitive to offline data. This paper finds that agents adopting similar risk responses when facing potential risks in different tasks contributes to enhancing the generalization of their cooperation policy. Specifically, this paper proposes an environmental risk knowledge (RIKD) representation and response method to achieve generalizable multi-agent cooperation policy learning in risk conditions. RIKD first trains a risk attitude selector based on offline data to learn the mapping from risk knowledge to risk attitudes. Then, RIKD coordinates the actions of agents with different risk attitudes through a mixing network, and learns a risk knowledge extraction network for agents that can extract risk knowledge from the current observational. Additionally, to enhance RIKD's ability to adapt to environmental risk changes in new tasks, we have provided its online version, RIKD (online). Empirical results on MARL benchmarks demonstrate that RIKD significantly improves the multitask generalization performance of cooperative policies under risk conditions.**

***Index Terms*—Multi task generalization, Multi-agent reinforcement learning, Risk knowledge, Risk conditions.**

I. Introduction

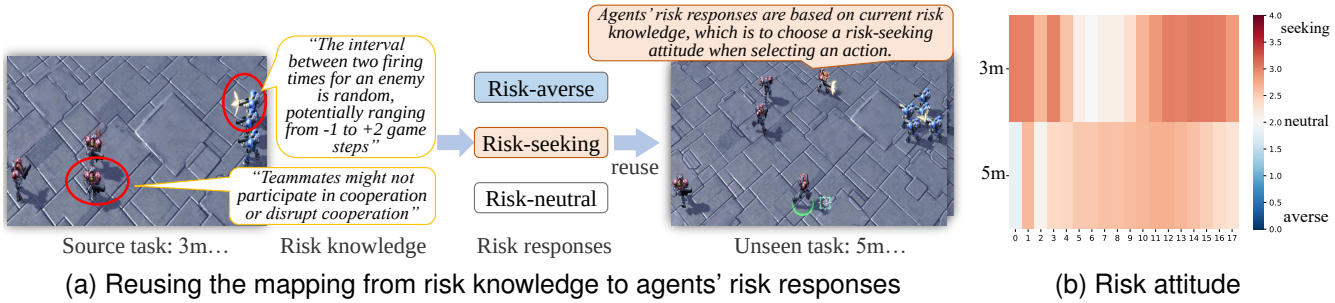
MULTI-agent reinforcement learning (MARL) can be used for learning cooperation policy among multiple agents and has shown great potential in applications such as real-time strategy games, voltage control, and traffic network control [1]–[6]. However, the cooperation policy learned in a specific task is difficult to adapt to another new task with a different number of agents [7]–[9].

To improve the generalization ability of multi-agent cooperation policies, previous work has primarily focused on designing various multi-task MARL methods [10]–[13]. In particular, learning general knowledge from offline data has become a trend [13], in which learning cooperation skills is the most representative [14]. However, cooperation skill learning approaches require manually specifying the number of skills and are sensitive to the offline data [15], [16]. Furthermore, these methods struggle with risk conditions that the agents may face. While risk-sensitive MARL methods consider environmental risks, they are tailored for specific tasks and are not adaptable to multi-task settings [17], [18]. Therefore, exploring general knowledge that insensitive to offline data is essential for enhancing agent policies' adaptability under risk conditions.

This paper finds that agents adopting similar risk responses when facing potential risks in different tasks contributes to enhancing the generality of cooperation policy. The potential risks arise from the inherent uncertainty of the environment and the uncertainty of other agents' policy. For instance, in StarCraft Multi-Agent Challeng (SMAC) tasks [19], the inherent uncertainty of the environment includes stochastic weapon firing intervals, while the uncertainty of other agents' policy primarily refers to the possibility that teammates might disrupt cooperation, as shown in Figure 1(a). Despite the variation in environmental risks across different tasks, the risk responses made by agents (i.e., the risk attitudes they adopt) may exhibit similarities. In Figure 1(b), we demonstrate our point on different SMAC tasks. In the `marine` environment, we used the trained RIKD model to collect the risk attitudes of agents under two different tasks of `3m` and `5m`. Although the tasks differed, agents used similar risk attitude selection policies, i.e., risk-seeking, in the initial 18 steps of the tasks.

Inspired by the consistent understanding of environmental risks by agents in similar and different tasks, we focus on learning the mapping from risk knowledge to risk response for generalization. The aim is to determine which risk attitude an agent should adopt when confronted with a certain level of environmental risk. Risk knowledge here refers to the awareness that agents have about potential risks in their environment, while risk response pertains to how the agent chooses risk attitude to react to these risks, whether by being risk-averse, risk-neutral, or risk-seeking. Our aim is to develop a general policy for agents running in different tasks, in which agents can adjust their action appropriately based on the risks they perceive, ensuring that their reactions are beneficial within the context of the multi-agent system they are integrated into.

Therefore, we propose a multi-agent multi-task risk knowledge representation and risk response framework based on distributed RL (RIKD), which learns the mapping from risk knowledge to risk response from offline data. RIKD includes three parts: risk knowledge representation, risk attitude selection, and cooperation policy learning. **(1) Risk knowledge representation**. To adapt to the different number of agents in different tasks, we employ a risk knowledge extraction network to represent environmental risk knowledge as latent variables. **(2) Risk attitude selection**. We design a risk attitude selector that uses an *option* framework [20] to generates targeted risk attitudes for each agent based on their risk knowledge, to assist agents in making risk-sensitive individual decisions. **(3) Cooperation policy learning**. RIKD implicitly incorporates environmental risks into the learning of cooperation policy by using policy value distribution [21], [22], and coordinates the policy of agents with different risk attitudes through centralized training. Next, RIKD applies the policy model to other unseen tasks during the deployment phase to enable agents to select appropriate risk attitudes based on current risk knowledge. Due to the characteristics of the offline



(a) Reusing the mapping from risk knowledge to agents' risk responses (b) Risk attitude

Fig. 1. (a) Agents learn a mapping from risk knowledge to risk response using offline data, and then, in an unseen task, select risk attitudes based on the current risk. (b) The heatmap of similar mean values of risk attitudes selected by all agents with RIKD model in the initial 18 steps of the `marine` (`3m`, `5m`) tasks.

algorithm, RIKD is difficult to adapt to environmental risks change in realistic tasks, so we further provide a version of RIKD with online fine-tuning ability, RIKD (online).

The main innovations are as follows:

- Firstly, this paper innovatively proposes using the mapping of risk knowledge to risk responses, which is more general and foundational than skills, as the basis for multi-task generalization, and it demonstrates sufficient robustness to offline data.
- Secondly, to our knowledge, this paper is the first attempt to learn about the potential environmental risks in multi-agent multi-task scenarios using distributed RL, and it investigates varied risk response methods of agents in conjunction with an options framework.
- Finally, we have demonstrated through extensive experiments that RIKD and RIKD (online) exhibits superior performance in multiple challenging tasks compared to the SOTA baseline.

The rest is organized as follows. Section II reviewed and summarized the related work. Section III provides a problem description and some preliminary content. Section IV details the process and steps of RIKD algorithm. Then, Section V validates the algorithm's policy generalization performance on multiple tasks through a large number of simulation experiments. Finally, Section VI summarizes the paper.

II. Related work

Knowledge is not just a simple policy, but can also be an abstraction of states, high-level policy, etc [23]–[25]. The curriculum learning method is also based on knowledge transfer, advocating for policy transfer between task sequences with different difficulty levels [26]–[29]. Some studies define cooperation knowledge between multi-agent as skills and suggest that cooperation skills between multi-agent are universal across different tasks [12], [30], [31]. However, although skills are crucial in multi-agent collaboration, they do not seem to be the most fundamental knowledge, as skills are closely related to the offline data involved. Therefore, to learn more general knowledge, this paper proposes using risk knowledge in multi-task as the transfer subject, and based on risk knowledge, learns the risk attitudes that agents can share between different tasks, thereby achieving the generalization of cooperation policy with risk-sensitive characteristics.

Regarding risk-sensitive policy, the distributional RL learns the parameterized representation of the policy value distribution through different forms, such as quantiles, while reinforcement learning methods based on fuzzy theory are used to address risks brought by uncontrollable environments [32]–[34], achieving state-of-the-art performance in control tasks [35], [36]. Recent works have attempted to apply distributional RL to offline learning to address the risk issues caused by inconsistent data distribution [37]. However, these methods are limited to application in single-agent scenarios and are powerless in the face of multi-task generalization in multi-agent policy. To our knowledge, there are currently no reports on using distributional RL methods in the multi-agent multi-task policy generalization process. Given the natural advantage of distributional RL in characterizing risk, we use risk knowledge as the basis for multi-agent policy generalization, and combine distributional RL and option frameworks in the algorithm to learn sufficiently general risk knowledge.

III. Preliminaries and Problem Formulation

Dec-POMDP. MARL is typically modeled as a decentralised partially observable Markov decision process (Dec-POMDP) [38], which can be formulated as a tuple $\mathcal{G} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \mathcal{Z}, \mathcal{O}, N, \gamma >$, where $s \in \mathcal{S}$ represents the environmental state and $\mathcal{A}$ is the set of actions of all agents. At each time step t, each agent $i \in \{1, 2, ..., N\}$ chooses an action $a_i \in \mathcal{A}$, forming a joint action $\boldsymbol{u} = [a_1, a_2, ..., a_N] \in \mathcal{A}^N$. The environment gets into next state s_{t+1} through a dynamic transition function $\mathcal{P}(s_{t+1}|s_t, \boldsymbol{u}) : \mathcal{S} \times \mathcal{A}^N \times S \mapsto [0, 1]$, and the agent receives a reward $r(s_t, \boldsymbol{u}) : \mathcal{S} \times \mathcal{A}^N \mapsto \mathbb{R}$. In a cooperation scenario, we assume that all agents share the same reward and the reward function is bounded, i.e. $|r(s_t, \boldsymbol{u})| \leq R_{\max}$. Each agent has a local observation $o \in \mathcal{O}$ determined by the observation function $\mathcal{Z}(s, i) : \mathcal{S} \times N \mapsto \mathcal{O}$. We use $h_i \in \mathcal{H} \equiv (\mathcal{O} \times \mathcal{A})^*$ to represent the historical information of trajectory data. The policy $\pi_i(a_i|h_i)$ represents the probability of the agent i selecting action a under h. $\gamma \in [0, 1)$ is the discount factor used to calculate cumulative returns. The tuple $\mathcal{G}$ of all offline data forms the offline dataset $\mathcal{D}$ for multi-task.

