TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 23/31 pp318–330
DOI: 10.26599/TST.2024.9010013
Volume 30, Number 1, February 2025

Optimizing Risk-Aware Task Migration Algorithm Among Multiplex UAV Groups Through Hybrid Attention Multi-Agent Reinforcement Learning

Yuanshuang Jiang†, Kai Di†, Ruiyi Qian, Xingyu Wu, Fulin Chen, Pan Li, Xiping Fu, and Yichuan Jiang*

Abstract: Recently, with the increasing complexity of multiplex Unmanned Aerial Vehicles (multi-UAVs) collaboration in dynamic task environments, multi-UAVs systems have shown new characteristics of inter-coupling among multiplex groups and intra-correlation within groups. However, previous studies often overlooked the structural impact of dynamic risks on agents among multiplex UAV groups, which is a critical issue for modern multi-UAVs communication to address. To address this problem, we integrate the influence of dynamic risks on agents among multiplex UAV group structures into a multi-UAVs task migration problem and formulate it as a partially observable Markov game. We then propose a Hybrid Attention Multi-agent Reinforcement Learning (HAMRL) algorithm, which uses attention structures to learn the dynamic characteristics of the task environment, and it integrates hybrid attention mechanisms to establish efficient intra- and inter-group communication aggregation for information extraction and group collaboration. Experimental results show that in this comprehensive and challenging model, our algorithm significantly outperforms state-of-the-art algorithms in terms of convergence speed and algorithm performance due to the rational design of communication mechanisms.

Key words: Unmanned Aerial Vehicle (UAV); multiplex UAV group structures; task migration; multi-agent reinforcement learning

1 Introduction

With the application of artificial intelligence technology in the field of multi-agent systems[1–4], Unmanned Aerial Vehicles (UAVs) have found widespread application in both civilian and military domains. The number and complexity of tasks performed by UAVs continue to increase[5]. For example, in the maritime domain, UAVs have taken on multiple tasks such as waterway patrol, law enforcement inspection, accident investigation, and

• Yuanshuang Jiang, Kai Di, Ruiyi Qian, and Yichuan Jiang are with School of Computer Science and Engineering, Southeast University, Nanjing 211189. E-mail: yuanshuangjiang@seu.edu.cn; dikai@seu.edu.cn; qianruiyi@seu.edu.cn; yjiang@seu.edu.cn.
• Xingyu Wu is with School of Software Engineering, Southeast University, Nanjing 211189. E-mail: wuxingyu@seu.edu.cn.
• Fulin Chen and Pan Li are with School of Cyber Science and Engineering, Southeast University, Nanjing 211189. E-mail: chenfulin@seu.edu.cn; lipan@seu.edu.cn.
• Xiping Fu is with PredictHQ, Auckland 1010, New Zealand. E-mail: xiping@predicthq.com.
• A preliminary version of this paper was presented at the 24th International Conference on Parallel and Distributed Computing, Applications and Technologies.
† Yuanshuang Jiang and Kai Di contribute equally to this work.
* To whom correspondence should be addressed.
Manuscript received: 2023-11-06; revised: 2023-12-25; accepted: 2024-01-04

emergency rescue[6–9]. Meanwhile, due to the dynamics of practical task environments, the completion of these complex tasks places higher demands on the collaborative capabilities of UAVs[10], UAVs need to learn to dynamically reallocate tasks through task migration in dynamic environments. In this context, UAV systems are beginning to show new trends of inter-coupling among multiple groups and intra-correlation within groups[11]. Compared with previous researches on UAV task migration problem, these multiplex UAV group task migration problems in dynamic environments face greater challenges.

However, previous studies mainly focus on the impact of task environment risks on individual UAV[11, 12], without fully considering the structural impact of task environment risks on UAVs under the multiplex UAV group structure. This can lead to algorithms falling into local optima. Therefore, there are two main challenges in designing algorithms for multiplex UAV group task migration under the dynamic environmental risks:

- The action space of agents is influenced by dynamic risks in the task environment, requiring dynamic adjustments of task strategies based on the risk. This leads to a significant increase in problem complexity.
- The task migration strategy of agents is influenced by structural constraints inter- and intra-groups, thus expanding the solution space of the problem.

To address these challenges, this paper propose the Hybrid Attention Multi-agent Reinforcement Learning (HAMRL) algorithm. First, unlike conventional approaches that rely on building accurate models, this framework exploits the powerful generalization capabilities of reinforcement learning in dynamic environments[13] to dynamically modify strategies and adapt to new environmental states. Second, it uses attention mechanisms to process time series data and learn about the dynamic nature of the task environment. Final, this algorithm relies on the hybrid attention communication mechanism to learn the structural influence of this group structure. Specifically, the leader agent within each group first learns the influence relationship among agents within the group through the intra-group attention network, analyzes their strategy association under the dynamic risk, and obtains the dynamic communication information of each group. Then, each agent synthesizes the communication information of multiple groups through the inter-group attention mechanism, learns the dynamic interaction among groups to obtain the task migration strategy, and guides the agent to realize efficient task migration and collaboration.

Overall, the main contributions of this work can be summarized as follows:

- To the best of our knowledge, this is the first study that integrates the structural impact of task environment risks on UAVs under the multiplex UAV group structure. This research constructs a risk-aware task migration model in multiplex UAV groups, extending previous research on UAV task migration in dynamic environments.
- We describe the problem as a Partially Observable Markov Game (POMG) and present a HAMRL algorithm. This algorithm effectively addresses the structural effects of multi-group structures in dynamic environmental risk settings.
- We empirically evaluate our algorithm in a simulated environment, and the experimental results demonstrate that our algorithm significantly outperforms state-of-the-art algorithms in terms of convergence speed and algorithm performance.

The remaining sections of this paper are organized as follows: Section 2 discusses related work, Section 3 outlines the risk-aware task migration model in multiplex UAV groups, Section 4 introduces the multi-agent reinforcement learning framework with a hybrid attention communication mechanism, Section 5 presents experimental results and analysis, and Section 6 concludes the paper and provides insights into future research directions.

2 Related Work

In recent years, task migration research has been widely studied in the field of UAVs. For example, Hua et al.[14] proposed a two-layer iterative algorithm utilizing convex optimization techniques. This approach alternately optimizes task migration and path planning for UAV swarms, with the primary objective of tackling the issue of an excessive number of UAV tasks. Wang et al.[15] proposed an online UAV swarm task migration method, this method combines tasks according to location and then migrates tasks to improve task utilization.

