Graph-Attention Multi-Agent Reinforcement Learning with LLM-Guided Grouping for UAV-based Factory Fault Inspection

Tianyu Gao, Bo Yang, and Qiufeng Hu
College of Information Engineering, Northwest A&F University, China
gaotianyu@nwafu.edu.cn, yangbo_010@163.com, huqiufeng@nwafu.edu.cn

***Abstract*—Multi-agent reinforcement learning (MARL) has been widely applied to the collaborative operation of unmanned aerial vehicles (UAVs). However, as the task complexity and system scale increases, MARL suffers from policy homogeneity and ambiguous credit assignment, leading to low learning efficiency and limited coordination capability in complex environments. To address these challenges, we propose a large language model guided multi-agent approximate policy optimization (MAPPO-LLM) framework for efficient multi-UAV collaboration. By incorporating semantically grouped information, the framework provides agents with high-level role priors. We leverage graph structure and contrastive learning to achieve unified modeling of intra-group coordination consistency and inter-group policy diversity. Building upon this foundation, a group-aware centralized critic with self-attention mechanism is constructed to effectively characterize multi-agent coordination relationships and improve reward attribution. Extensive evaluation results on the UAV-based factory fault inspection task demonstrate that MAPPO-LLM significantly outperforms the baselines in terms of multi-UAV coordination efficiency, fault coverage rate and inspection delay.**

***Index Terms*—Unmanned aerial vehicle, multi-agent reinforcement learning, large language models, graph attention.**

I. Introduction

The rapid development of smart factory has significantly improved production efficiency and product quality through interconnected production lines and automatic workflows. As an important type of mobile intelligent systems, unmanned aerial vehicles (UAVs) have shown increasing practical value in many fields such as industrial inspection and emergency response [1]. The flexible maneuverability, three-dimensional coverage and non-contact monitoring capabilities of UAVs can break through the physical barriers in complex and hazardous industrial fields, which can significantly improve inspection efficiency and operational safety [2].

As an important method for solving distributed collaborative decision-making problems, multi-agent reinforcement learning (MARL) has been widely applied in many fields such as UAV scheduling and control. Through the interaction and trial-and-error exploration among multiple agents (i.e., UAVs), MARL autonomously learns strategies to cope with the dynamic environment changes [3]. However, the production efficiency requirement of the smart factory still poses a huge challenge to multi-UAV rapid decision-making and collaborative planning.

The recent advancements of Large Language Models (LLMs) provide new opportunities to improve the MARL paradigm. LLMs perform well in task abstraction, semantic planning, and knowledge generalization. They are able to interpret high-level goals, constraints and role semantics to provide structured guidance for policy generation [4]. Compared with traditional policy networks that rely heavily on a large number of task-specific interactions, LLMs can generalize to new tasks under zero-shot or few-shot conditions, and demonstrate context-aware reasoning capabilities in different environments. In multi-agent systems, LLMs can act as macro-planners, which decompose complex global tasks into structured subtasks and assign semantic intentions to individual agents [5]. By integrating task context and domain knowledge, LLMs can perform causal reasoning and evaluate how agent behaviors affect long-term outcomes [6] [7]. In addition, due to the exploratory nature of MARL, LLMs can simulate task changes and generate diverse experience samples, which can improve sample efficiency and promote more adaptive policy learning in dynamic or sparse reward environments.

Despite the potential of LLMs, existing works on LLM-enhanced MARL rarely investigate the practical systems with continuous action spaces for resource-constrained agents. Existing MARL methods still face two fundamental challenges in multi-UAV systems: (1) Although parameter sharing algorithms (e.g., QMIX [8] and MAPPO [9]) can accelerate cooperative learning, they lead to the strategy homogeneity issue. The shared network struggles to generate adaptive responses, and the joint strategy is influenced. (2) Team rewards require implicit credit allocation through joint state-action value functions. As the UAV swarm scale increases, the inefficiency in credit allocation becomes more severe.

To address the above challenges, we present a LLM-guided multi-agent approximate policy optimization (MAPPO-LLM) framework to provide high-level semantic priors for multi-agent coordination. The proposed approach leverages LLM-guided semantic grouping to mitigate policy homogeneity caused by parameter sharing. We also design a group-aware centralized critic to capture structured inter-agent interactions and improve credit assignment in multi-UAV systems.

ISBN 978-3-903176-82-9 ©2026 IFIP

The contributions of this paper are summarized as follows:

- To alleviate policy homogeneity caused by parameter sharing, we propose a contrastive learning mechanism for local policies to form more diverse and adaptive behaviors in the UAV swarm.
- We design a group-aware centralized critic with self-attention mechanism to explicitly model structured inter-group interactions, which can enhance the critic's ability to capture coordination patterns.
- We present the MAPPO-LLM framework and validate its effectiveness in the UAV-based factory fault inspection task. Extensive evaluation results display the superiorities in multi-UAV coordination efficiency, fault coverage rate and inspection delay.

II. Related Work

A. Multi-Agent Reinforcement Learning

Multi-agent reinforcement learning (MARL) has emerged as a key paradigm for coordinating multiple agents. Regarding the training paradigm, centralized training with decentralized execution (CTDE) is widely adopted, which enables agents to leverage global information during training while maintaining decentralized autonomy during execution. In the CTDE routine, value decomposition methods (e.g., QMIX [8], VDN [10]) decompose joint action value functions into individual utilities while ensuring consistency between global value and local policies. Besides value decomposition methods, some actor-critic approaches typically follow the CTDE paradigm. For example, MADDPG [11] introduces a centralized critic and decentralized actors to address the instability and credit allocation problems in multi-agent environments. Wei *et al.* [12] introduce an entropy-guided attention-based value factorization approach to enhance credit assignment by balancing local and global agent contributions. Miao *et al.* [13] develop a relative-entropy-regularized CTDE actor-critic framework to enhance policy consistency and sample efficiency in continuous MARL. Although exhibiting good performance in benchmark environments, these methods are primarily designed for centralized training with global or manually specified local rewards, and often fail to explicitly incorporate structural heterogeneity or high-level semantic abstraction that is common in practical multi-agent tasks.

Recent studies have applied MARL to multi-UAV coordination in resource-constrained communication environments. For example, Lagos *et al.* [14] combine graph structure modeling and MARL to address the dynamic deployment problem of UAV-assisted cellular networks. Hevesli *et al.* [15] propose a MAPPO variant with Beta distributions for energy-efficient operation in air-ground integrated networks. Zhao *et al.* [16] propose a Graph Neural Network (GNN)-enhanced MARL approach that leverages graph attention to learn potential-field-based distributed control for cooperative UAV swarms. Goeckner *et al.* [17] propose a GNN-enhanced MARL approach to achieve resilient distributed coordination under dynamic and unreliable communication conditions. Existing MARL methods address either coordination or credit assignment, but still lack mechanisms to incorporate high-level semantic information into large-scale MARL problems.

