2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Transformer-based Multi-Agent Reinforcement Learning for Generalization of Heterogeneous Multi-Robot Cooperation

Yuxin Cai[1,2], Xiangkun He[1], Hongliang Guo[2], Wei-Yun Yau[2], and Chen Lv[1,*]

***Abstract*— Recent advances in multi-agent reinforcement learning (MARL) have significantly enhanced cooperation capabilities within multi-robot teams. However, the application to heterogeneous teams poses the critical challenge of combinatorial generalization—adapting learned policies to teams with new compositions of varying sizes and robots capabilities. This challenge is paramount for dynamic real-world scenarios where teams must swiftly adapt to changing environmental and task conditions. To address this, we introduce a novel transformer-based MARL method for heterogeneous multi-robot cooperation. Our approach leverages graph neural networks and self-attention mechanisms to effectively capture the intricate dynamics among heterogeneous robots, facilitating policy adaptation to team size variations. Moreover, by treating robot team decisions as sequential inputs, a capability-oriented decoder is introduced to generate actions in an auto-regressive manner, enabling decentralized decision-making that tailored each robot's varying capabilities and heterogeneity type. Furthermore, we evaluate our method across two heterogeneous cooperation scenarios in both simulated and real-world environments, featuring variations in team number and robot capabilities. Comparative results reveal our method's superior generalization performance compared to existing MARL methodologies, marking its potential for real-world multi-robot applications.**

I. INTRODUCTION

The dynamic and unpredictable nature of contemporary environments presents considerable challenges for the deployment of autonomous robots. The complexity of adapting to every conceivable scenario with a single robot type has necessitated the development of heterogeneous multi-robot systems (HMRS). Distinct characteristics and capabilities inherent to each robot type comprise cooperation. In robotics, HMRS are crucial across a broad spectrum of domains, including search and rescue operations [1], [2], collaborative exploration and mapping [3], and other industrial tasks [4]. Within these systems, robots exhibit varied capabilities and functionalities unlock significant potential for cooperative behaviors, facilitating the accomplishment of tasks that are beyond the reach of homogeneous robot team. However, heterogeneity that enhances their utility also complicates their cooperation. The generalization of learned HMRS policies to new team compositions, which specifically refers to variations in team size and capabilities of individual robots, poses a crucial challenge. For example, in search and rescue missions, composition of robot team changes due to new robots possessing different capabilities being introduced to the team. Robot team must be able to adapt to these changes and maintain efficient coordination.

Traditional control methodologies, such as rule-based systems [5] and heuristic approaches [6], often rely on pre-established protocols or strategies to due with different team compositions. While these strategies may prove sufficient in static and predictable environments, their reliance on expert knowledge and predefined conditions limits their applicability in more complex and unpredictable real-world scenarios.

MARL approaches have emerged as a promising solution to address these coordination challenges within HMRS [7]. The core advantage of MARL is their intrinsic capability to learn from interaction with the environment, thereby facilitating the adaptive development of cooperative strategies among robots [8] in response to changing team compositions. A pivotal advancement in MARL domain is the integration of transformer-based architectures [9], [10] into MARL frameworks, marking a significant stride towards sophisticated agent cooperation. These architectures recast the multi-agent decision making as a sequential decision problem, where the decision of each agent is coupled with the preceding agents. This sequential decision-making framework is inherently conducive to fostering group coordination, presenting a natural fit for managing complex interactions within HMRS.

However, despite these advancements, standard multi-agent transformer models face notable limitations, especially in accommodating team composition changes. These models often fail to seamlessly integrate teams with different coalitions into a coherent learning framework, thereby hindering their generalization capability when facing novel team compositions. The root of this challenge lies in the model treats each new team composition as a new task that necessitates retraining. In addition, the attention mechanism involved indeed can train on the varying number of agents simultaneously, but it needs to set a maximum number of agents in team, unable to work when robot team number exceeding the range. Moreover, they often fail to retain heterogeneous-specific knowledge essential for generalization, a critical element for navigating across different unseen team compositions inherent in heterogeneous settings.

This paper builds on a novel transformer-based MARL

This work was supported in part by the Agency for Science, Technology and Research (A*STAR), Singapore, under the MTC Individual Research Grant (M22K2c0079) and the Robotics Horizontal Technology Coordinating Office Grant (C221518004), the ANR-NRF Joint Grant (No.NRF2021-NRF-ANR003 HM Science), and the Ministry of Education (MOE), Singapore, under the Tier 2 Grant (MOE-T2EP50222-0002).