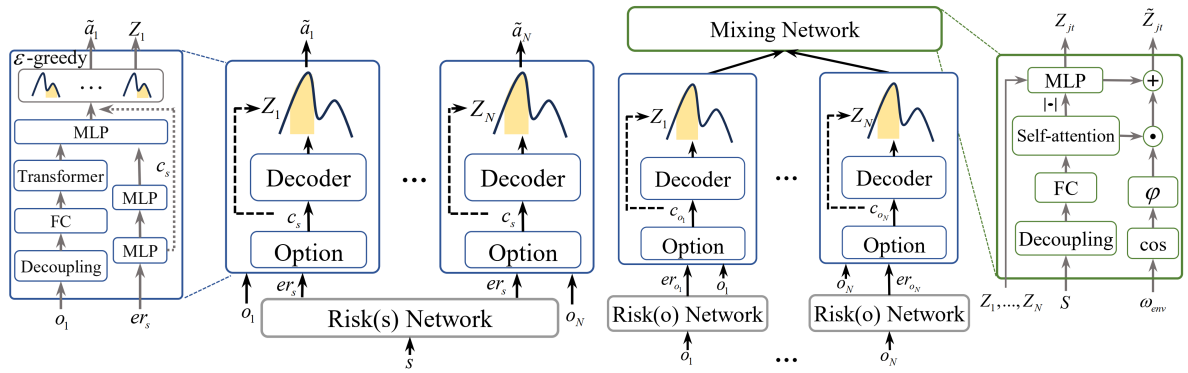


Fig. 2. Architecture of RIKD algorithm. Risk(s) network and Risk(o) network extract risk knowledge from observations and states, respectively, both using the self-attention mechanism. For detailed information, refer to the supplementary material.

This paper adopts the centralized training and decentralized execution (CTDE) framework, which minimizes the TD loss, $\min_{\mu} \mathbb{E}\left[\left(r+\gamma \max_{\boldsymbol{u}'} Q_{tot}(h', \boldsymbol{u}'; \mu^{-}) - Q_{tot}(h, \boldsymbol{u}; \mu)\right)^2\right]$, where Q_{tot} represents the joint state-action value and μ^{-} represents the target network parameters.

Distributional RL. Distributional RL [39] learns potential risks within the distributions and represents the policy value as a distribution $Z(s,a)$, where $\mathbb{E}[Z(s,a)] = Q(s,a)$. Under the multi-task setting, a multi-agent policy faces potential risks from unseen tasks. Therefore, we learn a cooperation policy based on its value distribution to improve its robustness. Given policy π, the distribution of state-action values $Z^{\pi}(s,a)$ satisfies the Bellman equation: $\mathcal{T}^{\pi} Z(s,u) \stackrel{D}{:=} r(s,u) + \gamma Z(s',u'), u' \sim \pi(\cdot|s)$, where $X \stackrel{D}{=} Y$ represents that the random variables X and Y follow the same probability distribution. Due to the difficulty in obtaining $Z(s,a)$ directly, we refer to the implicit quantile method in the IQN [22] and use the function $F_z^{-1}(s,a|\omega) = f(\varphi(s) \odot \phi(\omega))_u$ to approximate the return distribution, where $F_Z^{-1}(\omega)$ is the quantile function of the distribution $Z(s,a)$ on $\omega \in [0,1]$, $\omega \sim U(0,1)$, and φ, ϕ represent the differentiable functions that integrate state information and environmental risk ω, respectively. f is a subsequent non-linear mapping function that uses a fully connected layer to map $\varphi(s) \odot \phi(\omega)$ to the estimated return distribution, where $\odot$ denotes the element-wise product. The TD-Huber loss function at a given threshold k is:

$$\mathcal{L}_{TD-H}(s,a,r,s') = \frac{1}{N'_{env}} \sum_{i=1}^{N_{env}} \sum_{j=1}^{N'_{env}} \rho_{\omega}^{k}\left(\delta^{\omega,\omega'}\right) \quad (1)$$

where N_{env} and N'_{env} represent the number of samples $\omega, \omega' \sim U(0,1)$ used to calculate the loss, and the TD error is $\delta^{\omega,\omega'} = r + \gamma Z_{\omega'}(s',a') - Z_{\omega}(s,a)$, and

$$\rho_{\omega}^{k}(\delta) = \begin{cases} \frac{\delta^2}{2k} \left|\omega - \mathbb{I}\{\delta < 0\}\right|, \text{if } \delta \le k \\ \left|\omega - \mathbb{I}\{\delta < 0\}\right| \left(|\delta| - \frac{1}{2}k\right), \text{otherwise} \end{cases} \quad (2)$$

IV. Main Algorithm

RIKD learns the mapping from environmental knowledge to agent's risk responses from offline data. It then leverages this mapping, enabling agents to select optimal risk responses based on their perceived risk knowledge in a novel unseen task. As shown in Figure 2(a), we first utilize the Risk(s) network to extract risk knowledge from states s in the offline data, feeding this risk knowledge along with agent observations o into the risk attitude selector. We train the risk attitude selector using state-action records in offline data as supervisory signals. Subsequently, in an unseen task, we proceed with centralized training to coordinate the agents' actions with varying risk responses. This involves directly reusing the trained risk attitude selector and optimizing the parameters of the Risk(o) network, as shown in Figure 2(b). Therefore, during the deployment phase, the agents can acquire risk knowledge based on current observations and then utilize the risk attitude selector to output the optimal risk response adaptive to the current environmental risks.

A. Risk knowledge extraction networks

We use a self-attention mechanism and Transformer to extract environmental risk knowledge from state information and observation information in offline data. RIKD first extracts risk knowledge from state s, decouples s into ally information s^{ally} and enemy information s^{enemy}, and then uses MLP to project them into an embedding with a fixed dimension. Finally, based on the self-attention mechanism, er_s is obtained, which is the average level of attention s^{ally} has on (s^{ally}, s^{enemy}). Similarly, RIKD decouples observation o into its own information o^{own}, ally information o^{ally}, and enemy information o^{enemy} during the coordination policy optimization stage, and then extracts risk knowledge from observation o through a Transformer based on historical information h. Decoupling state s and observation o is to adapt to different tasks, while using attention mechanisms and Transformer is to make the agent pay more attention to the environmental risk knowledge hidden in the current information. The risk knowledge extraction network is shown in Figure 3.

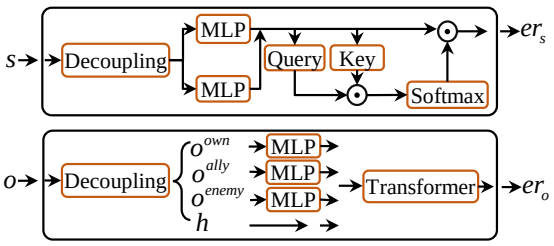


Fig. 3. Risk knowledge extraction networks.

B. Risk attitude selector

The risk attitude selector is an essential component of RIKD that can be applied between multiple tasks, with input being the observation o of each agent and the risk knowledge er. To learn risk-sensitive policy, the selector uses the Decoder and Option network to learn the policy value distribution Z and risk attitude c from the agent's observations o.

In practice, we use a combination of estimated values $\{q_j\}_{j=1,...,n_\tau}$ of n_τ quantiles at the quantile level $\{\tau_j\}_{j=1,...,n_\tau}$ to represent the distribution $Z(\tau)$ discretely. To enable the agent to have the ability to choose risk attitudes, we construct m options ($m < n_\tau$) in $Z(\tau)$ corresponding to different risk attitudes, i.e., $c = (c_1, ..., c_m)$. The risk attitude selector uses the option framework $c \sim f(c|er)$ to match the risk attitude of the agent that adapts to the current risk knowledge [40], then calculates the mean value function $z = \frac{1}{K}\sum_{k=(l-1)K+1}^{(l-1)K+K} q_k$ within the quantile region corresponding to the option (where $K = n_\tau/m$ represents that n_τ quantiles are divided into m windows, with each window containing K quantiles), and generates the joint action $\tilde{\boldsymbol{u}} = (\tilde{a}_1, ..., \tilde{a}_N)$ based on z, where $\tilde{a} \sim \varepsilon - greedy(z)$, $l \in [1, ..., m]$ represent the selected option.

The learning process for the risk attitude selector is entirely offline data-driven, aiming to generate policies for agents with varying risk attitudes based on their risk knowledge. We leverage the state information to learn risk knowledge, and then use it as input to train the selector network. The training objective adopts cross-entropy loss, which minimizes the difference between the policy value distribution z output by the risk attitude selector and the real action a in offline data. The optimization objective is as follows:

$$\mathcal{L}_1 = -\mathbb{E}_{(s,h,\boldsymbol{u})\sim\mathcal{D}}\left[\sum_{i=1}^{N}\mathbb{E}_{z_i\sim q(\cdot|h,er_s^i,c_s^i)}\left[a_i \log z_i\right]\right] \tag{3}$$

Among them, er_s^i represents the risk knowledge learned by the agent i from the state information, c_s^i is the option selected by the agent i based on er_s^i, and $q\left(\cdot|h, er_s^i, c_s^i\right)$ represents the risk attitude selector network. Due to the risk attitude selector is trained based on state information, after training, it only needs to freeze the network parameters.

C. Cooperation policy learning

Risk attitude selector utilizes state information for training, but during the decentralized deployment phase, we can only use the agent's local observation information. Therefore, we further consider utilizing the CTDE structure to optimize the risk knowledge extraction network based on the agent observation information by training cooperation policy.