B. LLM-Enhanced Multi-Agent Reinforcement Learning

Recently, the rapid development of LLMs have brought new momentum to MARL. With strong capabilities in knowledge representation, reasoning and context modeling, LLMs can enhance state interpretation, goal expression, policy generation and task planning [18] [19]. Integrating LLM with MARL has shown great potential in tackling sparse rewards, long-term dependencies and instruction-driven tasks. For example, Zhu *et al.* [20] combine Q-value transformation with LLM to improve UAV trajectory planning via Graph Convolutional Network (GCN) and self-attention. Emami *et al.* [21] introduce an in-context prompting mechanism, in which LLMs guide UAV scheduling through natural language task descriptions. Zhou *et al.* [22] integrate LLM's global reasoning with Q-learning's local exploration to improve the planning performance in the traveling salesman problem.

However, existing works typically focus on generic coordination benchmarks or isolated planning components, leaving the integration of semantic reasoning and scalable multi-agent coordination in complex industrial scenarios insufficiently explored. This gap motivates us to develop a LLM-guided semantic grouping framework with graph-augmented MARL for efficient multi-UAV industrial inspection applications.

III. System Model and Problem Formulation

We consider an UAV-assisted industrial inspection and emergency response system deployed in a smart factory. The UAV swarm $\mathcal{U} = \{1, 2, \ldots, N_U\}$ work together to perform device inspections, fault coverage monitoring, and rapid response to emergencies. Ground-deployed industrial devices are denoted as $\mathcal{F} = \{1, 2, \ldots, N_F\}$ and each device is associated with a latent operation state, which indicates normal operation or the presence of a fault. Each UAV can interact with a subset of ground devices within its sensing range. Moreover, to reflect practical safety constraints in industrial environments, certain high-voltage areas are considered inaccessible to UAVs, which require flexible deployment strategies to ensure effective fault coverage. At each time step $t \in \{1, 2, \ldots, T\}$, UAVs need to make control and scheduling decisions.

A. UAV Motion Model

Assume that the UAVs maintain a certain altitude H. At each time step t, UAV j ($j \in \mathcal{U}$) executes action based on the action vector $\mathbf{a}_t^j = (\varrho_t^j, \sigma_t^j)$, where $\varrho_t^j \in [-1, 1]$ denotes the normalized directional control and $\sigma_t^j \in [0, 1]$ is the speed ratio relative to the maximum velocity $v_{\max}$. The direction ϱ_t^j is mapped to the angular direction $\theta_t^j \in [0, 2\pi]$. The UAV motion at time step t can be expressed as $\theta_t^j = (\varrho_t^j + 1) \cdot \pi$, the practical speed is $v_t^j = \sigma_t^j \cdot v_{\max}$. The position of the j-th UAV at the next time step can be expressed as

$$x_{t+1}^j = x_t^j + v_t^j \cdot \cos(\theta_t^j) \cdot \Delta t, \tag{1}$$

$$y_{t+1}^j = y_t^j + v_t^j \cdot \sin(\theta_t^j) \cdot \Delta t, \tag{2}$$

where $x_{t+1}^j, y_{t+1}^j \in [0, L]$ such that the updated position remains within the spatial boundary L, and Δt denotes the duration of a single motion control step.

Let $E_{\max}$ denote the UAV's initial energy. Energy cost primarily stems from hovering and propulsion. The propulsion energy cost of UAV j comprises blade power, induced power, and parasitic power, which can be expressed as

$$P(V^j) = \hat{P}\left(1 + \frac{3(V^j)^2}{U_{\text{tip}}^2}\right) + \tilde{P}\left(\sqrt{1 + \frac{(V^j)^4}{4v_0^4}} - \frac{(V^j)^2}{2v_0^2}\right) + \frac{1}{2} d_0 \rho \dot{s} A (V^j)^3, \tag{3}$$

where V^j denotes the instantaneous flight speed of UAV j, $\hat{P}$ denotes the blade profile power during hover, U_{tip} denotes the tip speed of the rotor blades, $\tilde{P}$ and v_0 denote the induced power and the average induced velocity during hover respectively, d_0 denotes the fuselage drag ratio, ρ denotes the air density, $\dot{s}$ denotes the rotor solidity, and A denotes the rotor disk area [23].

The total energy cost of UAV j can be expressed as

$$\Xi_t^j = \sum_{t=1}^{T} P\left(V_t^j\right) \cdot \Delta t. \tag{4}$$

B. Communication Model

For any factory device i ($i \in \mathcal{F}$), its fixed position is denoted as $\mathbf{q}^i = (x^i, y^i, 0)$. For any UAV j ($j \in \mathcal{U}$), its trajectory is denoted by a series of 3D coordinates $\mathbf{q}_t^j = (x_t^j, y_t^j, H)$ at time step t. We assume that drones do not interfere with each other during data transmission. The drone establishes a wireless inspection link with the ground device via a constant communication bandwidth B. Note that the time for data transmission and the energy consumed by data transmission are negligible as compared to the propulsion energy cost. The maximum achievable transmission rate is

$$\vartheta_t^{j,i} = B \log_2(1 + \Gamma_t^{j,i}), \tag{5}$$

where $\Gamma_t^{j,i}$ denotes the signal-to-noise ratio (SNR) between UAV j and ground device i at time step t.

The SNR can be expressed as

$$\Gamma_t^{j,i} = \frac{P_{\text{tx}}}{\psi^2 \cdot L_t^{j,i}}, \tag{6}$$

where P_{tx} denotes the transmission power, ψ^2 denotes the noise power encountered by UAV j when visually inspecting ground device i, and $L_t^{j,i}$ denotes the path loss between UAV j and the corresponding inspection target at time step t.