These researches usually ignored the effect of environment dynamics on task migration. However, in real environments, task environments are often dynamic, and some studies have proposed a many of

works considering these dynamics. For example, Sacco et al.[16] proposed a distributed adaptive task migration algorithm to dynamically predict the length of the UAV's future tasks and perform task migration to avoid overloading the UAV due to risk fluctuations in dynamic environments. Liu et al.[17] proposed Markov approximation algorithms to construct the near-optimal solution of the mathematical model, and then use Lyapunov algorithms to deal with the dynamics of the environment to optimize the long-term benefits, which reduces the migration cost between UAV swarms and end-device groups in dynamic environments.

These studies are usually model-based traditional optimization algorithms, which require a large amount of a priori knowledge to construct an accurate mathematical model, a situation that leads to poor generalization ability of the model and makes it difficult to adapt to the new states that appear in dynamic environments[18]. To cope with the UAV group task migration problem in this dynamic task environment, some researches have proposed task migration methods based on multi-agent reinforcement learning, which uses deep neural networks to end-to-end the interaction relationship of multi-agent within the UAV group, and adaptively learn to update the strategy by interacting with the environment to improve the generalization ability of the model. For example, in order to cope with the effect of environmental dynamics on the stability of multi-UAV group, a clustering method based on reinforcement learning is proposed[11], which improves the stability of multi-UAV swarm formation by reorganizing the members of the swarms by adaptively adjusting the clustering parameters according to the external environment. Gao et al.[19] proposed a method to solve the computational problem caused by the large number of UAV groups in dynamic environments by using an attention mechanism.

However, all these reinforcement learning-based studies are usually under a single group or consider the structure of multiple group, but do not take into account the structural impact of the risk of the task environment on UAVs under a multiplex UAV group structures, thus leading these algorithms to fall into localized solutions. To solve this problem, we propose a multi-agent reinforcement learning method based on the hybrid attention communication mechanism to learn this structural influence.

3 Model and Problem Definition

Without sacrificing generality, we consider area $\mathcal{B}$ that is continuously destroyed by natural disasters, and some corresponding number of rescue tasks $K = \{k_1, k_2, \ldots, k_W\}$. The UAVs are denoted by $D = \{d_1, d_2, \ldots, d_N\}$, and the UAVs are divided into multiple groups according to their types, the task migration costs of UAVs are different in and out of groups. Therefore, UAVs can only obtain observation data from the same group of UAVs, but not from other groups. These UAVs must be attentive to the evolving risk cost v_t which may vary over time (for example, persistent aftershocks following an earthquake).

At the beginning, W tasks will be allocated to multiple UAV groups. Some UAVs may be overloaded because they are assigned multiple tasks. At the same time, if UAV perform a task without considering the risk cost, the risk cost may also rise. Excessive task workload and high risk costs can lead to an increase in the overall cost of task completion. As a result, UAVs must learn to dynamically migrate tasks to other agents within or outside the group and determine the appropriate time to execute tasks in order to decrease the probability of UAV damage and task completion cost.

In the following sections, we will detail the risk-aware task migration problem for multiple UAV groups.

3.1 Preliminaries

Risk-aware task migration problem in multiplex UAV groups has two constraints. First, UAV agents can only observe the information of the current UAV group, and the information of other UAV groups is unobserved. Second, in order to minimize the cost of all UAVs, agents are required to adaptively migrate task within and among multiple UAV groups while considering the risk cost.

3.1.1 Multiplex UAV groups

For the first constraint, we describe the multiplex UAV groups structure.

Multiplex UAV groups structure. The structure of multiplex UAV groups involves a configuration where n agents are connected by m different types of edges. Agents connected by the same type of edge together form a group. Multiple groups are represented by the vector $\mathcal{G} = \{g^1, g^2, \ldots, g^m\}$, whose elements are the UAV subscript set of the m-th group. For example

$\{i, j\} \in g^m$ indicates that when $c_{ij}^m = 1$, agent i and agent j are connected in g^m by type m. The observational limits for each agent $\{1, 2, \ldots, j\}$ in g^m can be defined as follows:

$$o^m = \{o_j | \forall j \in g^m\} \tag{1}$$

where o^m denotes view information of any agent in g^m, and o_j denotes local view information of agent j.

3.1.2 Task migration in multiplex UAV groups

For the second constraint, this paper builds a risky task migration model in multiplex UAV groups:

Task. Generally, in a UAV system, each task i has two pieces of important information: task workload and task allocation.

- Task workload h_t^k is the size of task k at time t, and h_t^k is variable. Each agent d_i has a current task workload queue E_i^t, where the elements represent the subscripts of the tasks in the queue. The workload size of the task queue H_i^t can be formalized as

$$H_i^t = \sum_{k \in E_i^t} h_t^k \tag{2}$$

- Task assignment variable P_{kj} denotes the assignment of tasks k to agent d_j. In cases where an agent is assigned multiple tasks, the task with the lower workload is given priority for execution or migration. The task execution speed of UAV agent d_i is denoted by s_i.

Risk. There are two kinds of risk in this paper, environment and task workload.

- For the UAV data collection environment, the risk cost of collecting data in area $\mathcal{B}$ is dynamic due to damage from natural disasters. The UAV must pay attention to the risk cost v_t (the risk cost is variable, the value is a non-monotonic function ranging from 0 to 1) in order to reduce the task cost. In other words, UAVs must stop collecting data at higher risk cost and start collecting data at lower risk cost.
- For task workload, if a significant number of tasks are assigned to agent d_i, the resulting workload becomes excessive. In such a case, the UAV may have to spend additional time exploring the area $\mathcal{B}$, resulting in an escalation of the overall task cost.

Task migration in multiplex groups. Due to the existence of risk, tasks with overloaded agents may need to be migrated to less overloaded agents to reduce the task cost. However, due to this multiplex group structure, agents may need to perform task migration in multiplex group, this migration cost is also different. For migration cost l^m in group g^m, cost value l^m is correlated with distance cost between agent d_i and agent d_j in group g^m. For migration costs l_{ij} in different groups, the cost value l_{ij} is related to the distance cost between the group of agent d_i and the group of agent d_j.

3.2 Problem formulation

The core objective of this paper is how to maximize the reward of all UAVs in the multiplex UAV group. This reward is mainly related to task completion cost and task completion reward.

- Task completion cost is the cost of completing all data collection tasks, all UAVs are required to complete data collection tasks as far as possible in time steps $\{1, 2, \ldots, T\}$ at the lowest cost.
- Task completion reward is gained by completing all tasks within T time steps, the higher reward mean higher task completion rate.