Before learning cooperation policy, we first use a mixing network to decompose the value distribution of the joint policy [17], i.e., learning $Z_{jt}(s, \boldsymbol{u}) = \mathcal{F}(Z_1(o_1, a_1), ..., Z_N(o_N, a_N))$, where $\mathcal{F}$ represents the value decomposition function. Therefore, during the centralized training phase, we adopted a hypernetwork similar to the QMIX [41] to encode the state s into nonnegative weights, and generated a joint policy value distribution $Z_{jt}(s, h, \boldsymbol{u})$ by weighting the policy values distribution $Z_i(o_i, a_i)$ of the agent. Then, we use an implicit quantile network to integrate environmental risk $\omega_{env} \sim U(0, 1)$ into $\mathcal{F}$, explicitly reflecting environmental risk in the joint policy value distribution $\tilde{Z}_{jt}(s, h, \boldsymbol{u}, \omega_{env})$. Furthermore, we use the TD-Huber loss in Equation 1 to optimize the mixing network and risk knowledge extraction network, where the TD loss is $\delta = \tilde{Z}_{jt}(s, h, \boldsymbol{u}, \omega_{env}) - \left(r + \gamma\tilde{Z}'_{jt}(s', h', \boldsymbol{u}', \omega'_{env})\right)$ and $\tilde{Z}'_{jt}(s', h', \boldsymbol{u}', \omega'_{env})$ represents the value distribution of the target joint policy that integrates environmental risks. The joint policy in the value distribution $\tilde{Z}_{jt}(s, h, \boldsymbol{u}, \omega_{env})$ of the joint policy and the target value distribution $\tilde{Z}'_{jt}(s', h', \boldsymbol{u}', \omega'_{env})$ are different, where $\boldsymbol{u} = (a_1, ..., a_N), a_i \in \mathcal{D}$ and $\boldsymbol{u}' = (a_1, ..., a_N), a_i = \arg\max_{a_i} z_i$.

We integrate environmental risk into the $\tilde{Z}_{jt}$ in the mixing network, which is closer to the natural environment than the Z_{jt}, as Z_{jt} only considers state information. Therefore, for the optimal policy $\boldsymbol{u}' = (a_1, ..., a_N)$, we believe that Z_{jt} should be close to the true value $\tilde{Z}_{jt}$. In practice, we encourage the transformed joint policy value distribution estimate Z_{jt} to follow the actual distribution $\tilde{Z}_{jt}$ by defining the loss $\mathcal{L}_\omega$, to thoroughly learn environmental risks, where:

$$\mathcal{L}_\omega = Z_{jt}(s, h, \boldsymbol{u}') - \mathbb{E}_{\omega_{env}}\left[\tilde{Z}_{jt}(s, h, \boldsymbol{u}', \omega_{env})\right] \tag{4}$$

Due to the input of the risk attitude selector in Figure 2(b) is no longer state, but observation of the agent, the options output by the risk attitude selector may not be able to match the current environment risk. However, it may be difficult to learn the compact coordination relationship between the risk knowledge extraction network and the risk attitude selector solely by learning TD-Huber loss. Therefore, we refer to the approach of ODIS [12] and introduce an auxiliary objective to make the risk attitude selector learn as consistent as possible when facing state and observation. We use cross-entropy loss to define the difference between option c_o learned based on observation and option c_s learned based on state, which introduces cross-entropy loss:

$$\mathcal{L}_c = \sum_{i=1}^{n}\mathbb{E}_{(s,o)\sim\mathcal{D}}\left[D_{KL}\left(c_o^i||c_s^i\right)\right] \tag{5}$$

In brief, the total loss in the learning and optimization stage of the joint policy is $\mathcal{L}_2 = \mathcal{L}_{TD-H} + \alpha_1\mathcal{L}_\omega + \beta_1\mathcal{L}_c$, where α_1 and β_1 are the weights of the terms $\mathcal{L}_\omega$ and $\mathcal{L}_c$.

TABLE I

THE AVERAGE TEST WIN RATE ON FOUR QUALITY MARINE DATASETS UNDER STANDARD SETTINGS. WE ADOPT TWO VARIANT ALGORITHMS OF THE UPDET IN [12], UPDET-M (COMBINED WITH THE ODIS MIXING NETWORK) AND UPDET-L (COMBINED WITH THE VDN LINEAR DECOMPOSITION NETWORK), AS WELL AS THE BEHAVIOR CLONING (BC) ALGORITHM. IN THE RESULTS, BC-BEST REPRESENTS THE BEST TEST WIN RATE BETWEEN BC-T AND BC-R, WHERE BC-T REPRESENTS THE BC BASED ON THE TRANSFORMER STRUCTURE, AND BC-R REPRESENTS APPEND RETURN TO GO INFORMATION ON BC-T.

Task	Expert					Medium				
	BC-best	UPDeT-l	UPDeT-m	ODIS	RIKD	BC-best	UPDeT-l	UPDeT-m	ODIS	RIKD
Source tasks										
3m	97.7±2.6	71.0±16.6	82.8±16.0	98.4±2.7	**99.1±0.7**	65.4±14.7	56.6±14.2	51.2±3.4	**85.9±10.5**	34.1±18.4
5m6m	50.4±2.3	12.1±12.6	17.2±28.0	53.9±5.1	**55.3±5.9**	21.9±3.4	5.6±4.8	6.3±4.9	22.7±7.1	**23.0±4.9**
9m10m	**95.3±1.6**	26.6±12.0	3.1±5.4	80.4±8.7	91.4±4.7	63.8±10.9	34.4±13.9	28.5±10.2	**78.1±3.8**	50.1±24.0
Unseen tasks										
4m	92.1±3.5	28.6±21.6	33.0±27.1	95.3±3.5	**97.0±4.3**	48.8±21.1	21.6±17.2	14.1±5.2	61.7±17.7	**65.9±10.8**
5m	87.1±10.5	40.1±25.9	33.6±40.2	89.1±10.0	**95.3±5.9**	76.6±14.1	77.4±16.0	67.2±21.3	85.9±11.8	**88.1±6.9**
10m	90.5±3.8	33.9±25.2	54.7±44.4	93.8±2.2	**95.4±3.5**	56.2±20.6	36.8±20.7	32.9±11.3	61.3±11.3	**64.3±26.5**
12m	70.8±15.2	10.9±18.9	17.2±28.0	58.6±11.8	**72.1±16.7**	24.0±10.5	4.0±5.3	3.2±3.8	35.9±8.1	**48.5±32.4**
7m8m	18.8±3.1	0.8±1.4	0.0±0.0	**25.0±15.1**	23.0±3.6	1.6±1.6	2.4±2.6	0.0±0.0	**28.1±22.0**	4.0±32.4
8m9m	15.8±3.3	1.6±1.6	0.0±0.0	19.6±6.0	**21.3±9.9**	3.1±3.8	3.1±3.1	2.3±2.6	4.7±2.7	**6.5±5.2**
10m11m	45.3±11.1	0.8±1.4	0.0±0.0	**42.2±7.2**	36±13.7	19.7±8.9	2.4±1.4	4.0±3.4	29.7±15.4	**30.6±15.9**
10m12m	1.0±1.5	0.0±0.0	0.0±0.0	1.6±1.6	**2.1±1.4**	0.0±0.0	0.0±0.0	0.0±0.0	**1.6±1.6**	0.5±0.5
13m15m	0.0±0.0	0.0±0.0	0.0±0.0	**2.3±2.6**	0.8±1.4	0.6±1.3	0.0±0.0	0.0±0.0	**1.6±1.6**	0.0±0.0
	Medium-expert					Medium-replay				
Source tasks										
3m	67.7±23.7	50.1±23.9	85.2±17.9	73.6±22.0	**75.5±31.7**	81.1±8.8	27.3±25.9	41.4±20.1	83.6±14.0	**90.2±3.2**
5m6m	31.3±6.3	2.3±2.6	1.6±1.6	9.4±2.2	**32.0±15.1**	25.0±3.1	0.8±1.4	0.8±1.4	16.6±4.7	**19.6±5.8**
9m10m	26.0±13.9	27.7±24.1	24.3±18.7	31.3±14.5	**37.7±15.0**	33.4±13.1	2.3±4.1	0.8±1.4	34.4±8.0	**48.5±12.2**
Unseen tasks										
4m	81.3±18.9	41.0±8.0	43.9±39.0	**82.8±13.5**	68.3±40.0	61.5±9.0	23.4±15.5	35.9±12.6	55.6±14.5	**56.6±19.0**
5m	74.0±2.9	65.7±10.1	33.6±40.2	82.8±17.7	**86.5±15.8**	75.0±24.2	54.7±23.5	61.7±20.3	**96.1±4.1**	74.4±42.4
10m	78.1±6.7	39.8±20.1	32.8±38.1	**82.8±16.8**	49.4±37.2	82.4±8.2	8.6±8.7	11.0±7.8	84.4±15.1	**90.1±8.6**
12m	64.8±24.3	9.4±7.9	9.4±8.6	**81.3±20.6**	39.6±39.4	83.4±4.5	2.3±4.1	2.3±2.6	84.4±6.6	**86.0±8.9**
7m8m	13.3±4.5	4.0±4.2	2.3±4.1	15.6±4.4	**18.6±10.7**	7.3±6.4	2.3±2.6	1.6±2.7	**9.4±2.2**	3.8±2.8
8m9m	10.2±4.6	5.6±4.8	9.5±8.6	10.9±4.7	**18.1±9.1**	11.5±3.9	0.8±1.4	0.8±1.4	11.7±8.7	**12.3±1.9**
10m11m	26.6±4.7	8.0±12.2	11.8±8.1	33.6±8.9	**35.4±13.3**	**46.8±6.6**	2.3±4.1	0.8±1.4	35.9±5.2	43.4±15.7
10m12m	0.0±0.0	0.0±0.0	0.0±0.0	1.6±1.6	**2.3±3.2**	1.6±2.7	0.0±0.0	0.0±0.0	**2.3±1.4**	0.8±0.8
13m15m	0.8±1.4	0.0±0.0	0.0±0.0	2.3±2.6	**3.1±3.3**	1.6±1.6	0.0±0.0	0.0±0.0	2.4±1.4	**2.6±1.4**

D. RIKD (online)

The above content introduces how we can use offline data to learn generalizable risk response. However, we found that learning generalizable risks in multi-task is not only effective in offline versions, but also applicable in online versions. Hence, to enhance RIKD's ability to adapt to environmental changes in target tasks, we propose an online version of the multi-task multi-agent policy transfer method RIKD (online) by integrating a risk knowledge extraction network and a risk attitude selector in the UPDeT [11].