In practical industrial environments, due to complex factory layouts and dynamic mechanical operations, there are factors such as wall occlusions between different workshops, and the channel characteristics may change during mission execution. The path loss between ground device i and UAV j is

$$L_t^{j,i} = Pr_{\text{LoS}}{}_t^{j,i} \cdot L_{\text{LoS}}(d_t^{j,i}) + \left(1 - Pr_{\text{LoS}}{}_t^{j,i}\right) \cdot L_{\text{NLoS}}(d_t^{j,i}), \tag{7}$$

where L_{LoS} and L_{NLoS} denote the average additional path losses under LoS and NLoS conditions respectively. $Pr_{\text{LoS}}{}_t^{j,i}$ denotes the LoS link probability between UAV j and device i at time step t, $d_t^{j,i}$ denotes the 3D Euclidean distance between UAV j and device i at time step t, which is given as

$$d_t^{j,i} = \sqrt{(x_t^j - x^i)^2 + (y_t^j - y^i)^2 + H^2}. \tag{8}$$

C. Problem Formulation

This paper aims to minimize the overall inspection delay and maximize the spatial coverage in a multi-UAV collaborative inspection task. Let $\mathbf{Q} = \{\mathbf{q}_t^j\}$ denote the time-dependent trajectory of the UAV during the entire inspection period, $\varsigma^{j,i}$ denote the time required for UAV j to inspect the fault point i, and $\mathbf{X} = \{x^{j,i} \mid j \in \mathcal{U}, i \in \mathcal{F}\}$ denote the UAV task assignment matrix. The optimization problem $(\mathbf{P})$ is formulated as follows:

$$\min_{\mathbf{Q},\mathbf{X}} \quad w_1 \cdot \frac{1}{N_F} \sum_{\forall i \in \mathcal{F}} \sum_{\forall j \in \mathcal{U}} \varsigma^{j,i} \cdot x^{j,i} + w_2 \cdot \frac{1}{N_F} \sum_{\forall i \in \mathcal{F}} (1 - C_i), \tag{9a}$$

$$\text{s.t.} \quad x^{j,i} \in \{0,1\}, \quad \sum_{\forall j \in \mathcal{U}} x^{j,i} \leqslant 1, \tag{9b}$$

$$\sum_{\forall i \in \mathcal{F}} \varsigma^{j,i} \cdot x^{j,i} \leqslant T_{\max}^j, \tag{9c}$$

$$\|\mathbf{q}_{t+1}^j - \mathbf{q}_t^j\| \leqslant v_{\max} \cdot \Delta t, \forall t, \tag{9d}$$

$$x_{\min} \leqslant x_t^j \leqslant x_{\max}, \forall t, \tag{9e}$$

$$y_{\min} \leqslant y_t^j \leqslant y_{\max}, \forall t, \tag{9f}$$

$$C_i = \min\{1, \sum_{j \in \mathcal{U}} x^{j,i}\}. \tag{9g}$$

Constraint (9b) defines the binary assignment variable $x^{j,i}$ to ensure that each fault point is inspected by at most one UAV. Constraint (9c) limits the total inspection workload of each UAV by constraining the accumulated inspection time within $T_{\max}^j$. Constraint (9d) jointly enforces the UAV motion feasibility and operational safety by bounding per-step displacement. Constraints (9e) and (9f) restrict the UAV positions within the inspection area, and constraint (9g) defines the coverage indicator C_i, which equals to 1 if fault point i is successfully inspected.

Based on the above formulation, the original NP-hard mixed-integer nonlinear programming problem is approximately solved through cooperative MARL under the CTDE paradigm, in which each UAV learns a decentralized policy by using shared team rewards, enabling scalable decision-making in dynamic inspection environments.

IV. MAPPO-LLM Algorithm Framework

In this section, we present the MAPPO-LLM framework for multi-UAV factory inspection (as shown in Fig. 1). The environment is modeled as a decentralized partially observable Markov decision process (Dec-POMDP) under the CTDE paradigm. Building upon the problem formulation, we incorporate LLM-guided semantic reasoning into the MAPPO

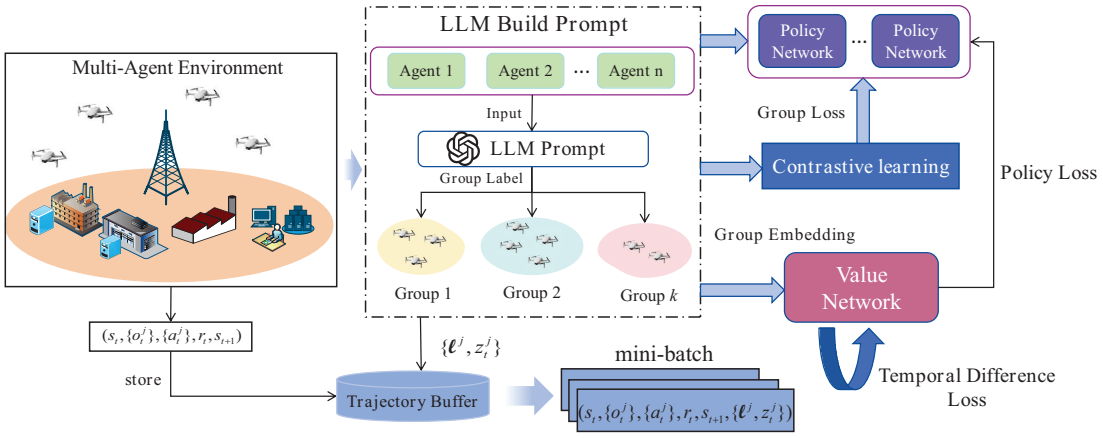


Fig. 1: MAPPO-LLM Framework

model to facilitate structured coordination and policy differentiation.

A. Dec-POMDP Formulation

In the smart factory inspection task, multiple UAVs operate in a dynamic and partially observable setting, where each UAV makes sequential decisions based on its internal state and local observations. The overall decision-making process is formulated as a Dec-POMDP. The problem is defined by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ denotes the global state space, $\mathcal{A}$ denotes the joint action space, $\mathcal{O}$ denotes the joint observation space, $\mathcal{R}$ defines the reward function, and $\gamma \in [0, 1]$ is the discount factor.

1) *Observations:* The observation of UAV j at time step t is defined as $o_t^j = \{\mathbf{q}_t^j,\ E_t^j,\ \mathcal{T}_t^j,\ \mathcal{N}_t^j\}$, where $\mathbf{q}_t^j$ denotes the position of UAV j, E_t^j denotes its remaining energy, $\mathcal{T}_t^j$ is the set of target points within its observation radius, and $\mathcal{N}_t^j$ includes the positions and velocities of neighboring UAVs within the same observation range.
2) *Actions:* At each time step t, the action of UAV j is defined as a continuous control vector $\mathbf{a}_t^j = \{\varrho_t^j, \sigma_t^j\}$, which denote the commands of direction and velocity ratio.
3) *Rewards:* The reward is defined as a weighted combination of inspection delay, coverage gain and collision penalty. The reward at time step t is defined as

$$r_t = -w_1\, T_t + w_2\, \Delta C_t - w_3\, \mathcal{CP}_t, \tag{10}$$

where $T_t = \sum_{j\in\mathcal{U}} \sum_{i\in\mathcal{F}} x^{j,i}(t)\, T^{j,i}$ denotes the total inspection delay at time step t, $\Delta C_t = \sum_{i\in\mathcal{F}} \left(C_t^i - C_{t-1}^i\right)$ denotes the coverage gain variation, and $\mathcal{CP}_t = \sum_{j\in\mathcal{U}} \mathcal{CP}_t^j$ denotes the total penalty over all UAVs at time step t.