Task completion cost $\mathcal{T}_{ij}^t$ of a task k migrating from agent d_i to agent d_j and the task execution cost $\mathcal{E}_i^t$ can be formalized as

$$\mathcal{T}_{ij}^t = l_{ij} \cdot x_{ij}^t \cdot u_t \tag{3}$$

$$\mathcal{E}_i^t = y_i^t \cdot s_i \cdot v_t \cdot u_t \tag{4}$$

$$x_{ij}^t + y_i^t \leqslant 1 \tag{5}$$

$$x_{ij}^t \in \{0,1\}, \quad y_i^t \in \{0,1\}, \quad \forall i, j \in D, \quad t \in T \tag{6}$$

where u_t denotes whether the task was executed at time t, $x_{ij}^t = 1$ denotes the migration of task k from agent i to agent j, and otherwise 0. $y_i^t = 1$ denotes the execution of task k by agent i at time t, otherwise 0. Task completion cost $\mathcal{T}$ of completing all tasks within T time steps can be formalized as

$$\mathcal{T} = \sum_{t \in T} \mathcal{T}_t = \sum_{t \in T} \sum_{i \in D} \left(\sum_{j \in D} \mathcal{T}_{ij}^t + \mathcal{E}_i^t \right) \tag{7}$$

Task completion reward is related to all agents' task completion reward. A failed task migration can lead to a decrease in the task completion reward, and the agent either chooses to migrate the task or chooses to execute the task, therefore n UAVs task completion reward $\mathcal{R}$ can be formalized as

$$\mathcal{R} = \eta \cdot \sum_{t \in T} \mathcal{R}_t = \eta \cdot \sum_{t \in T} \left(\sum_{i \in D} (y_i^t \cdot s_i) \right) \tag{8}$$

where η is reward weight.

This form of reward enables to meet different demands by adjusting the weights, for example, for time-sensitive systems such as rapid disaster response set-up, a larger η is recommended to meet the strict requirements.

In this section, the optimization objective can be formalized. To minimize UAV costs by migrating tasks in multiplex UAV groups. The objective function $\mathcal{L}$ is defined as

$$\mathcal{L} = \max\ (-\mathcal{T} + \mathcal{R}) \tag{9}$$

Based on the above description, we propose the definition of the multiplex UAV groups risk-aware task migration problem: Within T time steps, given the multiplex UAV groups $\mathcal{G}$ containing agents and tasks, the goal of the multiplex UAV groups risk-aware task migration problem is to find an optimal task migration and execution strategy with minimum cost.

4 Multi-Agent Reinforcement Learning Methodology

In the previous section, the multiplex UAV groups risk-aware task migration problem is reduced to a maximization optimization problem. The task migration and task assignment problem in Eq. (10) can be shown to be NP-hard[20]. To address this problem, this section proposes a multi-group risk-aware task migration approach based on multi-agent reinforcement learning.

4.1 Reinforcement learning settings

To solve this multiple UAV group risk-aware task migration problem using multi-agent reinforcement learning, we describe it as a POMG problem. The POMG model can be defined as a tuple (S, A, R). We assume that each UAV d_i as an agent must decide on its task migration object and task execution time based on the state S in order to maximize the discounted expected reward R. The immediate reward R_i^t of the reinforcement learning system is calculated by the task execution cost and the task execution reward at time t. This section provides the formal definitions of each agent (S, A, R).

State space *S*. The reinforcement learning system consists of UAVs and tasks. Therefore, the system state is a description of all task and UAV states that includes task information, agent information, and risk cost with time variation. An agent state at time t consists of these 8 dimensions.

(1) Task workload queue E_i^t (Task information);

(2) Task workload queue size H_i^t (Task information);

(3) Task execution speed s_i (Task information);

(4) Migration cost in the group g^m (Agent information);

(5) Migration cost in different group (Agent information);

(6) Agent group information g^m (Agent information);

(7) State information of other agent in the group g^m. To satisfy first constraint, only the information in the g^m group is known, the information in the other groups is unknown (Agent information);

(8) Risks cost v^t (Time-varying information).

Action *A*. All agents have the same amount of available action space. The action space of agent d_i can be represented as follows:

$$a_i = \begin{cases} \text{Agent standby}, & \text{if } a_i = 0; \\ \text{Task execution}, & \text{if } a_i = d_i; \\ \text{Task migration to } d_j, & \text{otherwise} \end{cases} \tag{10}$$

where $a_i \in [0, N]$ and if the value of a_i is 0, agent d_i remains on standby. If the value of a_i is i, agent d_i executes the task. In all other cases, specifically when a_i equals d_j, agent d_i migrates the task to agent d_j.

Reward *R*. Each UAVs in the multiplex UAV groups migration problem has the same objective, which is to maximize the cumulative reward. According to Eq. (9), the rewards for all UAVs at time t are as follows:

$$R_t = -\mathcal{T}_t + \mathcal{R}_t \tag{11}$$

4.2 Hybrid attention multi-agent reinforcement learning

In order to solve above POMG problems, we propose a multi-agent reinforcement learning algorithm based on the hybrid attention communication mechanism, as shown in Fig. 1. It is a method based on the actor-critic architecture, which learns the structural influences of dynamic features mainly through the critic. First, during the critic training process, each agent must first listen to message, while m agents are elected as leader agent in leader election period. Second, each leader agent then use intra-group hybrid attention mechanism to encode intra-group information as broadcast information $\hat{\mathcal{M}}_i$ to send to all agents during convergence of intra-group communication information period. Agents then use inter-group hybrid attention mechanism to decode all $\check{\mathcal{M}}_i$ to help generate Q_i value in convergence of inter-group communication