The optimization objectives of RIKD (online) include $\mathcal{L}_{TD-H}$ and $\mathcal{L}_{\omega}$, which is the same as RIKD. However, due to the lack of offline data support, we cannot directly obtain the optimal option through greedy policy. Therefore, RIKD (online) refers to the option error in the Quota [20] when optimizing the distribution of joint policy values:

$$\mathcal{L}_{opt} = \sum \left(q_c - \left(r + \gamma \left(\upsilon \max_{c'} q'_{c'} + (1-\upsilon) q'_c\right)\right)\right) \quad (6)$$

where $q'_{c'}$ represents the option value output by the target policy network, and υ represents the termination probability of the option c. Therefore, the total loss of RIKD (online) is $\mathcal{L}_{online} = \mathcal{L}_{TD-H} + \alpha_2 \mathcal{L}_{\omega} + \beta_2 \mathcal{L}_{opt}$, where α_2 and β_2 are the weights. In the decentralized deployment phase, RIKD (online) only needs to reuse the trained agent policy network in new tasks and fine tune the network parameters.

RIKD (online) has proven from an online perspective that risk knowledge can be transferred as more general knowledge, which is a powerful supplement to RIKD. We provide RIKD (online) algorithm's pseudocode Algorithm 1.

V. EXPERIMENT

In this section, we test the policy generalization performance of our method on different benchmark tasks. (1) Multi-task generalization ability. We validated the policy generalization ability of RIKD by designing multi-task scenarios under risk factors in the collaborative navigation (CN) and the SMAC benchmark environments. (2) The impact of offline data. To further verify the versatility of risk knowledge, we analyzed how the type of source task affects algorithm performance. (3) Robustness. We designed ablation experiments to investigate the impact of essential components on algorithm performance.

A. Benchmark and risk construction

Environment. We use the multi-agent benchmark environment MPE [42] and SMAC to validate the algorithm's performance. We constructed multiple CN tasks in the MPE, where the goal of the multi-agent is to learn to cover all

Algorithm 1 RIKD (online)

Initialize policy network parameters and mixing network parameters.
Initialize target policy network parameters and target mixing network parameters.
Initialize Experience Replay (ER) buffer.
Initialize option framework parameters and set quantiles ω_{env}.
for $episode = 1$ to M **do**
  Collect state s and agents' observation $o_{i=1,...,N}$.
  for $t = 1, ..., T$ **do**
    Calculate the current policy value distribution Z and option value q_c of the agent.
    Select option c based on the option value and calculate the mean of the action value function within the option based on c.
    Choose actions a based on the $\varepsilon - greedy$.
    After executing joint action $\boldsymbol{u}$, the agent receives observations and rewards (s', o', r).
    Store $(s, h, \boldsymbol{u}, c, r, s', h')$ in the ER and sample a small batch of samples.
    Obtain the mixing network output Z_{jt}, $\tilde{Z}_{jt}$, and the target network output Z'_{jt}, $\tilde{Z}'_{jt}$.
    Calculate TD-Huber loss $\mathcal{L}_{TD-H}$.
    Calculate $\mathcal{L}_{\omega}$.
    Calculate option loss $\mathcal{L}_{opt}$.
    $\mathcal{L}_{online} = \mathcal{L}_{TD-H} + \alpha_2 \mathcal{L}_{\omega} + \beta_2 \mathcal{L}_{opt}$.
    Update policy network and mixing network parameters by minimizing loss $\mathcal{L}_{online}$.
    Update target network parameters with period I.
  end for
end for

landmarks without collision. To adapt to multi-task settings, we constructed multiple sets of tasks in the hard mode of the SMAC. And we used the QMIX to collect offline data and organized four different quality datasets: expert, medium, medium expert, and medium replay.

(1) The MPE task abstracts cooperative tasks in the real world as interactions between particles. Taking the collaborative navigation (CN) task as an example, we construct multiple CN tasks with different numbers of agents. In the CN task, n agents need to collaborate to navigate to all landmarks (i.e. target points) in the task, as shown in Figure 4(a). Each agent calculates rewards based on the distance from all agents to each landmark. If an agent conflicts with other agents, it will be punished. Therefore, agents must learn to cover all landmarks while avoiding collisions. The agent's observation in this task includes information such as speed and distance, and there are five actions of the agent, including movement on the x-axis and y-axis plus no action. Each game ends with reaching the set number of steps.

(2) SMAC tasks aim to evaluate the ability of multi-agent collaboration to solve complex adversarial tasks. The SMAC is an environment used to validate MARL on StarCraft 2, where micro control functions can provide fine-grained control over individual units and is often used as a benchmark testing

(a) CN (b) SMAC(3m) (c) SMAC(2s3z)

Fig. 4. Benchmark environment.

environment for discrete MARL algorithms. The maps in SMAC are carefully designed, with each scene including a confrontation between two armies, and the initial position, quantity, and unit type of each army vary depending on the scene. In this paper, we used two types of scenarios, including the marine scenario and the stacker-zealot scenario, with difficulty levels set to hard, as shown in Figure 4(b)(c). In the marine scenario, we learned from source tasks 3m (homologous & metric), 5m_vs_6 (homologous & metric) m, and 9m_vs_10m, respectively. In the stacker-zealot scenario, we learned from source tasks 2s3z, 2s4z, and 3s5z (heterologous & metric).

Benchmark algorithm. Currently, there are relatively few baseline algorithms that can be compared in the field of multi-task MARL. Therefore, this paper mainly compares RIKD with the most advanced offline multi-task algorithm ODIS and the online transfer algorithm UPDeT. ODIS is entirely suitable for the multi-task generalization of multi-agent policy. Although the UPDeT is not set for multi-task, it can also verify the general of risk knowledge.

Risk design. In the SMAC task, we adopted the risk factors designed in the DRIMA [18] to verify the effectiveness of RIKD. (1) **Dilemma**: Rewards not only consider our damage to the enemy (positive rewards), but also our damage (negative rewards), meaning that the agent needs to learn better cooperation policy to reduce risk. (2) **Exploration**: Reduce the decay rate of exploration rate, allowing for long-term high exploration during the training phase and increasing the difficulty of cooperation. (3) **Noise**: A random agent among allies randomly selects actions (random cooperation policy) with a 30% probability when selecting a policy. In the dilemma setting, we changed the reward_only_positive in the *sc2.yaml* of the SMAC environment from $True$ to $False$, indicating that the reward needs to consider our damage (negative reward). Under the exploration setting, we have changed the exploration annexing schedule from "50k" to "500k" time steps, which reduces the decay rate of the exploration rate and increases the difficulty of multi-agent cooperation. Under the noise setting, we select actions based on policy $\varepsilon - greedy$ in the distribution of agent policy value, and then interfere with the policy of one ally, i.e., the ally randomly selects actions with a 30% probability. The exploration and noise risk in the CN tasks are the same as the setting method in SMAC, while the standard (S) setting includes the risk of dilemma, so there is no need to set them separately. In addition, we also used the QMIX to collect offline data under three risk settings: dilemma, exploration, and noise. In the CN tasks, since the

TABLE II
PROPERTIES OF OFFLINE DATASETS WITH DIFFERENT QUALITIES ON SMAC TASKS.

Task	Quality	Trajectories(S/D/E/N)	Average return(S/D/E/N)	Average win rate(S/D/E/N)
3m	Expert	2000/2000/2000/2000	19.8929/16.9868/18.6295/16.6613	0.991/1/0.8875/0.7375
	Medium	2000/2000/2000/2000	13.9869/11.4575/15.1534/12.3592	0.5402/0.625/0.6125/0.4188
	Medium-expert	4000/4000/4000/4000	16.9399/14.2222/16.8915/14.5103	0.756/0.8125/0.75/0.5782
	Medium-relpay	3630/4114/2759/4116	N/A	N/A
5m6m	Expert	2000/2000/2000/2000	17.3424/12.365/18.4745/13.4476	0.7185/0.6875/0.8375/0.3875
	Medium	2000/2000/2000/2000	12.6408/9.0098/14.3502/12.521	0.2751/0.4062/0.4562/0.3063
	Medium-expert	4000/4000/4000/4000	14.9916/10.6874/16.4124/12.9843	0.4968/0.5469/0.6469/0.3469
	Medium-relpay	32607/4298/1307/1163	N/A	N/A
9m10m	Expert	2000/2000/2000/2000	19.614/14.2715/19.67/19.1575	0.9431/0.9563/0.9263/0.8812
	Medium	2000/2000/2000/2000	15.5049/9.8616/16.895/15.5718	0.4146/0.5125/0.6062/0.4688
	Medium-expert	4000/4000/4000/4000	17.5594/4.9308/18.2825/17.3647	0.6789/0.7344/0.7663/0.675
	Medium-relpay	13731/3574/646/81	N/A	N/A
2s3z	Expert	2000/2000/2000/2000	19.7655/14.2379/19.6515/18.246	0.9602/1/0.9375/0.7188
	Medium	2000/2000/2000/2000	16.6279/9.9906/16.8199/16.3265	0.4146/0.5813/0.4875/0.4688
	Medium-expert	4000/4000/4000/4000	18.196712.1143/18.2357/17.2863	0.6789/0.7907/0.7125/0.5938
	Medium-relpay	4505/4970/608/1200	N/A	N/A
2s4z	Expert	2000/2000/2000/2000	19.7402/13.2069/18.6706/18.3698	0.9509/0.9625/0.7688/0.7438
	Medium	2000/2000/2000/2000	16.8735/9.1544/6.7776/16.1619	0.4965/0.5375/0.451/0.425
	Medium-expert	4000/4000/4000/4000	18.3069/11.1807/12.7241/17.2659	0.7237/0.75/0.6099/0.5844
	Medium-relpay	6172/1985/4638/1787	N/A	N/A
3s5z	Expert	2000/2000/2000/2000	19.785/12.9535/19.2365/19.5416	0.9518/0.975/0.825/0.8875
	Medium	2000/2000/2000/2000	16.3126/9.0605/17.5949/16.9961	0.3114/0.5062/0.4813/0.4562
	Medium-expert	4000/4000/4000/4000	18.0488/11.007/18.4157/18.2689	0.6316/0.7406/0.6532/0.6719
	Medium-relpay	11528/3349/4916/1281	N/A	N/A

rewards in standard settings CN tasks include both positive and negative rewards, so standard settings are also considered a dilemma.