B. Algorithm Workflow

As illustrated in Fig. 1, MAPPO-LLM integrates LLM-guided semantic grouping with MARL to enable structured coordination. At each grouping update, multi-agent observations and task contexts are encoded into prompts and fed to the LLM, which outputs a dynamic partition of agents into semantically consistent groups.

The resultant group labels are mapped to dense embeddings and incorporated into both the policy and value networks. A contrastive learning objective is applied to the group embeddings to encourage intra-group representation consistency and inter-group diversity, which can promote role-aware policy specialization. Meanwhile, graph-based encoders with attention mechanisms capture inter-agent interactions and provide structured inputs for decentralized decision-making.

During training, agent trajectories are stored in a buffer and sampled to jointly optimize the policy loss, value estimation loss and group contrastive loss. This unified training enables efficient credit assignment and scalable coordination learning.

C. LLM-Guided UAV Group Partitioning

To address the problems of homogeneous policy and inefficient credit assignment in MARL, we propose a LLM-guided adaptive grouping module that introduces high-level semantic priors into the coordination process. As the number of agents and the task complexity increase, it becomes increasingly difficult to design effective agent grouping and coordination strategies by using hand-crafted heuristics. Traditional rule-based grouping often fails to capture dynamic environment changes and the underlying functional dependencies among agents. To overcome this issue, our approach leverages the Qwen-VL Plus model[1] to dynamically assign semantically consistent group labels to agents prior to policy optimization.

Consistent with the Dec-POMDP formulation, the observation vector of UAV j is denoted as $\mathbf{o}^j$. It consists of the UAV's absolute position $\mathbf{p}^j$, velocity $\mathbf{v}^j$, and aggregated relative

[1]https://help.aliyun.com/zh/dashscope/developer-reference/qwen-vl-plus

positions to surrounding entities (including neighboring UAVs, fault points and restricted landmarks). Specifically,

$$\mathbf{o}^j = \left[\mathbf{p}^j,\, \mathbf{v}^j,\, \Delta\mathbf{p}^{j,u}_{\text{UAV}},\, \Delta\mathbf{p}^{j,i}_{\text{fault}},\, \Delta\mathbf{p}^{j,l}_{\text{landmark}}\right], \tag{11}$$

where $\Delta\mathbf{p}^{j,u}_{\text{UAV}}$ denotes the relative position between UAV j and another UAV $u \neq j$, $\Delta\mathbf{p}^{j,i}_{\text{fault}}$ denotes the relative position to fault point i, and $\Delta\mathbf{p}^{j,l}_{\text{landmark}}$ denotes the relative position to restricted landmark l.

The observations of $\{\mathbf{o}^j\}_{j=1}^{N_U}$, along with high-level task descriptors, are encoded as structured natural language prompt $\mathcal{P}_t$. This prompt is constructed by a modular template that clearly distinguishes among task-level constraints, agent-level properties and spatial layout descriptors. The modular template is designed to ensure model parsability and consistency across different environments. The LLM-based grouping function then returns a discrete partition of the agent population:

$$f_{\text{LLM}}(\mathcal{P}_t) = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k, \ldots, \mathcal{G}_K\}, \tag{12}$$

where $\mathcal{G}_k \subseteq \{1, 2, \ldots, N_U\}$ denotes the set of indices of the agents assigned to group k, and K denotes the total number of groups.

Each agent j is then assigned with a discrete group label ℓ^j:

$$\ell^j = k \quad \text{if} \quad j \in \mathcal{G}_k. \tag{13}$$

Each agent is assigned to exactly one group, and the resultant groups form a disjoint partition of the agent set. The group label ℓ^j is then converted into a dense semantic embedding via a learnable mapping:

$$\mathbf{z}^j_t = \text{Embedding}(\ell^j). \tag{14}$$

This embedding is integrated as auxiliary input into the downstream policy network, while group labels are used to construct structured representations for the critic. $\mathbf{z}^j_t$ can provide structured semantic prior knowledge to facilitate policy specialization and group-aware coordination. Through this approach, the MARL model can adaptively construct role-aware policies under dynamic task conditions and enhance inter-agent collaboration efficiency.

D. Policy Network based on Contrastive Learning and LLM

In the standard MAPPO model, the actor network $\pi_\theta(a^j_t|o^j_t)$ takes the local observation o^j_t of agent j as input, and outputs a probability distribution over actions. The policy parameters are updated by the Proximal Policy Optimization (PPO) objective:

$$\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_t\Big[\min\Big\{\varpi_t(\theta)\hat{A}_t, \text{clip}(\varpi_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\Big\}\Big], \tag{15}$$

where $\varpi_t(\theta) = \frac{\pi_\theta(a^j_t|o^j_t)}{\pi_{\theta_{\text{old}}}(a^j_t|o^j_t)}$ denotes the importance sampling ratio between the updated and previous policies, $\hat{A}_t$ denotes the advantage estimate provided by the centralized critic, and ϵ denotes the PPO clipping parameter that limits excessive policy updates.

However, conventional actor networks often rely on local observations, ignoring inter-agent structural relations or role semantics. To address this limitation, we design a structure-enhanced actor network that incorporates two additional information streams: (1) group label embeddings generated by the LLM-guided grouping module; (2) spatial message features extracted via a graph attention mechanism. The input to each agent's actor network is augmented as:

$$\tilde{o}^j_t = \left[o^j_t,\, \mathbf{m}^j_t,\, \mathbf{z}^j_t\right], \tag{16}$$

where $\mathbf{z}^j_t$ denotes the semantic group embedding derived from the LLM, and $\mathbf{m}^j_t$ denotes the spatial message aggregated from neighboring agents.

Let $\mathbf{X}_t = [\mathbf{x}^1_t, \ldots, \mathbf{x}^j_t, \ldots, \mathbf{x}^{N_U}_t] \in \mathbb{R}^{N_U \times d_x}$ denote the stacked feature matrix of all N_U agents, where $\mathbf{x}^j_t \in \mathbb{R}^{d_x}$ is the feature vector of agent j and d_x denotes the feature dimension of each agent. Let $\mathcal{E}$ denote the interaction edge set among agents, which defines the graph structure for inter-agent message passing. We utilize a two-layer GCNConv encoder (i.e., GCNConv–ReLU–GCNConv) to produce the query representations:

$$\begin{aligned} \mathbf{H}^Q_t &= \text{ReLU}\Big(\text{GCNConv}_1(\mathbf{X}_t, \mathcal{E}; \hat{\mathbf{W}}^Q)\Big), \\ \mathbf{Q}_t &= \text{GCNConv}_2(\mathbf{H}^Q_t, \mathcal{E}; \tilde{\mathbf{W}}^Q), \end{aligned} \tag{17}$$

where $\hat{\mathbf{W}}^Q$ and $\tilde{\mathbf{W}}^Q$ are learnable vectors.