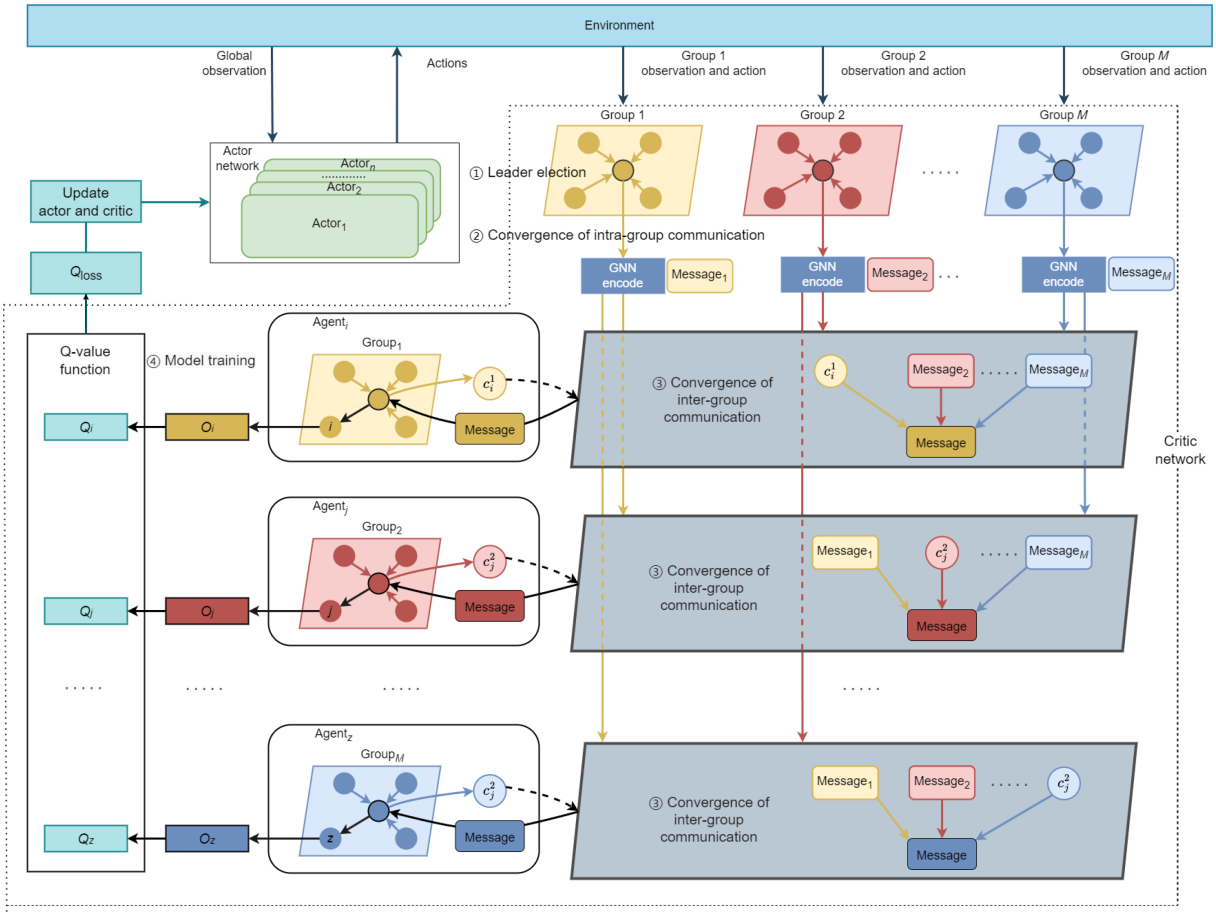


Fig. 1 System architecture diagram.

information period. Final, the algorithm updates the actor and critic networks by a certain batch, which is the fourth stage of model training. The detailed procedure of each step is given below.

Leader election. During the initialization period, the algorithm first needs to determine the leader agent of each group $g^m \in \mathcal{G}$. Specifically, the agent group is first divided into N groups according to the prior knowledge, and then the leader agent w^m of each group is determined according to the task processing ability s_i of the agent in each group. This can be formalized as follows:

$$w^m = \arg\max_{i \in g^m} s_i \tag{12}$$

Convergence of intra-group communication information. After each group $g^m \in \mathcal{G}$ elects a leader w^m, the leader w^m must collect information o^m about all agents in g^m. However, due to the limited communication bandwidth and computational problems caused by the large number of UAVs, the communication information needs to be further processed and encoded. Here, we first encode the multi-dimensional communication information o_j of each agent d_j in the group into a lower-dimensional feature embedding e_j,

$$e_j = f_j(o_j) \tag{13}$$

where f_j is a single-layer MLP embedding function.

After completing the feature embedding encoding, we aggregate the feature embedding e_j of each agent in the group into the inter-group communication information $\hat{\mathcal{M}}_m$ through the hybrid attention mechanism,

$$\hat{\mathcal{M}}_m = \sum_{i \in g^m} \alpha_{w^m j} V^m e_j \tag{14}$$

where V^m is the weight matrix shared by all agents in group g^m, and $\alpha_{w^m j}$ is the attention weight from leader w^m to agent d_j. The attention weight $\alpha_{w^m j}$ is obtained by computing the similarity score between the agent feature embedding e_j and the leader feature embedding e_{w^m} and subjecting the score to a softmax operation. $\alpha_{w^m j}$ can be described formally as follows:

$$\alpha_{w^m j} \propto \exp(e_{w^m}, e_j) \tag{15}$$

In this way, we effectively aggregated the data within each group and were able to reduce the frequency of information transfer and increase the efficiency of information use through graphical attention.

4.2.1 Convergence of inter-group communication information

After the inter-group communication information of each group has been broadcast and sent to other groups. The agent d_i in group g^m will eventually receive a broadcast communication information message from the leader w^m. When the algorithm evaluates the strategy of agent d_i, the set of communication information received by w^m, $\mathcal{M}_m$, can be represented as

$$\mathcal{M}_m = \left\{o_j, \hat{\mathcal{M}}_k | \forall j \in g^m, \forall k \in \{1, 2, \ldots, M\} \wedge k \neq m\right\} \tag{16}$$

For agent d_j, this algorithm uses the inter-group hybrid attention network to model the relationship between d_j and the communication information message. Since the message contains not only the inter-group communication information $\hat{\mathcal{M}}_k$, but also the observation information of other agents in g^m, in order to avoid the confusion of information in different dimensions, which affects the final decision effect, the algorithm first uses the hybrid attention network to extract the communication information c_i^m of this group, which can be described as follows:

$$c_i^m = \sum_{j \in g^m} \alpha_{ij} V^m e_j \tag{17}$$

where $e_i = f_i(o_i)$, V^m is the weight matrix shared by all agents in group g^k, and $j \neq i$, α_{ij} is the attention weight of agent i to other agent j in group g^m. α_{ij} can be obtained in the same way as in Formula (15).

To compute the inter-group communication of $\mathcal{M}_m$, the algorithm again uses hybrid attention nets. The computation process is as follows:

$$\check{\mathcal{M}}_i = \alpha_{im} V^{m'} c_i^m + \sum_{k \in [1,M]} \alpha_{ik} V^k \hat{\mathcal{M}}_k \tag{18}$$

where V^k and $V^{m\prime}$ represent the weight matrix, α_{ik} is the attention weight of agent d_i to the other group message, and α_{im} is the attention weight of agent d_i to the current group g^k.