B. Offline data collection

The RIKD is based on the most advanced open-source framework PYMARL[1] implementation currently available. And then we refer to the data collection method of the ODIS [12] to collect offline data in risk environments, i.e., we use the QMIX to collect data. We have produced offline data of four qualities: expert, medium, medium expert, and medium replay. The offline data without risk interference under standard settings comes from the ODIS. The detailed information of SMAC's offline data can be found in Table II. The meanings of these four datasets are as follows:

- Expert dataset: The expert policy was obtained by the QMIX after 2 million training steps, and trajectory data was collected based on the expert policy. We also recorded the expert policy's average return and test win rate.
- Medium dataset: The test win rate of the medium dataset corresponding to the medium policy is approximately half that of the expert policy, and the data collected using the medium policy is called the medium dataset.
- Medium expert dataset: A mixture of all data from the expert dataset and all data from the medium dataset, expanding the diversity of data in offline datasets.
- Medium replay dataset: A replay buffer for training medium policy, containing trajectory data from low to medium quality.

In the CN tasks, the reward of the agent is represented by the distance from the ball to the landmark, and the closer the distance is to 0, the higher the quality of task completion. Therefore, in this task, the expert dataset represents the expert policy obtained by the QMIX after 5 million steps of training, and based on the trajectory data collected by the expert policy, we also recorded the average return corresponding to the expert policy. The medium dataset represents data collected using a medium policy, where the average return of the medium policy used is approximately half that of the expert policy. The detailed information of CN's offline data can be found in Table III.

[1]https://github.com/oxwhirl/pymarl

TABLE III
PROPERTIES OF OFFLINE DATASETS WITH DIFFERENT QUALITIES ON CN TASKS.

Task	Quality	Trajectories(S/D/E/N)	Average return(S/D/E/N)
CN_2	Expert	2000/2000/2000	-38.71/-39.99/-42.74
	Medium	2000/2000/2000	-43.28/-44.88/-47.69
CN_3	Expert	2000/2000/2000	-52.5/-60.28/-53.5
	Medium	2000/2000/2000	-59.56/-66.12/-60.42

C. Multi-task generalization performance

CN tasks.

We constructed four types of the CN tasks with different numbers of agents, where CN_2 and CN_3 were used as source tasks to generate offline data (CN_n represents n agents navigating to n targets). Then, we generalized them on unseen tasks CN_4 and CN_5. Taking the expert datasets and medium datasets as examples, we compared RIKD with the currently best-performing multi-task algorithm ODIS under three risk settings, and the results are shown in Table IV and Table V. We found that RIKD performs better than ODIS on both source and unseen tasks, which also proves that risk knowledge is more general in the CN tasks. Specifically, under the two risk settings of exploration and noise, RIKD performs better than

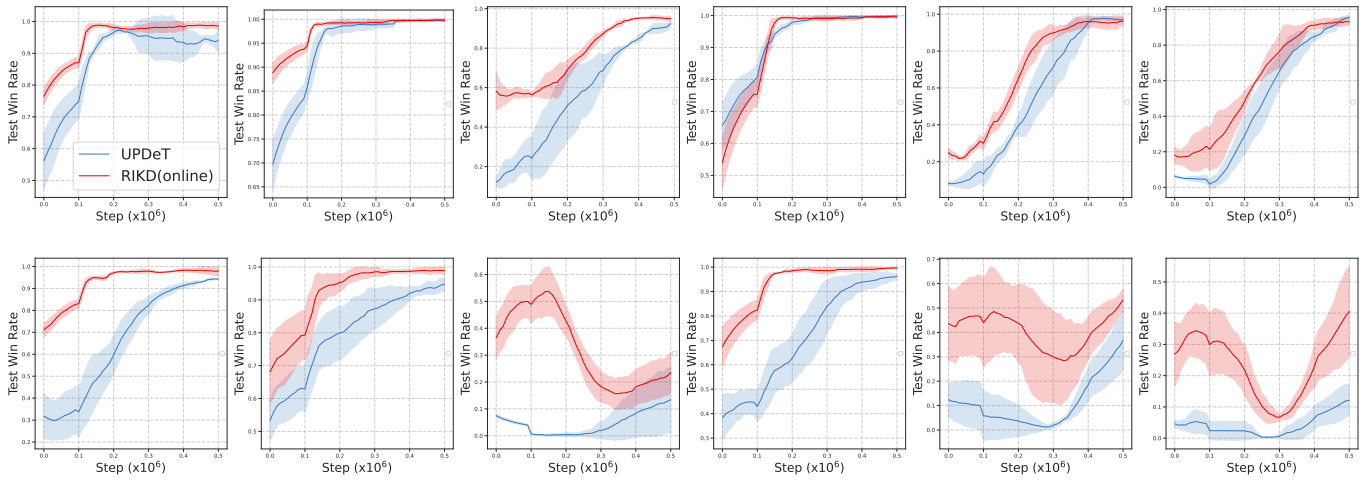


Fig. 5. Generalization performance of RIKD (online). The first row represents the transfer from 3m to 5m, and the second row represents the transfer from 3s5z to 2s3z. The first to sixth columns represent the transfer from the source task with risk set as (standard, dilemma, exploration+noise, standard, standard, standard) to the target task with risk set as (standard, dilemma, exploration+noise, dilemma, exploration, exploration+noise).

ODIS. This is because RIKD incorporates environmental risks into the mixing network for learning during the joint policy optimization stage, while ODIS cannot effectively perceive risks by discovering skill.

TABLE IV
AVERAGE TEST RETURN MEAN OF RIKD AND ODIS IN THE EXPERT DATASET OF THE CN TASKS.

Task	Standard (Dilemma)		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
CN_2	-47.6±9.5	**-42.4±0.9**	-47.6±9.7	**-44.8±2.3**	-61.1±8.8	**-46.4±0.7**
CN_3	-60.9±6.5	**-58.7±1.3**	-64.6±4.7	**-61.6±2.9**	-69.9±6.5	**-61.4±2.2**
Unseen tasks						
CN_4	-73.4±4.2	**-73.4±1.8**	-77.0±2.3	**-75.1±5.6**	-78.5±5.3	**-72.7±3.8**
CN_5	**-83.9±2.8**	-84.2±2.6	-88.8±4.5	**-88.4±7.6**	-85.9±2.6	**-84.4±5.0**

TABLE V
AVERAGE TEST RETURN MEAN OF RIKD AND ODIS IN THE MEDIUM DATASET OF THE CN TASKS.

Task	Standard (Dilemma)		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
CN_2	-57.8±9.2	**-47.3±0.8**	-61.1±6.5	**-48.6±1.2**	-52.1±6.8	**-49.3±1.0**
CN_3	-68.4±4.9	**-67.9±7.9**	-71.1±1.7	**-67.4±1.5**	-64.0±4.4	**-62.0±0.3**
Unseen tasks						
CN_4	**-79.8±4.0**	-81.3±1.8	**-84.3±10.5**	-89.7±2.9	-75.9±2.2	**-73.4±1.1**
CN_5	-89.3±6.6	**-89.2±2.4**	-93.5±16.4	**-92.4±8.8**	-87.1±1.9	**-84.1±2.7**

SMAC tasks.

We test the multi-task generalization performance of different algorithms in the standard settings SMAC environment. Table I shows the experimental results on three offline task datasets (3m, 5m6m, 9m10m) with four different data qualities. The results of the BC best, UPDeT-l, UPDeT-m, and ODIS algorithms are all from ODIS. The RIKD reported better results than the ODIS for most unseen tasks on four quality offline datasets. Although the number of agents varies in marine tasks, choosing similar risk attitudes by agents when faced with potential environmental risks in different tasks can contribute to enhancing the generalization performance of cooperation policy. We have noticed that the ODIS has also shown good performance, but there are significant differences in performance on different quality datasets.

TABLE VI
THE AVERAGE TEST WIN RATE ON THE MARINE TASK UNDER THREE RISK SETTINGS (EXPERT DATASETS)

Task	Dilemma		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
3m	**98.8±0.4**	98.3±1.2	90.6±2.5	**95.6±2.5**	48.4±10.9	**62.2±14.9**
5m6m	**44.0±4.5**	40.1±17.7	53.9±13.8	**59.3±27.9**	4.4±2.0	**10.6±4.1**
9m10m	57.9±22.7	**95.4±3.8**	71.8±9.2	**94.1±4.5**	40.9±12.0	**78.1±6.6**
Unseen tasks						
4m	85.4±2.2	**93.3±8.5**	67.8±20.1	**88.5±9.1**	39.6±9.4	**52.1±16.2**
5m	81.0±9.6	**92.9±8.7**	73.0±20.2	**93.0±14.3**	56.0±12.3	**77.2±18.3**
10m	64.1±25.7	**97.9±1.6**	76.1±3.9	**82.0±31.9**	38.1±24.2	**75.8±17.8**
12m	36.9±22.6	**63.1±30.8**	57.4±11.6	**61.0±38.6**	14.0±5.3	**55.5±25.1**
7m8m	21.1±18.7	**39.6±7.1**	10.3±5.0	**22.0±12.2**	4.6±2.4	**6.5±3.1**
8m9m	26.9±15.0	**55.6±14.6**	**31.5±15.3**	25.5±18.6	11.8±7.2	**15.7±7.9**
10m11m	42.0±29.3	**84.1±7.1**	53.1±10.5	**63.8±32.0**	22.8±8.9	**59.9±13.3**
10m12m	5.9±7.2	**14.8±8.3**	2.6±1.6	**4.6±6.1**	0.6±0.6	**3.0±2.4**

To further verify the generalization performance of RIKD, we conducted experiments under three risk settings for marine tasks with varying data quality, as shown in the Table VI, Table VII, Table VIII and Table IX. Although RIKD did not have a significant advantage in the source task under dilemma, its test win rate was significantly better than ODIS in unseen tasks. RIKD has achieved significant advantages in exploration and noise risk settings. This is because RIKD considers environmental risks in the joint policy optimization stage and matches risk attitudes adapted to the current environmental risks for each agent through a risk attitude selector, resulting in significant performance improvement in different risk settings. In addition, we further tested the generalization performance of the RIKD on the stalker-zealot task, and the results are shown in Table X. Similarly, we found that the performance difference between the RIKD based on risk knowledge discovery and the ODIS based on skill knowledge

TABLE VII
THE AVERAGE TEST WIN RATE ON THE MARINE TASK UNDER THREE RISK SETTINGS (MEDIUM DATASETS).