As for the non-shared parameters $(\hat{\mathbf{W}}^K, \tilde{\mathbf{W}}^K)$, the key representations $\mathbf{K}_t$ at time step t are calculated in the same manner as $\mathbf{Q}_t$. It allows the query and key encoders to capture different semantic roles in inter-agent interactions by using non-shared parameters, which are beneficial for modeling heterogeneous behaviors and role differentiation. The linear transform in the first Query layer can be written as

$$\mathbf{U}^Q_t = \underbrace{\begin{bmatrix} x^{1,1}_t & \cdots & x^{1,d_x}_t \\ \vdots & \ddots & \vdots \\ x^{N_U,1}_t & \cdots & x^{N_U,d_x}_t \end{bmatrix}}_{\mathbf{x}_t \in \mathbb{R}^{N_U \times d_x}} \underbrace{\begin{bmatrix} w^Q_{(1,1)} & \cdots & w^Q_{(1,d_h)} \\ \vdots & \ddots & \vdots \\ w^Q_{(d_x,1)} & \cdots & w^Q_{(d_x,d_h)} \end{bmatrix}}_{\widetilde{\mathbf{W}}^Q \in \mathbb{R}^{d_x \times d_h}}, \tag{18}$$

where d_h denotes the hidden feature dimension after the first graph convolutional projection.

For clarity, the above expression highlights the learnable linear projection in GCNConv, while the complete operation includes degree-normalized message passing over the inter-action graph $\mathcal{E}$. The query and key representations are only used to calculate inter-agent attention weights for message construction, and are not directly involved in action prediction. Each GCNConv further performs normalized neighborhood propagation on $\mathcal{E}$. The attention weights are calculated by

$$\begin{aligned} \boldsymbol{\alpha}_t &= \text{softmax}_{\text{row}}(\mathbf{Q}_t\mathbf{K}_t^\top) \in \mathbb{R}^{N_U \times N_U}, \\ \bar{\boldsymbol{\alpha}}_t &= \text{RemoveDiag}(\boldsymbol{\alpha}_t) \in \mathbb{R}^{N_U \times (N_U - 1)}, \end{aligned} \tag{19}$$

where $\text{RemoveDiag}(\cdot)$ removes the self-attention terms after the softmax operation. For agent j, let $\mathbf{R}^j_t \in \mathbb{R}^{(N_U-1)\times 2}$ stack the relative-position vectors $\{\mathbf{r}^{j,\kappa}_t\}_{\kappa \neq j}$ (κ denotes another agent that interacts with agent j), and let $\bar{\boldsymbol{\alpha}}^j_t \in \mathbb{R}^{N_U - 1}$ denote the corresponding attention weights.

The spatial message is formed via weighted concatenation:

$$\mathbf{m}_t^j = \mathrm{vec}\Big(\mathrm{diag}\big(\bar{\boldsymbol{\alpha}}_t^j\big)\,\mathbf{R}_t^j\Big) \in \mathbb{R}^{2(N_U-1)}. \tag{20}$$

To enhance the policy's structural characterization, we further introduce contrastive learning under advantageous conditions. Unlike traditional contrastive regularization, we explicitly link the contrastive objective to the advantage estimates provided by the PPO algorithm. The contrastive loss is

$$\begin{aligned}\mathcal{L}_{\text{contrast}} = \mathbb{E}_t^j\Bigg[&\frac{1}{|\mathbb{P}_t^j|}\sum_{\kappa\in|\mathbb{P}_t^j|}\frac{|\hat{A}_t^j|}{\bar{A}}\,\ell_{\text{pos}}\Big(p_t^{j\to\kappa}\Big)\\ &+\frac{1}{|\mathbb{N}_t^j|}\sum_{\kappa\in|\mathbb{N}_t^j|}\frac{|\hat{A}_t^j|+|\hat{A}_t^\kappa|}{2\bar{A}}\,\ell_{\text{neg}}\Big(p_t^{j\to\kappa}\Big)\Bigg],\end{aligned} \tag{21}$$

where $\mathbb{P}_t^j$ and $\mathbb{N}_t^j$ denote the positive (same-group) and negative (different-group) agent sets with respect to agent j at time step t, respectively. The normalization term $\bar{A} = \mathbb{E}_t^j[|\hat{A}_t^j|] + \xi$ is introduced to stabilize training across batches with varying advantage magnitudes. Based on the latent representations $\mathbf{z}_t^j$ and $\mathbf{z}_t^\kappa$, we define the similarity-induced contrast probability $p_t^{j\to\kappa}$ as follows:

$$p_t^{j\to\kappa} = \frac{\exp\Big(\mathrm{sim}(\mathbf{z}_t^j,\mathbf{z}_t^\kappa)/\chi\Big)}{\sum_{v\neq j}\exp\Big(\mathrm{sim}(\mathbf{z}_t^j,\mathbf{z}_t^v)/\chi\Big)}, \tag{22}$$

where χ is a temperature factor, and $\mathrm{sim}(\cdot,\cdot)$ denotes the cosine similarity:

$$\mathrm{sim}(\mathbf{z}^j,\mathbf{z}^\kappa) = \frac{(\mathbf{z}^j)^\top\mathbf{z}^\kappa}{\|\mathbf{z}^j\|\,\|\mathbf{z}^\kappa\|}. \tag{23}$$

Accordingly, the positive and negative contrastive terms are defined on the induced probability as

$$\ell_{\text{pos}}\Big(p_t^{j\to\kappa}\Big) = -\log\Big(p_t^{j\to\kappa}\Big), \tag{24}$$

$$\ell_{\text{neg}}\Big(p_t^{j\to\kappa}\Big) = -\log\Big(1-p_t^{j\to\kappa}\Big). \tag{25}$$

The contrastive loss is optimized jointly with the PPO objective as an auxiliary regularization term for the actor network. By weighting the contrastive loss with the advantage magnitude, the regularization strength is adaptively adjusted to emphasize transitions that contribute more significantly to policy updates.

The overall optimization objective for the actor network is

$$\mathcal{L}_{\text{actor}}(\theta) = \mathcal{L}_{\text{PPO}}(\theta) + \lambda_{\text{contrast}}\cdot\mathcal{L}_{\text{contrast}}(\theta). \tag{26}$$

and the parameters are updated via gradient descent:

$$\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\text{actor}}(\theta), \tag{27}$$

where $\lambda_{\text{contrast}}$ is the contrastive loss coefficient, and η is the learning rate.

In summary, the policy network integrates semantic priors from LLM-guided grouping and spatial reasoning from graph structures. The incorporation of contrastive learning can foster specialization within groups and diversity across groups, and can make role-aware, structurally consistent and behaviorally diverse policies in cooperative multi-agent settings.