After the above processing, the final information collected by the agent d_i is the observation-action pair of d_i with the communication information $\check{\mathcal{M}}_i$ from w^m, which can be formalized as $O_i = \{o_i, a_i, \check{\mathcal{M}}_i\}$.

Model training. In the final step, the actor-critic structure is used to train each agent's actor and critic networks separately. For the critic network, this loss function trains the critic network by minimizing the difference between the critic network's estimate of the action value and the target value. This loss function can be calculated using the following formula:

$$\mathcal{L}(\psi_i) = \frac{1}{S} \sum_j (Q_i^\mu(o_i^j, a_i^j, \check{\mathcal{M}}_i^j)|_{a_i = \mu_i(o_i)} - y^j)^2 \tag{19}$$

where S is the number of samples, j is the index of a sample within a batch, μ_i is the policy network, which refers to the way the agent d_i chooses actions based on its current state. $y^j = r_i^j + \gamma Q_i^{\mu'}(o_i^{j\prime}, a_i^{j\prime}, \check{\mathcal{M}}_i^{j\prime})|_{a_i' = \mu_i'(o_i)}$ is the target, computed by the target function Q, r_i^j is the immediate reward received by the agent d_i in response to the action taken in sample j, γ is the discount factor, o_i' is the next state observed by agent d_i, a_i' is the next action taken by agent d_i in the next state o_i', $\check{\mathcal{M}}_i^{j\prime}$ represents the next external information, and $Q_i^{\mu'}\left(o_i', a_i', \check{\mathcal{M}}_i^{j\prime}\right)$ represents s the maximum action value expected by agent d_i, respectively.

The actor network is updated using the policy gradient method. Its policy gradient formula is as follows:

$$\nabla_{\theta_i} J \approx \frac{1}{S} \sum_j \nabla_{\theta_i} \mu_i(o_i^j) \nabla_{a_i} Q_i^\mu(o_i^j, a_i^j, \check{\mathcal{M}}_i^j) \tag{20}$$

where θ_i is a parameter of the actor network, S is the number of samples, j is the index of a sample within a batch, μ_i is the policy, which refers to the way the agent d_i chooses actions based on its current state, $\nabla_{\theta_i}\mu_i\left(o_i^j\right)$ represents the gradient of the policy function μ_i with respect to the parameter $\theta_i, \nabla_{a_i} Q_i^\mu$ represents the gradient of the Q function Q_i^μ with respect to the action a_i. The formula updates the actor network by calculating the probability gradient of the action chosen by the strategy μ_i in the sample data. Finally, Hybrid Attention Multi-agent Reinforcement Learning (HAMRL) algorithm can be derived in Algorithm 1.

5 Experiment

5.1 Experimental settings

Experiments were conducted on a personal computer equipped with an NVIDIA RTX GeForce RTX 3090 GPU, Ubuntu 20.04 LTS operating system, 2.10 GHz Xeon(R) Silver 4216 CPU, 64 GB RAM. Each

Algorithm 1 Hybrid Attention Multi-Agent Reinforcement Learning

Initialize replay buffer $\mathcal{D}$
Initialize the UAV into m groups according to Eq. (12)
for episode = 1 to max – iters **do**
  Reset environments, and get initial o_i for each agent i
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial state $\boldsymbol{x} = \{o_1, o_2, ..., o_N\}$
  for t = 1 to max – episode – length **do**
    for each agent i, select action $a_i = \boldsymbol{\mu}_{\theta_i}(o_i) + \mathcal{N}_t$ w.r.t. the current policy and exploration
    Execute task migration actions $a = (a_1, a_2, ..., a_N)$ and observe reward r and new state $\boldsymbol{x}' = \{o_1, o_2, ..., o_N\}$
    Store $(x, a, r, \boldsymbol{x}')$ in replay buffer $\mathcal{D}$
    set $\boldsymbol{x} \leftarrow \boldsymbol{x}'$
    Each agent sample j-th minibatch sample (o_i, r, o_i') from $\mathcal{D}$
    for each leader agent i = 1 to M **do**
      Compute intra-group information $\hat{\mathcal{M}}_i$ by Eq. (14) and send to other agent
    for agent i = 1 to N **do**
      Compute inter-group communication information $\check{\mathcal{M}}_i$ by Eq. (18)
      Update critic by minimizing the loss by Eq. (19)
      Update actor using the sampled policy gradient by Eq. (20)
    Update target network parameters for each agent i: $\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$

experiment was iterated 10 000 times to obtain average reward results. The experimental algorithms were developed using Python 3.7 with PyTorch 2.0 as the deep learning framework. The experiments were conducted in a dynamic risk environment constructed based on the MPE architecture[21]. To emulate the uncertain dynamic risk environment, the wave height dataset[22] from Kaggle was used. The initial environment consisted of 50 agents organized into 10 groups. The skill of each agent was randomly assigned, with values ranging from 1 to 9, with an average of 5. The size of each task was also randomly assigned, ranging from 5 to 20, with an average of 12.5[23]. In each experiment, the initial number of items is set to 400.

Our proposed algorithm uses a hybrid attention actor-critic network architecture, as shown in Fig. 1 (system architecture diagram), which consists of attention modules. The learning rate of the network is set to 0.001, and the hidden layer dimension is 128. The reward discount factor is set to 0.99, the replay buffer length for training is 1 000 000, the data size for each training batch is 1024, the number of environments running in parallel is 20, and the maximum number of training steps is set to 10 000.

5.2 Experimental metrics and comparative algorithms

The following performance metrics are used in the experimental evaluation:

(1) Task completion rate: This metric quantifies the total reward the system receives when it executes the task migration strategy provided by the algorithm and successfully completes the tasks.

(2) Task completion cost: Task completion cost consists of task migration cost and task execution cost: (a) Task migration cost is further divided into intra-group migration (within the same group of agents) and inter-group migration (across different groups of agents), with inter-group migration incurring a higher cost. (b) Task execution cost is influenced by task size and real-time risk factors.

This experiment provides three other algorithms for comparison:

(1) Nearest Neighbour Task Migration algorithm (NNTM)[24]: The NNTM algorithm strategically selects the nearest available neighbor for task migration, ensuring that tasks are efficiently assigned to the closest resources in the system.

(2) Multi-Actor-Attention-Critic (MAAC)[25]: MAAC is a variant of the classical actor-critic algorithm designed for applications in multi-agent reinforcement learning. Its main innovation lies in the incorporation of an attention mechanism within the critic, which allows the dynamic selection of relevant information for each intelligent agent. This approach mitigates the potential problem of dimensionality explosion that can occur with an increasing number of agents, thereby increasing the effectiveness of the algorithm.