Task	Dilemma		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
3m	71.3±9.7	**77.1±6.7**	80.9±3.0	**82.2±4.9**	31.4±13.5	**37.4±6.1**
5m6m	**24.9±7.2**	20.6±10.8	46.3±13.2	**54.0±5.7**	5.0±4.7	**6.4±2.6**
9m10m	71.0±3.0	**78.1±6.3**	65.9±13.4	**71.9±10.0**	27.1±15.7	**27.2±4.2**
Unseen tasks						
4m	82.2±2.6	**89.4±4.8**	87.5±3.9	**90.1±9.6**	36.9±8.1	**48.0±8.7**
5m	89.1±6.4	**94.6±3.6**	97.4±3.2	**99.6±0.6**	56.0±24.4	**78.1±9.2**
10m	96.9±1.2	**97.9±1.6**	66.8±11.3	**77.4±23.4**	66.6±16.9	**78.9±23.6**
12m	87.3±9.6	**93.7±8.8**	**44.6±15.8**	38.2±36.3	32.9±12.7	**54.3±25.3**
7m8m	35.9±13.6	**37.4±4.0**	18.0±9.7	**29.6±14.7**	2.9±1.1	**3.0±1.8**
8m9m	37.4±12.9	**48.2±10.6**	**32.4±10.8**	26.5±6.6	**6.1±3.2**	5.4±3.0
10m11m	56.5±6.8	**68.4±8.6**	46.4±6.8	**58.9±11.9**	15.1±7.2	**24.4±7.7**
10m12m	3.5±2.1	**4.6±3.2**	**2.8±2.2**	2.3±2.1	0.0±0.0	**0.4±0.8**
13m15m	1.4±1.4	**9.7±4.8**	**0.9±0.8**	0.1±0.3	0.1±0.3	**0.8±0.5**

TABLE VIII
THE AVERAGE TEST WIN RATE ON THE MARINE TASK UNDER THREE RISK SETTINGS (MEDIUM-EXPERT DATASETS).

Task	Dilemma		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
3m	85.5±5.6	**85.6±10.2**	81.0±11.4	**86.6±4.3**	33.4±23.0	**40.4±8.5**
5m6m	35.0±8.9	**36.6±11.7**	39.6±9.2	**52.0±20.4**	4.3±4.4	**7.6±2.9**
9m10m	67.4±16.4	**91.4±6.5**	52.9±15.7	**86.9±7.7**	20.1±9.3	**51.3±11.2**
Unseen tasks						
4m	79.8±14.4	**94.1±3.3**	82.9±13.2	**91.9±7.0**	39.8±12.1	**43.0±12.4**
5m	87.9±12.3	**99.3±0.7**	75.1±16.2	**99.6±0.8**	50.3±28.5	**82.7±10.8**
10m	84.3±10.1	**98.0±1.4**	57.8±23.0	**90.2±13.5**	55.6±18.5	**87.3±6.2**
12m	72.1±13.1	**97.9±1.6**	47.0±35.1	**65.7±25.1**	32.3±19.6	**66.7±9.7**
7m8m	41.0±12.3	**46.9±13.3**	15.9±11.5	**22.6±6.6**	3.6±2.4	**4.2±1.8**
8m9m	43.1±12.9	**46.9±9.4**	30.5±19.6	**59.4±16.1**	7.8±3.0	**9.3±4.7**
10m11m	53.6±15.2	**84.0±7.9**	33.9±19.8	**69.1±14.7**	25.5±12.5	**39.0±16.3**
10m12m	3.8±3.2	**13.9±6.8**	2.0±2.7	**2.3±3.1**	0.5±0.2	**1.8±1.7**
13m15m	2.1±1.5	**10.6±6.3**	2.6±4.9	**5.1±7.6**	0.0±0.0	**1.7±2.3**

discovery is relatively small under standard settings, but the ODIS has almost no adaptability under the three risk settings, while the RIKD has a significant generalization advantage.

TABLE IX
THE AVERAGE TEST WIN RATE ON THE MARINE TASK UNDER THREE RISK SETTINGS (MEDIUM-REPLAY DATASETS).

Task	Dilemma		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
3m	61.3±6.4	**72.0±7.7**	0.8±1.1	**8.0±11.1**	38.0±18.1	**39.4±5.8**
5m6m	9.5±3.4	**25.1±5.2**	20.3±9.8	**48.9±7.4**	3.4±2.6	**4.9±2.7**
9m10m	21.3±7.4	**54.0±16.5**	10.0±7.2	**48.5±27.1**	18.9±8.1	**25.0±5.9**
Unseen tasks						
4m	55.9±18.3	**86.1±10.2**	22.4±19.9	**73.4±39.6**	40.5±7.3	**51.5±10.3**
5m	77.3±12.6	**95.2±4.6**	63.1±14.3	**99.8±0.3**	62.6±7.3	**76.1±12.8**
10m	57.8±14.2	**98.3±1.3**	32.6±21.1	**78.4±33.7**	75.1±9.3	**82.1±17.3**
12m	46.6±14.4	**95.7±3.6**	20.1±19.6	**65.8±33.6**	**51.6±12.3**	51.0±41.0
7m8m	10.3±7.8	**25.9±25.5**	12.5±17.7	**28.2±16.5**	4.3±2.8	**10.4±5.5**
8m9m	15.8±6.1	**33.0±12.2**	8.8±7.3	**40.4±17.1**	4.6±2.9	**10.6±3.1**
10m11m	20.1±9.2	**59.1±16.8**	12.3±10.1	**54.0±30.1**	17.6±6.2	**21.8±7.7**
10m12m	1.0±1.6	**3.8±2.2**	0.1±0.3	**1.6±1.7**	0.0±0.0	**0.4±0.8**
13m15m	1.6±2.2	**10.8±3.0**	0.4±0.6	**7.8±3.8**	0.1±0.3	**0.9±1.0**

In addition, to verify the online generalization ability of risk knowledge, we provide RIKD (online) results on the SMAC task. We present the transfer results from simple task 3m to complex task 5m and the transfer from complex task 3s5z to simple task 2s3z. We found that the transfer performance of RIKD (online) is better than that of the UPDeT, especially in the early stages of transfer. This also indicates the adaptability of RIKD (online) to potential risks in unseen tasks. Furthermore, we demonstrate the transfer of RIKD (online) between tasks with asymmetric risk, which involves pretraining from standard settings tasks and then transferring to other tasks with three risk settings. The results are shown in Figure 5. Although the risks between pretraining and unseen tasks are inconsistent, RIKD (online) also demonstrates better results than the UPDeT.

TABLE X
THE AVERAGE TEST WIN RATE ON THE STACKER-ZEALOT TASK DATASET UNDER THREE RISK SETTINGS (USING THE EXPERT DATASET AS AN EXAMPLE).

Task	Dilemma		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks						
2s3z	77.3±12.9	**88.5±4.8**	84.0±16.4	**97.9±1.4**	24.1±13.1	**52.6±18.0**
2s4z	52.9±14.6	**80.3±12.6**	57.9±15.8	**81.3±12.7**	11.0±4.8	**42.0±19.6**
3s5z	60.5±5.0	**86.9±16.8**	58.4±11.6	**81.4±8.6**	22±8.1	**58.4±6.8**
Unseen tasks						
1s3z	55.4±33.9	**74.5±34.0**	37.2±30.8	**86.9±7.7**	5.5±2.8	**13.1±6.4**
1s4z	31.9±27.1	**50.6±12.9**	35.4±20.7	**62.9±10.0**	0.9±1.0	**6.9±4.9**
1s5z	24.2±12.3	**32.6±24.9**	26.0±21.8	**43.8±7.9**	2.0±4.1	**7.1±4.9**
2s5z	46.5±5.7	**59.6±16.7**	51.9±19.4	**66.1±11.6**	13.3±6.9	**48.2±21.8**
3s3z	51.7±17.2	**66.6±21.8**	71.1±11.7	**90.2±8.7**	9.1±6.0	**39.1±22.1**
3s4z	60.2±7.1	**83.8±15.9**	67.3±10.0	**90.9±4.1**	16.5±4.0	**68.1±11.4**
4s3z	30.6±7.0	**51.5±17.5**	45.5±14.1	**77.0±14.0**	5.3±5.6	**15.1±13.2**
4s4z	30.3±9.9	**44.8±19.1**	36.5±10.7	**79.3±6.6**	12.6±7.5	**24.5±20.6**
4s5z	**34.0±12.7**	32.4±11.7	36.5±8.8	**57.4±13.9**	9.3±6.6	**28.0±19.4**
4s6z	20.0±7.7	**33.6±13.2**	13.0±6.7	**34.9±20.3**	14.1±6.9	**23.1±18.9**

D. The impact of offline data

To demonstrate that the risk knowledge is sufficiently general and is insensitive to offline data, we only use a relatively simple source tasks (such as 3m, 5m6m, and 2s3z) for learning, with other tasks as the unseen tasks. Through this setting, it is possible to effectively reduce the impact of the diversity of source tasks on learning general knowledge. When the number and types of source tasks are sufficient, they can almost cover any unseen task. Still, learning general knowledge from fewer and single-type source tasks requires stricter algorithm performance. Therefore, we conducted experiments on three types of multi-task under standard settings and three risk settings, and the results are shown in Table XI. Although RIKD performs average on the source task, it showed significantly better performance than ODIS on almost all unseen tasks, whether under standard or risk settings. This also proves that risk knowledge can serve as an underlying knowledge for generalization between multi-task.

E. Ablation experiment

RIKD uses an option framework in the risk attitude selector to select risk attitudes for agents, so we conducted ablation experiments on this component to verify the algorithm's robustness. The two most essential parameters in the risk attitude selector are the number of quantiles n_τ and options m. We designed a simple multi-task generalization experiments for

TABLE XI
COMPARISON OF PERFORMANCE BETWEEN RIKD AND ODIS BASED ON OFFLINE DATA LEARNING FROM A SINGLE SOURCE TASK.