E. Value Network with Group-Level Structural Attention

To enhance structural awareness and stabilize advantage estimation in MARL, we design a group-aware critic, whose internal representation explicitly models group-level structure while retaining a shared global value output.

We utilize the term group to denote each semantic group produced by the LLM-guided clustering. Let $\mathcal{G}_k = \{\, j \mid \ell^j = k \,\}$ denote the set of agents assigned to group k. We aggregate agent observations within each group to obtain a compact group-level representation as:

$$g_t^k = \frac{1}{|\mathcal{G}_k|}\sum_{j\in\mathcal{G}_k}\phi(o_t^j), \tag{28}$$

where $\phi(\cdot)$ is an agent-level encoder implemented by an MLP with two hidden layers. Group semantics are injected into the critic through the LLM-guided partition rather than explicit group embeddings. Then we apply an attention-style similarity weighting over group representations. Each group embedding is refined via a residual projection:

$$\tilde{g}_t^k = g_t^k + W g_t^k, \tag{29}$$

where W is a trainable linear transformation. The residual formulation preserves original group semantics while allowing task-adaptive refinement, which improves training stability. The attention weight from group k to group h is

$$\beta^{k,h} = \frac{\exp(\tilde{g}_t^k\cdot\tilde{g}_t^h/\sqrt{d_g})}{\sum_{\iota=1}^K\exp(\tilde{g}_t^k\cdot\tilde{g}_t^\iota/\sqrt{d_g})}, \tag{30}$$

where d_g denotes the feature dimension of each group-level representation $\tilde{g}_t^k$.

When inter-group dependencies are weak or uninformative, the attention weights tend to be nearly uniform, which reduce to simple averaging.

Each group aggregates information from all other groups, which is expressed as

$$\mathbf{h}_t^k = \sum_{h=1}^K \beta^{k,h}\,\tilde{g}_t^h. \tag{31}$$

The global group context is obtained by averaging all group representations as follows:

$$\mathbf{h}_t^{\text{grp}} = \frac{1}{K}\sum_{k=1}^K\mathbf{h}_t^k. \tag{32}$$

This aggregation yields a permutation-invariant global structural context that summarizes inter-group relationships without imposing an ordering bias.

To ensure training stability, we formulate the value function as a gated residual of the standard MAPPO critic:

$$V(s_t) = V_{\text{base}}(s_t) + g(s_t)\cdot f_{\text{grp}}(\mathbf{h}_t^{\text{grp}}), \tag{33}$$

where $s_t = \{\mathbf{q}_t^{all}, E_t^{all}, \mathcal{T}_t^{all}, \mathcal{N}_t^{all}\}$ denotes the global state at time step t, $V_{\text{base}}(s_t)$ is the standard centralized value function, $f_{\text{grp}}(\cdot)$ is an MLP with two hidden layers that map the group context to a scalar, and $g(s_t)$ is a learnable gate.

Algorithm 1: MAPPO-LLM: Training with LLM-Guided Grouping and Group-Attention Value Network

Initialize policy network π_θ, value network V_{θ_v}, and trajectory buffer $\mathcal{D}$;
for *each episode* $= 1$ *to* E **do**
  Reset environment and obtain initial state s_0 and observations $\{o_0^j\}_{j=1}^{N_U}$;
  Query LLM to obtain semantic group labels $\{\ell^j\}_{j=1}^{N_U}$ and embeddings $\{z_t^j\}_{j=1}^{N_U}$ based on task context;
  for *each time step* $t = 1$ *to* T **do**
    Construct policy inputs from local observations o_t^j, graph-based neighbor messages, and fixed group embeddings z_t^j;
    Sample actions $\{a_t^j\}_{j=1}^{N_U}$ from the policy π_θ;
    Execute the joint action $\mathbf{a}_t$ and observe reward r_t, next state s_{t+1} and observations $\{o_{t+1}^j\}$;
    Store transition $(s_t, \{o_t^j\}, \mathbf{a}_t, r_t, s_{t+1}, \{\ell^j, z_t^j\})$;
  for *each mini-batch sampled from* $\mathcal{D}$ **do**
    Build group-level features based on group labels and agent trajectories;
    Aggregate group-level structural context via graph-based message passing;
    Evaluate state value $V(s_t)$ via the group-aware value network;
    Calculate temporal difference targets and advantages $\hat{A}_t^j$ by using generalized advantage estimation;
    Calculate advantage-weighted contrastive loss $\mathcal{L}_{\text{contrast}}$ for the actor representation;
    Update policy parameters θ by minimizing $\mathcal{L}_{\text{PPO}}(\theta) + \lambda_{\text{contrast}}\mathcal{L}_{\text{contrast}}(\theta)$;
    Update value network parameters θ_v by minimizing $\mathcal{L}_{\text{value}}(\theta_v)$;

The calculated state values are used to estimate advantages for each agent by means of Generalized Advantage Estimation (GAE) [24], which can be expressed as

$$\hat{A}_t^j = \sum_{l=0}^{T-t} (\gamma\lambda)^l \left(r_{t+l} + \gamma V(s_{t+l+1}) - V(s_{t+l})\right), \tag{34}$$

where λ denotes the GAE parameter that controls the bias-variance trade-off in advantage estimation. The value network is optimized by the Temporal Difference (TD)-based mean squared error loss, which is expressed as

$$\mathcal{L}_{\text{value}}(\theta_v) = \sum_t \left(V_{\theta_v}(s_t) - R_t\right)^2, \tag{35}$$

where R_t denotes the TD-based value target calculated from shared team rewards under the standard MAPPO formulation, while the corresponding advantage estimates are calculated by generalized advantage estimation. To optimize this objective, the critic parameters (i.e., θ_v) are updated via gradient descent:

$$\theta_v \leftarrow \theta_v - \eta_v \nabla_{\theta_v} \mathcal{L}_{\text{value}}(\theta_v). \tag{36}$$

In addition, a contrastive loss $\mathcal{L}_{\text{contrast}}$ is applied to the region encoder to enforce semantic consistency within each LLM-guided group and sharpen inter-group discrepancies. Although it does not directly update the value network, this regularization improves the quality of region embeddings and thereby enhances the critic's structural modeling capability. The training procedure is presented in Algorithm 1.

TABLE I: Simulation Parameter Settings

Parameter	Value	Parameter	Value
η	3×10^{-4}	$v_{\max}$	10 m/s
$\lambda_{\text{contrast}}$	0.2	γ	0.99
η_v	1×10^{-4}	ξ	0.2
λ	0.95	B	1 MHz
P_{tx}	0.1 W	ψ^2	10^{-13} W
$Pr_{\text{LoS}}^{j,i}$	0.7	$E_{\max}$	10000 J
$\hat{P}$	79.86 W	$\tilde{P}$	88.63 W
U_{tip}	120 m/s	v_0	3 m/s
d_0	0.6	ρ	1.225 kg/m^3
$\dot{s}$	0.05	A	0.503 m^2

V. Simulation Results

We evaluate the MAPPO-LLM framework in a 100m$\times$ 100m factory scenario with multiple UAVs and ground fault points. To assess robustness and scalability, we vary the numbers of UAVs and fault points to cover different levels of inspection task complexity. In the simulations, UAVs and fault points are randomly initialized, and the policies are updated by PPO. The main parameters are listed in Table I.