(3) Hierarchical graph Attention based Multi-agent Actor-critic (HAMA) approach[26]: HAMA extends the traditional multi-agent actor-critic model by incorporating hierarchical graph attention mechanisms. This strategic allows agents to selectively focus on critical information within their groups, facilitating more informed decision making.

5.3 Experimental evaluation

This section first compares the HAMRL approach with

other algorithms to analyze its convergence in the simulated environment. The performance of HAMRL is then evaluated by performing ablation tests, varying the number of tasks and groups of agents.

5.3.1 System average reward

Parameter settings: In the experiment, the environment was set up with 50 agents divided into 10 groups, and a total of 400 tasks are assigned. To evaluate the convergence of the HAMRL model in terms of average reward, we performed 10 000 training iterations and simultaneously compared it with heuristic, MAAC, and HAMA methods.

Experimental phenomena: Figure 2 shows the average system reward achieved by HAMRL in the simulation environment. The results show that the average rewards of all task migration approaches, except the heuristic-based approach, increase with the number of training iterations until they reach convergence. In particular, HAMRL shows the fastest convergence and achieves the highest average system reward.

Results analysis: The HAMRL approach demonstrates rapid convergence and excels in optimizing system average reward metrics. This performance is due to the hybrid attention network model used in the approach, which is specifically designed to facilitate multi-agent group communication in dynamic risk environments. This approach allows the model to acquire global information through a limited number of inter-group communications, enabling adaptive task migration to mitigate high risk costs. The increased communication efficiency accelerates training convergence, while precise global feature extraction optimizes the task migration strategy, ultimately leading to superior system average rewards.

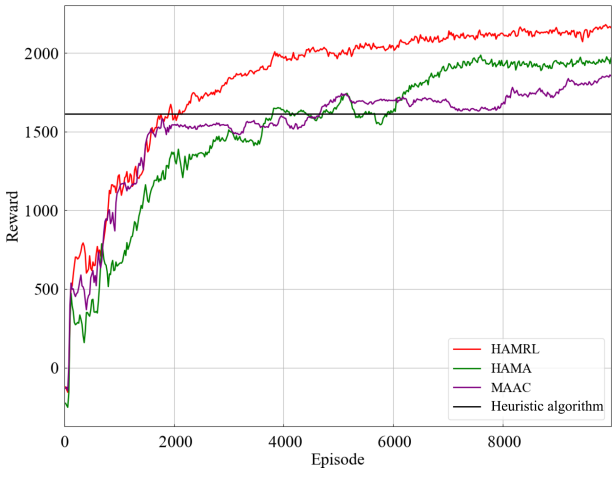


Fig. 2 Cumulative rewards during the learning process.

In comparison, the HAMA-based approach neglects the impact of dynamic risk environments on agent communication and collaboration. The MAAC-based approach performs even worse because it ignores the group composition of multiple agents, leading to suboptimal results. While the heuristic-based approach may produce a quick result in a single run, its lack of adaptability due to predetermined rules and prior knowledge makes it unsuitable for achieving optimal results in ever-changing risk scenarios.

5.3.2 Influence of the number of groups on the optimization effect

Parameter settings: In the experiments conducted, the system was tasked with managing a total of 400 tasks, with each group consisting of 5 agents. The number of groups varied between 8 and 16.

Experimental phenomena: Figure 3 shows the task completion rates achieved by different task migration algorithms, including MAAC, HAMA, and heuristic algorithms, in the context of 8 groups and 40 agents. The completion rates are comparable, but HAMRL significantly outperforms the other algorithms in achieving an optimal task completion rate. As the number of groups and the size of the problem increase, the task completion rate of the heuristic algorithm-based task migration approach gradually improves. However, the performance gap widens when compared to the optimization effects of the other approaches.

Overall, the task completion rate of the HAMA-based migration approach exceeds that of the MAAC-based approach. Furthermore, the optimization gap

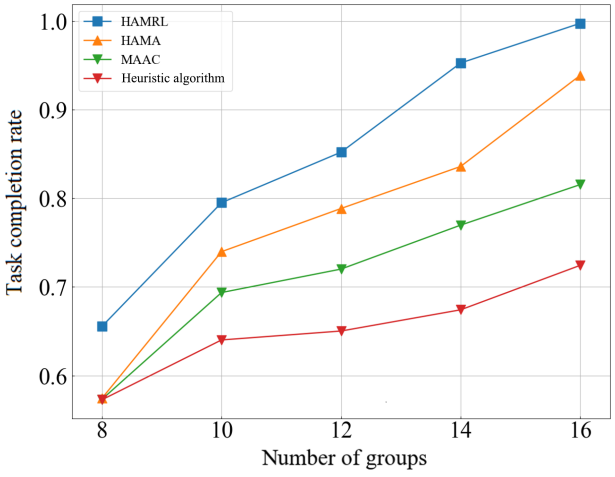


Fig. 3 Task completion rate comparison in UAV multiplex group.

between these two approaches consistently widens as the number of groups increases. Although the HAMRL approach consistently outperforms the other methods in terms of task completion rates, the completion rate continues to improve as the number of groups increases.

Figure 4 shows a marginal difference in total task completion cost for each approach with 8 groups. As the number of groups increases, the total task completion cost of the heuristic-based approach gradually increases, in line with the experimental data on task completion rates. In contrast, the remaining methods show a steeper increase in total task completion cost, but the differences between them remain relatively small.

Results analysis: HAMRL maintains a total task completion cost that is comparable to other algorithms, yet it also achieves the highest task completion rate. This achievement is attributed to our proposed hybrid attention network model, which has effective communication capabilities in dynamic high-risk environments. The model adopts a task migration strategy and directs task collaboration among agents, thereby optimizing task migration costs and improving task completion rates. The small differences in the total task migration costs using the HAMA-based and MAAC-based migration methods suggest that the former is more efficient in terms of task completion rates. It can be concluded that the HAMA-based approach focuses more on optimizing task migration costs than the MAAC-based approach, while still achieving a comparatively high task completion rate.

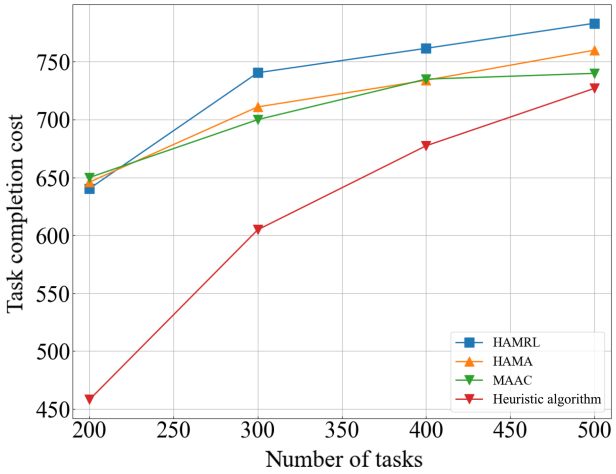


Fig. 4 Task completion cost comparison in UAV multiplex group.