Task	Standard		Dilemma		Exploration		Noise	
	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD	ODIS	RIKD
Source tasks								
3m	**99.1±1.0**	98.9±0.5	**98.5±2.1**	97.8±2.0	96.4±1.7	**96.6±1.8**	**34.9±9.0**	7.9±5.3
Unseen tasks								
4m	63.6±9.8	**78.5±15.2**	74.4±5.5	**82.3±10.9**	31.9±21.7	**69.3±14.0**	11.4±11.4	**29.4±6.7**
5m	24.3±22.4	**52.9±32.3**	36.1±31.6	**69.9±21.1**	13.9±25.1	**50.6±17.9**	11.1±7.4	**19.6±14.1**
10m	0.0±0.0	**3.9±4.9**	0.0±0.0	**41.1±6.2**	0.0±0.0	0.0±0.0	0.0±0.0	**0.6±0.9**
Source tasks								
5m6m	52.0±16.6	**68.9±9.9**	**68.2±2.8**	49.9±39.6	75.5±8.6	**82.3±3.6**	10.0±5.7	**27.0±6.9**
Unseen tasks								
7m8m	6.9±3.4	**23.6±9.9**	10.3±8.4	**14.9±10.7**	5.2±8.7	**13.5±22.3**	0.6±1.1	**12.4±21.1**
8m9m	1.8±2.7	**8.1±6.9**	0.6±0.8	**4.3±5.6**	0.3±0.6	**3.8±6.3**	0.5±1.1	**3.1±4.8**
10m11m	0.4±0.8	**2.3±3.4**	0.0±0.0	**1.4±2.4**	0.0±0.0	**5.6±12.6**	0.1±0.3	**0.4±0.6**
Source tasks								
2s3z	**94.0±2.8**	90.2±7.0	**94.2±3.3**	91.0±5.5	94.4±5.2	**96.8±3.1**	27.9±25.2	**79.5±8.2**
Unseen tasks								
1s4z	34.0±15.4	**44.2±10.7**	48.0±21.5	**52.7±24.1**	33.1±19.8	**82.8±16.8**	8.1±12.9	**38.3±10.0**
1s5z	13.6±12.7	**28.9±13.7**	13.9±13.4	**22.8±18.5**	15.4±14.9	**51.8±22.5**	2.9±5.7	**13.6±14.8**
2s4z	71.2±16.8	**79.1±14.9**	54.9±14.4	**68.4±7.3**	73.4±10.7	**91.3±2.9**	18.5±29.0	**64.8±15.9**
2s5z	51.9±29.0	**62.6±21.2**	14.5±12.5	**55.0±21.9**	30.0±23.0	**61.0±22.5**	12.1±19.4	**34.9±20.4**
3s3z	67.6±8.8	**87.4±10.0**	39.0±17.4	**71.1±22.7**	57.6±24.5	**80.6±22.9**	22.0±25.8	**52.6±23.3**
3s4z	54.5±24.1	**79.1±16.1**	13.7±12.0	**52.1±21.3**	33.5±21.3	**73.9±12.7**	20.6±25.3	**53.0±21.1**
3s5z	24.8±11.6	**63.5±28.3**	2.9±4.8	**35.5±11.2**	12.4±20.7	**45.8±22.5**	12.3±11.0	**28.6±22.3**
4s3z	26.1±13.0	**40.9±23.0**	8.4±6.1	**24.9±19.5**	12.1±16.2	**33.8±14.1**	8.3±11.5	**22.6±18.9**
4s4z	24.7±11.9	**40.6±17.1**	2.2±2.2	**21.3±25.1**	10.5±11.6	**32.9±13.0**	6.6±7.2	**31.6±19.9**
4s5z	12.0±12.5	**48.6±15.5**	1.6±2.7	**15.8±22.1**	9.5±16.8	**19.1±12.6**	3.5±4.3	**11.1±8.0**
4s6z	6.5±12.5	**41.6±24.3**	0.0±0.0	**7.1±8.2**	4.6±8.6	**15.6±13.4**	1.6±2.5	**2.5±1.8**

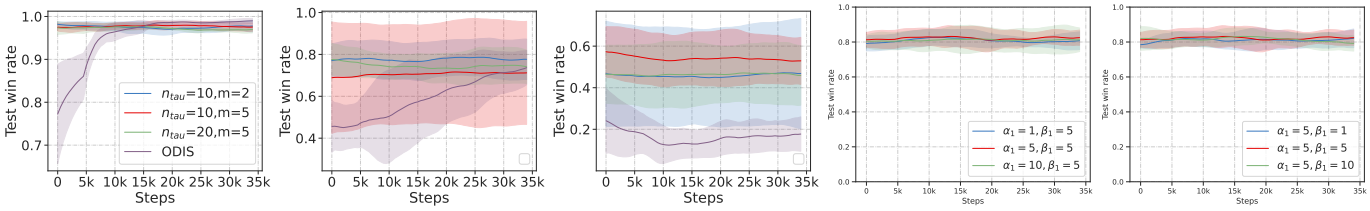


Fig. 6. The impact of option parameters. (a) 3m. (b) 4m. (c) 5m. (d) α_1. (e) β_1.

testing (i.e. learning from 3m expert data and then generalizing it to the 4m and 5m tasks), corresponding to $n_\tau = 10, m = 2$, $n_\tau = 10, m = 5$, and $n_\tau = 20, m = 5$. The results are shown in the Figure 6(a-c). We found that the performance of RIKD can maintain robust transitions on both the source task 3m and the unseen tasks 4m and 5m, and its performance is better than that of OIDS. Additionally, we conducted an ablation experiments on the weights of the loss function in the RIKD (study performance variations of the RIKD on unseen task 4m by adjusting loss weights), where loss $\mathcal{L}_\omega$ influences the accurate learning of risk knowledge and loss $\mathcal{L}_c$ affects the learning of mapping relationships. The results shown in Figure 6(d-e) indicate that the RIKD maintains favorable stability in performance when the weights α_1 (for loss $\mathcal{L}_\omega$) and β_1 (for loss $\mathcal{L}_c$) vary within a wide range of values.

VI. CONCLUSION

This paper proposes RIKD for learning generalizable risk response in multi-task. It is based on distributional RL and can effectively transfer the learning and responses of environmental risks in source tasks to unseen tasks through risk knowledge extraction modules and risk attitude selector. Since risk knowledge is more general, it is insensitive to offline data and can adapt to tasks for which it is difficult to collect offline data in the real world. Besides, we have provided an online version of RIKD, RIKD (online), which further validates the generality of our method. Finally, we tested the algorithm in multi-task under risk setting, and the results proved that RIKD achieved significant performance improvement.

RIKD achieves multi-task generalization by learning risk knowledge, but no further research has been conducted on the attributes of risks in the environment, as this may have a significant impact on the performance of the algorithm. This may be more pronounced when transferring tasks with asymmetric risks, although this paper focuses primarily on learning the mapping from risk knowledge to risk response. In future work, we will explore the differences in environmental risks in different scenarios and their impact on the selection of agent risk attitudes. For example, by clustering environmental risks in historical tasks, we can design an adaptive switching method for agent risk attitudes towards unknown tasks, further expanding the MARL's applicability.

APPENDIX
ENVIRONMENTAL SETTINGS

A. RIKD (online)

The policy network structure of RIKD (online) is shown in the Figure 7. RIKD (online) is implemented based on the centralized training decentralized execution architecture, where the policy network adds a risk knowledge extraction module and a risk attitude selector to the policy network of the UPDeT.

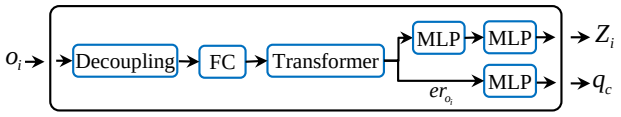


Fig. 7. The policy network structure of RIKD (online), where Z_i represents the distribution of agent policy value and q_c represents the option value.

B. Algorithm details

This section we will provide a detailed introduction to the network parameter settings and related training details of RIKD and RIKD (online). RIKD includes a risk knowledge extraction network, a risk attitude selector, and a mixing network. RIKD (online) consists of a policy network and a mixing network.

(1) Risk knowledge extraction network

Based on state information. When obtaining risk knowledge from the state of offline data, RIKD first decouples the state s into ally information s^{ally} and enemy information s^{enemy}, and then obtains an embedding with a dimension of 64 through a single-layer neural network. Afterward, the 64-dimensional embeddings of s^{ally} and s^{enemy} are cascaded as inputs for the self-attention mechanism. In the self-attention mechanism, we use a fully connected layer to represent query Q and key K, and map the input to an 8-dimensional embedding, respectively, and calculate the attention value.

Based on observational information. Due to the observation information having fewer features that can be obtained globally than state information, we supplemented it with (using 64-dimensional embedding representation) that reflects historical information h, and decoupling observation information is different from decoupling state information. RIKD decouples the observation information o of the agent into its own feature o^{own}, ally feature o^{ally}, and enemy feature o^{enemy}, and then maps them to embeddings with a dimension of 64 through fully connected layers, and cascades them together with history information h as input to the Transformer. The depth and head of the Transformer are both 1, and the embedding dimension of the hidden layer is 64.

(2) Risk attitude selector

The risk attitude selector generates risk attitudes with different risk preferences for each agent based on risk knowledge. The option network uses fully connected layers to map risk knowledge to embeddings with dimensions equal to the number of options, used to represent option values. In RIKD, we set the number of options to 5, indicating that the agent can choose from 5 risk attitudes. Then, we use the option value as input to Decode and learn the policy value distribution. In this paper, we set the number of quantiles $n_\tau = 10$.

TABLE XII
HYPERPARAMETER SETTINGS.