A. Evaluation on Reward and Sample Number

To validate the performances of the MAPPO-LLM framework, we select three representative baselines: QMIX [8], MAPPO [9] and MADDPG [11].

The reward curves during training are shown in Fig. 2. The QMIX and MADDPG methods maintain relatively low rewards. Both methods rely primarily on local observations and lack effective global coordination, which result in uneven workload distribution among UAVs and convergence to suboptimal policies. The MAPPO method benefits from centralized training and achieves more stable learning. However, it cannot capture the evolving relationships between UAVs and fault points. As a result, its reward improvement remains slow. In contrast, our MAPPO-LLM method exhibits faster growth in reward. MAPPO-LLM achieves a steeper mid-training reward increase, indicating more efficient coordination learning. It also converges to a higher and more stable reward level, while the baselines converge slowly or plateau at lower rewards. As shown in Table II, MAPPO-LLM consistently reaches the target rewards with fewer sample numbers than all baselines, which validates the training efficiency of MAPPO-LLM.

TABLE II: Comparisons on sample numbers under different rewards

reward	QMIX	MAPPO	MADDPG	MAPPO-LLM
10	58.5	53.6	39.4	**31.4**
30	135.5	117.7	135.5	**85.6**
50	211.3	156.5	166.9	**99.8**
70	238.4	163.2	186.0	**137.4**
90	263.0	184.8	194.0	**149.7**

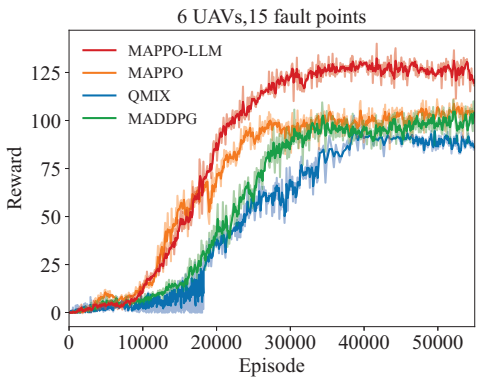


Fig. 2: Comparisons on the rewards

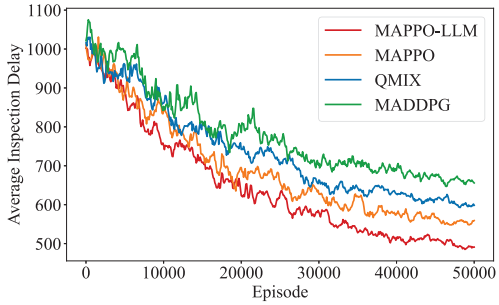


Fig. 3: Comparisons on average inspection delays

B. Evaluation on Inspection Delay

Fig. 3 compares the average inspection delays. MADDPG shows the slowest convergence rate and obvious oscillations, which indicates its instability under multi-agent dynamics. QMIX exhibits more stable learning than MADDPG but converges to a suboptimal delay level due to the restricted expressiveness imposed by monotonic value decomposition. The MAPPO method converges slower and remains at a relatively higher delay than our MAPPO-LLM method due to the limited scalability of centralized critics in multi-UAV inspection tasks. In contrast, MAPPO-LLM converges significantly faster and achieves the lowest steady-state inspection delay. The LLM-guided grouping mechanism can reduce redundant inspections by virtue of effective task partitioning, and the graph-enhanced policy can capture spatial relationships among UAVs and inspection points to support efficient coordination.

C. Evaluation on Coverage Rate

As shown in Fig. 4(a), in the scenario with 3 UAVs and 8 fault points, all methods improve the coverage rates of fault points over time. However, MAPPO, QMIX and MADDPG converge more slowly and achieve lower coverage rates, whereas our MAPPO-LLM method rapidly stabilizes at near-complete coverage. As shown in Fig. 4(b), in the scenario with 5 UAVs and 12 fault points, MAPPO, QMIX and MADDPG exhibit noticeable oscillations and slower convergence. In contrast, our method achieves faster and more stable coverage growth. This advantage becomes more pronounced in the expanded scenario with 8 UAVs and 20 fault points, in which the baselines struggle to scale and yield lower coverage rates, as shown in Fig. 4(c). Differently, MAPPO-LLM can maintain high coverage rates throughout the training.

D. Evaluation on Energy Cost

As shown in Fig. 5, the energy costs of all methods steadily increase as the swarm scale increases from 3 UAVs to 8 UAVs due to longer flight trajectories and higher coordination requirements. Among the baselines, MADDPG and QMIX incur relatively high energy costs, which indicate the inefficiencies caused by unstable policy updates and over-exploration. The energy cost trend of MAPPO is relatively stable, but its centralized value function leads to redundant trajectories, which increase the overall energy cost. In contrast, our MAPPO-LLM method consistently achieves the lowest energy cost across the test scenarios. Notably, the performance gaps between our method and the baselines are enlarged as the scenario size increases, which demonstrate that the proposed framework is effective in large-scale inspection tasks. The reason is that our method can exploit structural relationships among drones through the LLM-guided grouping strategy and graph augmentation strategy, which can significantly reduce redundant trajectories and improve coordination efficiency as the numbers of UAVs and fault points increase.

VI. Conclusion

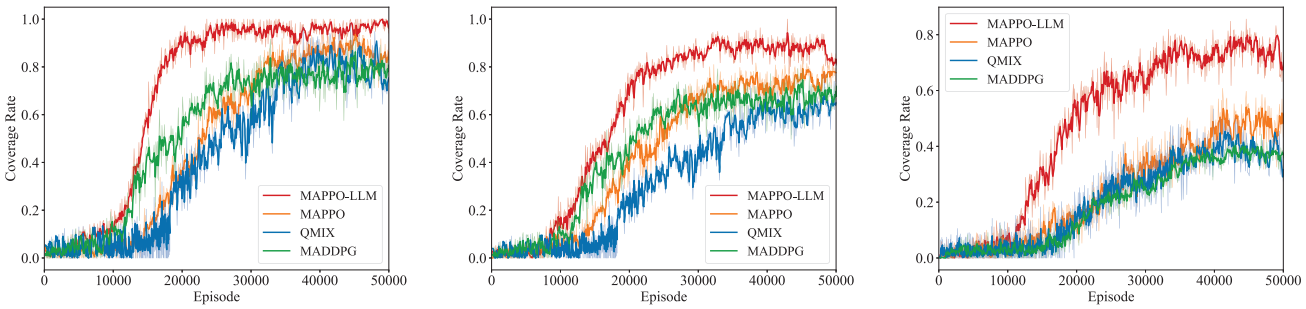
This paper presents a multi-UAV reinforcement learning framework termed MAPPO-LLM for factory fault inspection. By introducing a semantic grouping mechanism guided by large language models, our method can achieve structured task partitioning and reduce redundant inspections. Moreover, we design a graph-attention-enhanced learning mechanism to explicitly model coordination relationships among UAVs. The MAPPO-LLM framework combines advanced semantic reasoning with structured relationship to deliver a scalable solution for multi-UAV inspection tasks. Extensive simulations across diverse factory inspection scenarios demonstrate the superiorities of our method in terms of convergence efficiency, fault coverage rate, inspection delay and energy efficiency.