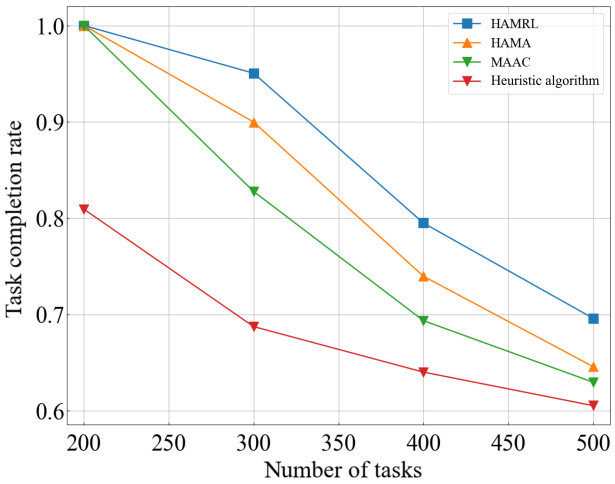


Fig. 5 Task completion rate comparison for the number of tasks.

5.4 Influence of the number of tasks on the optimization effect

Parameter settings: In the experiment, a total of 10 groups are utilized, each consisting of 5 agents. The number of tasks ranged from 200 to 500. Figures 5 and 6 show the effect of the number of tasks on the task completion rate and the total cost of task completion. When there are 200 tasks, the HAMRL, the HAMA-based approach, and the MAAC-based approach get a task completion rate of 1.0. In contrast, the heuristic-based approach resulted in a significantly lower task completion rate, which ultimately led to a reduction in the total cost of completing these tasks. As the number of tasks increased, the task completion rates of all four approaches decreases. However, HAMRL consistently maintained the highest task completion rate despite

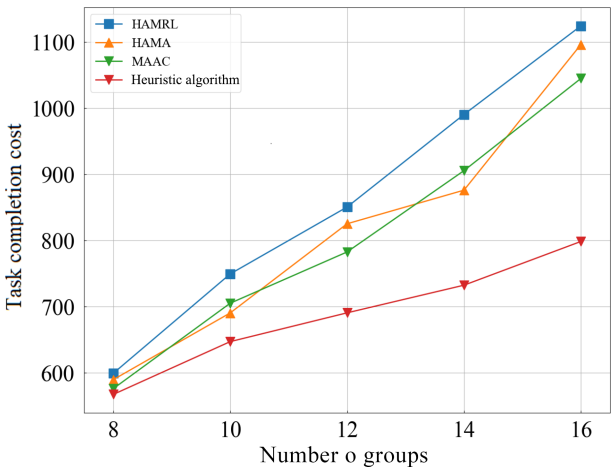


Fig. 6 Task completion cost comparison for the number of tasks.

having the highest task completion cost. The HAMA approach is superior to the MAAC approach in terms of task completion rate. However, the two approaches have similar total task completion costs, and the difference in optimization effect between the heuristic-based approach and the other approaches gradually decreases.

Results analysis: When there are 200 tasks, although the total task completion cost using HAMRL, the HAMA-based approach, and the MAAC-based approach are quite similar, HAMRL still has the lowest total task completion cost. This suggests that our proposed hybrid attention network model accurately learns the network topology across groups and effectively reduces the cost of task migration. Although HAMRL achieves the highest task completion rate and completes the most tasks, it incurs the highest total task completion cost in Fig. 6. However, HAMRL reduces the total task completion cost at high task completion rates because the method optimizes task migration costs. The HAMA-based method takes into account the group structure among agents and emphasizes the optimization of task migration costs, resulting in superior performance compared to the MAAC-based approach. Although the heuristic-based approach shows significant differences compared to other methods, it can still be used as a reliable and fast task migration strategy.

6 Conclusion

In this paper, we integrate the structural effects of dynamic risk on multiplex group agents into a task migration framework based on multi-agent reinforcement learning. We propose a hybrid attention multi-agent reinforcement learning algorithm. The experimental results show that our algorithm significantly outperforms state-of-the-art algorithms in terms of convergence speed and algorithm performance. For future research, an interesting direction involves extending the dynamic environment risk multiplex group task migration model to complex adversarial environments, aiming to enhance the robustness of the model.

Acknowledgment

This work was supported by the Key Research and Development Program of Jiangsu Province of China (No. BE2022157), the National Natural Science Foundation of China (Nos. 62303111, 62076060, and 61932007), the Defense Industrial Technology Development Program (No. JCKY2021214B002), and the Fellowship of China Postdoctoral Science Foundation (No. 2022M720715).

References

[1] Y. Jiang, K. Di, Z. Hu, F. Chen, P. Li, and Y. Jiang, ε-maximum critic deep deterministic policy gradient for multi-agent reinforcement learning, in *Proc. Int. Conf. on Parallel and Distributed Computing*: *Applications and Technologies*, Singapore, 2024, pp. 180–189.

[2] Y. Pan, Q. Ran, Y. Zeng, B. Ma, J. Tang, and L. Cao, Symmetric Bayesian personalized ranking with softmax weight, *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 53, no. 7, pp. 4314–4323, 2023.

[3] F. Yan and K. Di, Multi-robot task allocation in the environment with functional tasks, in *Proc. Thirty-First Int. Joint Conf. Artificial Intelligence*, Vienna, Austria, 2022.

[4] F. Yan and K. Di, Solving the Multi-robot task allocation with functional tasks based on a hyper-heuristic algorithm, *Appl. Soft Comput.*, vol. 146, p. 110628, 2023.

[5] L. Panait and S. Luke, Cooperative multi-agent learning: The state of the art, *Auton. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, 2005.

[6] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer, Protect: A deployed game theoretic system to protect the ports of the United States, presented at the 11th Conference in autonomous agents and multiagent systems, Minneapolis, MN, USA, 2012.

[7] I. Tkach and S. Amador, Towards addressing dynamic multi-agent task allocation in law enforcement, *Auton. Agents Multi-Agent Syst.*, vol. 35, no. 1, p. 11, 2021.