Hyper-parameter	Value
Hidden layer dimension	64
Learning rate	0.0005
optimizer	Adam
Double q	True
n_τ	10
Number of options	5
ω	8
Pretrain steps	15M(RIKD-online: 2M, train steps)
Coordination policy optimization	35M(RIKD-online: 0.5M, test steps)

(3) Mixing network

To meet the IGM criterion [41], we use a hypernetwork structure to encode state information into network parameters when weighting the value distribution of agent policy. Then we decompose the distribution of joint policy values based on the RIGM criterion [17]. The mixing network adopts a two-layer neural network to learn the value distribution of joint policy. RIKD approximates the joint policy value distribution in the mixing network to learn about environmental risks using an implicit quantile network function $F_z^{-1}(s,u|\omega) = f(\varphi(s)\odot\phi(\omega))_u$. ϕ first extends the scalar ω representing environmental risk to the n dimension through $\cos(\pi i\omega)_{i=0}^{n-1}$, and then uses a fully connected layer (64-64) with weights w_{ij} and b_j to obtain a quantile embedding $\phi(\omega) = \left[\phi(\omega)_j\right]_{j=0}^{\dim(\phi(\omega))-1}$, where $\phi(\omega)_j := \mathrm{Re}\,lu\left(\sum_{i=0}^{n-1}\cos(\pi i\omega)w_{ij}+b_j\right)$. In this paper, we set $n=64$, and set the dimensions of ω to 8. When optimizing the policy, we set $\alpha_1 = 5$ and $\beta_1 = 5$.

The hyperparameter settings for RIKD (online) and RIKD are the same on the relevant modules, as shown in the Table XII. But in the optimization of cooperation policy, $\alpha_2 = 3$, $\beta_2 = 1$. The experimental results of RIKD in this paper are all based on the average of five random seeds, and RIKD (online) results are all based on three random seeds. We use NVIDIA GeForce RTX 3060Ti GPU and 64 core CPU to run the code on the machine. Generally, the training process of RIKD takes 12-14 hours, while RIKD (online) takes 7-9 hours.

REFERENCES

[1] L. Zheng, J. Chen, J. Wang, J. He, Y. Hu, Y. Chen, C. Fan, Y. Gao, and C. Zhang, "Episodic multi-agent reinforcement learning with curiosity-driven exploration," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 3757–3769.

[2] H. Huang, Z. Hu, Z. Lu, and X. Wen, "Network-scale traffic signal control via multiagent reinforcement learning with deep spatiotemporal attentive network," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 262–274, 2023.

[3] Y. Zhang, M. Yue, J. Wang, and S. Yoo, "Multi-agent graph-attention deep reinforcement learning for post-contingency grid emergency voltage control," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 3, pp. 3340–3350, 2024.

[4] X. Yang, Y. Xu, L. Kuang, Z. Wang, H. Gao, and X. Wang, "An information fusion approach to intelligent traffic signal control using the joint methods of multiagent reinforcement learning and artificial intelligence of things," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9335–9345, 2022.

[5] G. Hu, Y. Zhu, D. Zhao, M. Zhao, and J. Hao, "Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 8, pp. 3966–3978, 2023.

[6] W. Li, X. Wang, B. Jin, D. Luo, and H. Zha, "Structured cooperative reinforcement learning with time-varying composite action space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8618–8634, 2022.

[7] X. Wang, H. Xu, Y. Zheng, and X. Zhan, "Offline multi-agent reinforcement learning with implicit global-to-local value regularization," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 52 413–52 429.

[8] E. Hannes, B. Debabrota, A. Mina, and D. Christos, "Risk-sensitive bayesian games for multi-agent reinforcement learning under policy uncertainty," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 2022, pp. 1–7.

[9] C. Li, S. Dong, S. Yang, Y. Hu, T. Ding, W. Li, and Y. Gao, "Multi-task multi-agent reinforcement learning with interaction and task representations," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 7, pp. 13 431–13 445, 2025.

[10] L. Meng, M. Wen, C. Le, y. Li, D. Peng, W. Zhang, Y. Wen, H. Zhang, J. Wang, Y. Yang, and B. Xu, "Offline pre-trained multi-agent decision transformer," *Mach. Intell. Res.*, vol. 20, pp. 233–248, 2023.

[11] S. Hu, F. Zhu, X. Chang, and X. Liang, "Updet: universal multi-agent reinforcement learning via policy decoupling with transformers," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[12] F. Zhang, C. Jia, Y.-C. Li, L. Yuan, Y. Yu, and Z. Zhang, "Discovering generalizable multi-agent coordination skills from multi-task offline data," in *Proc. Int. Conf. Learn. Represent.*, 2023.

[13] W.-C. Tseng, T.-H. J. Wang, Y.-C. Lin, and P. Isola, "Offline multi-agent reinforcement learning with knowledge distillation," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 226–237.

[14] Y. Liu, Y. Li, X. Xu, Y. Dou, and D. Liu, "Heterogeneous skill learning for multi-agent tasks," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 37 011–37 023.

[15] S. He, J. Shao, and X. Ji, "Skill discovery of coordination in multi-agent reinforcement learning," *CoRR*, vol. abs/2006.04021, 2020. [Online]. Available: https://arxiv.org/abs/2006.04021

[16] R. Wang, L. Zheng, W. Qiu, B. He, B. An, R. Zinovi, Y. Hu, Y. Chen, T. Lv, and C. Fan, "Towards skilled population curriculum for multi-agent reinforcement learning," 2023.

[17] S. Shen, C. Ma, C. Li, W. Liu, Y. Fu, S. Mei, X. Liu, and C. Wang, "Riskq: Risk-sensitive multi-agent reinforcement learning value factorization," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 34 791–34 825.

[18] K. Son, J. Kim, S. Ahn, R. D. D. Reyes, Y. Yi, and J. Shin, "Disentangling sources of risk for distributional multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, ser. Proceedings of Machine Learning Research, vol. 162, 17–23 Jul 2022, pp. 20 347–20 368.

[19] S. Mikayel, R. Tabish, S. d. W. Christian, F. Gregory, N. Nantas, G. R. Tim, H. Chia-Man, H. T. Philip, F. Jakob, and W. Shimon, "The starcraft multi-agent challenge." in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, ser. AAMAS'19, 2019, pp. 2186–2188.

[20] S. Zhang and H. Yao, "Quota: The quantile option architecture for reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, ser. AAAI'19, vol. 33, 2019.

[21] B. Mavrin, H. Yao, L. Kong, K. Wu, and Y. Yu, "Distributional reinforcement learning for efficient exploration," in *Proc. Int. Conf. Mach. Learn.*, ser. ICML'19, vol. 97, 09–15 Jun 2019, pp. 4424–4434.

[22] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, ser. ICML'18, vol. 80, 10–15 Jul 2018, pp. 1096–1105.

[23] A. A. Taiga, R. Agarwal, J. Farebrother, A. Courville, and M. G. Bellemare, "Investigating multi-task pretraining and generalization in reinforcement learning," in *The Proc. Int. Conf. Learn. Represent.*, 2023.

[24] Z. Zhao, Y. Fu, J. Chai, Y. Zhu, and D. Zhao, "Meta learning task representation in multiagent reinforcement learning: From global inference to local inference," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 8, pp. 14 908–14 921, 2025.

[25] Z. Hu, Z. Zhao, X. Yi, T. Yao, L. Hong, Y. Sun, and E. Chi, "Improving multi-task generalization via regularizing spurious correlation," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 11 450–11 466.

[26] Q. Long, Z. Zhou, G. Abhibav, F. Fang, Y. Wu, and X. Wang, "Evolutionary population curriculum for scaling multi-agent reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[27] J. Chen, Y. Zhang, Y. Xu, H. Ma, H. Yang, J. Song, Y. Wang, and Y. Wu, "Variational automatic curriculum learning for sparse-reward cooperative multi-agent problems," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9681–9693.

[28] H. Huang, D. Ye, L. Shen, and W. Liu, "Curriculum-based asymmetric multi-task reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7258–7269, 2023.

[29] W. Wang, T. Yang, Y. Liu, J. Hao, X. Hao, Y. Hu, Y. Chen, C. Fan, and Y. Gao, "From few to more: Large-scale dynamic multiagent curriculum learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, pp. 7293–7300, Apr. 2020.

[30] J. Yang, I. Borovikov, and H. Zha, "Hierarchical cooperative multi-agent reinforcement learning with skill discovery," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, ser. AAMAS'20, Richland, SC, 2020, pp. 1566–1574.

[31] S. Liu, Y. Shu, C. Guo, and B. Yang, "Learning generalizable skills from offline multi-task data for multi-agent cooperation," in *International Conference on Learning Representations*, 2025.

[32] Y. Zhang, M. Chadli, and Z. Xiang, "Prescribed-time formation control for a class of multiagent systems via fuzzy reinforcement learning," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 12, pp. 4195–4204, 2023.

[33] J. Li, H. Shi, and K.-S. Hwang, "Using fuzzy logic to learn abstract policies in large-scale multiagent reinforcement learning," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 12, pp. 5211–5224, 2022.

[34] Q. Fu, Z. Pu, Y. Pan, T. Qiu, and J. Yi, "Fuzzy feedback multiagent reinforcement learning for adversarial dynamic multiteam competitions," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 5, pp. 2811–2824, 2024.

[35] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proc. AAAI Conf. Artif. Intell.*, ser. AAAI'18, 2018.

[36] M. Rowland, R. Dadashi, S. Kumar, R. Munos, M. G. Bellemare, and W. Dabney, "Statistics and samples in distributional reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, ser. ICML'19, vol. 97, 09–15 Jun 2019, pp. 5528–5536.

[37] G. An, S. Moon, J.-H. Kim, and H. O. Song, "Uncertainty-based offline reinforcement learning with diversified q-ensemble," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.

[38] O. Frans, A and A. Christopher, "A concise introduction to decentralized pomdps," 2016.

[39] M. G. Bellemare, N. L. Roux, P. S. Castro, and S. Moitra, "Distributional reinforcement learning with linear function approximation," in *Proc. Int. Conf. Artif. Intell. Stat.*, ser. Proceedings of Machine Learning Research, vol. 89, 16–18 Apr 2019, pp. 2203–2211.

[40] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proc. AAAI Conf. Artif. Intell.*, ser. AAAI'17, 2017, pp. 1726–1734.

[41] 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.*, ser. ICML'18, vol. 80, 10–15 Jul 2018, pp. 4295–4304.

[42] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Adv. Neural Inf. Process. Syst.*, ser. NIPS'17, 2017, pp. 6379–6390.