Acknowledgements

This research was supported by the Key Research and Development Projects of Shaanxi Province (No. 2025SF-YBXM-291), Fundamental Frontier Research Projects of Shaanxi Provincial Agriculture Department (No. 2025JCQY036), Qinchuangyuan Innovation and Entrepreneurship Talent Project (No. QCYRCXM-2022-353), and Chinese Universities Scientific Fund (No. 2452025417). Bo Yang is the corresponding author.

References

[1] G. Lee, W. Saad, M. Bennis, C. Kim, and M. Jung, "An online framework for ephemeral edge computing in the Internet of things," *IEEE Transactions on Wireless Communications*, vol. 22, no. 3, pp. 1992–2007, 2023.



(a) 3 UAVs, 8 fault points (b) 5 UAVs, 12 fault points (c) 8 UAVs, 20 fault points

Fig. 4: Comparisons on coverage rates under different inspection configurations

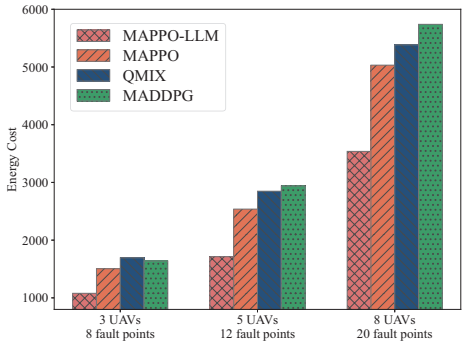


Fig. 5: Comparisons on energy costs

[2] X. Li, Y. Qin, Y. Liu, X. Xu, W. Huangfu, and K. Long, "Efficient and secure UAV-assisted industrial Internet of things based on confidence-weighted reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 12, pp. 5307–5319, 2026.

[3] Y. Bai, B. Xie, Y. Liu, Z. Chang, and R. Jntti, "Dynamic UAV deployment in multi-UAV wireless networks: A multimodal-feature-based deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 18765–18778, 2025.

[4] B. Han, Y. Chen, J. Li, J. Li, and J. Su, "Swarmchain: Collaborative LLM inference for UAV swarm control," *IEEE Internet of Things Magazine*, vol. 8, no. 5, pp. 64–71, 2025.

[5] Y. Cao, H. Zhao, Y. Cheng, T. Shu, Y. Chen, G. Liu, G. Liang, J. Zhao, J. Yan, and Y. Li, "Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 6, pp. 9737–9757, 2025.

[6] N. Wang, B. Yao, J. Zhou, Y. Hu, X. Wang, Z. Jiang, and N. Guan, "Large language model for verilog generation with code-structure-guided reinforcement learning," in *2025 IEEE International Conference on LLM-Aided Design (ICLAD)*, 2025, pp. 164–170.

[7] S. Alam and W.-C. Song, "Enhancing network intelligence with LLM-based IBN and DRL: A dynamic approach for SAGIN resource management," in *2025 International Conference on Computing, Networking and Communications (ICNC)*, 2025, pp. 723–727.

[8] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[9] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *2022 Conference on Neural Information Processing Systems (NeurIPS)*, 2022, pp. 24611–24624.

[10] T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Value-decomposition networks for cooperative multi-agent learning," in *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2017, pp. 208–216.

[11] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *2017 Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6379–6390.

[12] W. Wei, H. Li, S. Zhou, B. Li, and X. Liu, "Attention with system entropy for optimizing credit assignment in cooperative multi-agent reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 14775–14787, 2025.

[13] C. Miao, Y. Cui, H. Li, and X. Wu, "Effective multi-agent deep reinforcement learning control with relative entropy regularization," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 3704–3718, 2025.

[14] D. M. M. Lagos, C. A. Azurdia-Meza, and J. Ruiz-del Solar, "Fair coverage for unmanned aerial vehicle-assisted cellular networks with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 13, no. 3, pp. 5202–5223, 2026.

[15] M. Hevesli, A. M. Seid, A. Erbad, and M. Abdallah, "Multi-agent DRL for queue-aware task offloading in hierarchical MEC-enabled air-ground networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 12, pp. 217–236, 2026.

[16] B. Zhao, M. Huo, Z. Li, Z. Yu, and N. Qi, "Graph-based multi-agent reinforcement learning for large-scale UAV swarm system control," *Aerospace Science and Technology*, vol. 150, pp. 109166, 2024.

[17] A. Goeckner, Y. Sui, N. Martinet, X. Li, and Q. Zhu, "Graph neural network-based multi-agent reinforcement learning for resilient distributed coordination of multi-robot systems," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 5732–5739.

[18] C. Sun, S. Huang, and D. Pompili, "LLM-based multi-agent decision-making: Challenges and future directions," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5681–5688, 2025.

[19] Y. Ren, H. Zhang, F. R. Yu, W. Li, P. Zhao, and Y. He, "Industrial Internet of things with large language models (LLMs): An intelligence-based reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 24, no. 5, pp. 4136–4152, 2025.

[20] F. Zhu, F. Huang, Y. Yu, G. Liu, and T. Huang, "Task offloading with LLM-enhanced multi-agent reinforcement learning in UAV-assisted edge computing," *Sensors*, vol. 25, no. 1, p. 175, 2024.

[21] Y. Emami, H. Zhou, S. Nabavirazavi, and L. Almeida, "LLM-enabled in-context learning for data collection scheduling in UAV-assisted sensor networks," *IEEE Internet of Things Journal*, vol. 12, no. 23, pp. 51664–51676, 2025.

[22] Q. Zhou, J. Wu, M. Zhu, Y. Zhou, F. Xiao, and Y. Zhang, "LLM-QL: A LLM-enhanced Q-Learning approach for scheduling multiple parallel drones," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 9, pp. 5393–5406, 2025.

[23] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[24] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *International Conference on Learning Representations (ICLR)*, 2016.