[8] D. Chen, M. R. Hajidavalloo, Z. Li, K. Chen, Y. Wang, L. Jiang, and Y. Wang, Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic, *IEEE Trans. Intell. Transport. Syst.*, vol. 24, no. 11, pp. 11623–11638, 2023.

[9] J. Ko, J. Jang, and C. Oh, A multi-agent driving simulation approach for evaluating the safety benefits of connected vehicles, *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 5, pp. 4512–4524, 2022.

[10] J. Zhang, Y. Cui, and J. Ren, Dynamic mission planning algorithm for UAV formation in battlefield environment, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 3750–3765, 2023.

[11] J. Guo, H. Gao, Z. Liu, F. Huang, J. Zhang, X. Li, and J. Ma, ICRA: An intelligent clustering routing approach for UAV ad hoc networks, *IEEE Trans. Intell. Transport. Syst*., vol. 24, no. 2, pp. 2447–2460, 2023.

[12] B. Liu, W. Zhang, W. Chen, H. Huang, and S. Guo, Online computation offloading and traffic routing for UAV swarms in edge-cloud computing, *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8777–8791, 2020.

[13] A. Kaur and K. Kumar, Energy-efficient resource allocation in cognitive radio networks under cooperative multi-agent model-free reinforcement learning schemes, *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 3, pp.

1337–1348, 2020.
[14] M. Hua, Y. Wang, C. Li, Y. Huang, and L. Yang, UAV-aided mobile edge computing systems with one by one access scheme, *IEEE Trans. Green Commun. Netw*., vol. 3, no. 3, pp. 664–678, 2019.
[15] J. Wang, K. Liu, and J. Pan, Online UAV-mounted edge server dispatching for mobile-to-mobile edge computing, *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1375–1386, 2020.
[16] A. Sacco, F. Esposito, and G. Marchetto, Resource inference for sustainable and responsive task offloading in challenged edge networks, *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1114–1127, 2021.
[17] B. Liu, W. Zhang, W. Chen, H. Huang, and S. Guo, Online computation offloading and traffic routing for UAV swarms in edge-cloud computing, *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, p. 1, 2020.
[18] Y. Liu, H. Yu, S. Xie, and Y. Zhang, Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks, *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158–11168, 2019.
[19] Z. Gao, L. Yang, and Y. Dai, Large-scale computation offloading using a multi-agent reinforcement learning in heterogeneous multi-access edge computing, *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3425–3443, 2023.
[20] X. Zhu, Y. Luo, A. Liu, M. Z. A. Bhuiyan, and S. Zhang, Multiagent deep reinforcement learning for vehicular computation offloading in IoT, *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9763–9773, 2021.
[21] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, Multiagent actor-critic for mixed cooperative-competitive environments, presented at the 31th Conference on Neural Information Processing Systems, California, CA, USA, 2017.
[22] A. Melin, Ocean Wave Prediction with LSTM, https://www.kaggle.com/code/akdagmelih/ocean-waveprediction-with-lstm, 2020.
[23] Y. Jiang, Y. Zhou, and Y. Li, Reliable task allocation with load balancing in multiplex networks, *ACM Trans. Auton. Adapt. Syst.*, vol. 10, no. 1, p. 3, 2015.
[24] W. Zhang, Y. Wen, and D. O. Wu, Collaborative task execution in mobile cloud computing under a stochastic wireless channel, *IEEE Trans. Wirel. Commun.*, vol. 14, no. 1, pp. 81–93, 2015.
[25] S. Iqbal and F. Sha, Actor-attention-critic for multi-agent reinforcement learning, arXiv preprint arXiv: 1810.02912, 2018.
[26] H. Ryu, H. Shin, and J. Park, Multiagent actor-critic with hierarchical graph attention network, presented at the 34th Conference on Association for the Advancement of Artificial Intelligence, New York, NY, USA, 2020.

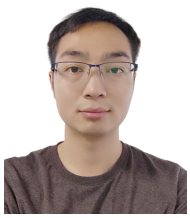
Yuanshuang Jiang received the MSc degree from Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, in 2021. He is currently working toward the PhD degree in computer science from School of Computer Science and Engineering, Southeast University, Nanjing, China. His research interests include reinforcement learning, multi-agent, and systems and data mining.

Kai Di received the PhD degree from School of Computer Science and Engineering, Southeast University, Nanjing, China. He is currently a postdoctoral researcher of School of Computer Science and Engineering, Southeast University. He has authored several scientific articles in refereed journals and conference proceedings, such as *ACM Transactions on Autonomous and Adaptive Systems*, and *ACM Transactions on Intelligent Systems and Technology*. His current research focuses on multi-agent systems.

Ruiyi Qian received the BE degree from Zhengzhou University, Zhengzhou, China, in 2021. He is currently working toward the master degree in computer science from School of Computer Science and Engineering, Southeast University, Nanjing, China. His current research focuses on multi-agent reinforcement learning.

Xingyu Wu received the BE degree from Southwest Jiaotong University, Chengdu, China, in 2023. He is currently working toward the master degree in computer science from School of Software Engineering, Southeast University, Nanjing, China. His current research focuses on multi-agent reinforcement learning.

Fulin Chen received the MSc degree in automation control from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2021. He is currently working toward the PhD degree at School of Cyber Science and Engineering, Southeast University, Nanjing, China. His current research focuses on data mining.



Pan Li received the MSc degree from Hebei University, Baoding, China, in 2022. He is currently pursuing the PhD degree in computer science from School of Cyber Science and Engineering, Southeast University, Nanjing, China. His current research focuses on time series analysis.

Yichuan Jiang received the PhD degree in computer science from Fudan University, Shanghai, China, in 2005. He is currently a distinguished professor and the director of Laboratory of Intelligent Systems and Social Computing, School of Computer Science and Engineering, Southeast University, Nanjing, China. He has authored or coauthored more than 100 scientific articles in refereed journals, such as *IEEE Transactions on Parallel and Distributed Systems, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing, IEEE Transactions on Systems Man, and Cybernetics: Systems, IEEE Transactions on Cybernetics, ACM Transactions on Autonomous and Adaptive Systems, ACM Transaction on Intelligent Systems, Technology Journal of Autonomous Agents and Multi-Agent Systems*, and in conference proceedings, such as IJCAl, AAAl, and AAMAS. His research interests include multi-agent systems, social computing, and social networks.

Xiping Fu is a senior data scientist at PredictHQ in New Zealand. He received the BS degree in mathematics from Zhejiang Normal University, China in 2007, the MSc degree in mathematics from Southeast University, China in 2010, and the PhD degree from University of Otago, New Zealand in 2016. His research interests encompass data mining, pattern recognition, and anomaly detection.