[1]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore {caiy0039, xiangkun.he, lyuchen}@ntu.edu.sg

[2]Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (ASTAR), Singapore {stucaiy, guo_hongliang, wyyau}@i2r.a-star.edu.sg

*Corresponding author

framework, specifically tailored to address the combinatorial generalization challenges of coordinating HMRS in dynamic team compositions, including variations in team size and robots with novel capabilities. By exploiting the transformer architecture's ability to process inter-agents relationships and its scalability, our framework further provides a novel solution for generalizing across diverse and previously unseen team compositions. Our approach leverages graph neural networks and self-attention mechanisms to effectively capture the heterogeneous inter-agent interaction features within robot team. This enables our trained policy to flexibly adjust to team size alterations. Moreover, we introduce a capability-oriented decoder that autoregressively outputs agents' actions. It features a shared initial processing for all agents, followed by a decentralized decision-making module tailored to each robot's heterogeneity type and capabilities. Such a design not only facilitates efficient common knowledge sharing among heterogeneous agents but also preserves heterogeneity-specific information.

Our contributions can be summarized as follows:

- We present a transformer-based MARL framework equipped with a graph-based representation encoder to handle team size changes.
- We design a decentralized capability-oriented decoder to output agents' actions in an auto-regressive manner, inherently reserve heterogeneous-specific knowledge essential for capability generalization.
- We evaluate our method's generalization capabilities on two heterogeneous multi-robot tasks in both simulated and physical platforms. The results demonstrate that our approach significantly enhances the combinatorial generalization capabilities, outperforms traditional MARL methods in terms of task-specific metrics.

II. Related Works

In this section, we will introduce the intricate interplay between cooperative HMRS, the ensuing combinatorial generalization challenges it presents, and the pivotal role of MARL as our methodology to address these complexities.

A. Heterogeneity in Cooperative Multi-Robot Systems

Cooperative HMRS (C-HMRS) is a class of multi-robot systems that designed to coordinate a team of robots to work together to achieve a common goal. Most real-world multi-robot problems require a diverse set of capabilities aggregated with heterogeneous robots [11], [12]. These robots have different diverse functionalities , from mobility to sensing and communication aspects [13]. Therefore, the team should be able to coordinate effectively, leverage each robot's own strengths, to achieve the mission objectives. This presents significant challenges in designing a generalized policy capable of accommodating the diverse behavioral patterns of heterogeneous agents as team compositions change.

B. Combinatorial Generalization for C-HMRS

C-HMRS Tasks in real-world scenarios often involve different team configurations. The optimal team control policies varies according to the team composition, i.e. the size of the team and the capabilities of individual robots. [14] formally address these variations in team configuration as the Combinatorial Generalization (CG) problem. In such contexts, retraining agents in every new team configuration is computationally onerous. Therefore, it is desirable that the model can generalize "zero-shot" to unfamiliar team compositions, which it has not previously seen or cooperated during the training phase.

Previous work [15] has demonstrated that Graph Neural Network (GNN) policies possess promising transferability for CG problems within homogeneous robot team, attributed to their capability to manage a dynamic number of nodes. However, extending this generalization to heterogeneous settings often poses challenges, as noted in [14]. [16] introduced a meta-reinforcement learning approach for facilitating collaboration among different robot types. While promising, this method is constrained to a fixed number of robots and does not adapt to changes in the capabilities of individual robots. In addition, [17] tackles the challenges of combinatorial generalization, striving to learn a team policy that generalizes across different team compositions. Yet, their approach, the coach-player framework, relies on the coach having an omniscient view of the environment, which may not be feasible in real-world settings.

C. Multi-Agent Reinforcement Learning

Cooperative MARL finds wide-ranging applications in multi-robot systems, encompassing areas such as multi-robot path planning [18], coverage strategies [19], and task allocation mechanisms [20]. The integration of recent advancements in deep learning into MARL frameworks presents significant opportunities to overcome the complexities associated with traditional control paradigms in multi-robot systems [7]. Notably, among popular multi-robot testbed environments under the MARL framework, such as the 2D task-oriented multi-agent particle environments (MPE) [21] and MultiRoboLearn [22], it is MARBLER [23] that uniquely facilitates realistic simulations critical for Sim2Real evaluation. Algorithms such as QMIX [24], MADDPG [21], and MAPPO [25] exemplify the application of this framework. However, these methodologies generally rest on a critical assumption: the composition of agent team is both homogeneous and fixed number of agents. In an effort to mitigate this constraint, some methods have adopted the attention mechanism [26] to accommodate a varying number of agents [27], [28]. Yet, these approaches tend to be effective only within a predefined range of team size and lack the ability to generalize to scenarios outside of that range.

Some MARL strategies recognize the importance of team heterogeneity through diverse behaviors. A prevalent technique, known as behavioral typing [11], involves augmenting each agent's observations with a unique identifier to promote behavior differentiation [29]. Although this approach has demonstrated considerable success, recent research has highlighted issues concerning scalability [30] and generalize to new agents [31]. In addition, this method frequently

necessitate retraining when introducing agents with new capabilities, as they do not sufficiently address the adaptability to changes in the capabilities of each heterogeneous robot. Multi-Agent Transformer (MAT) [9] represents a significant leap in MARL, embodying practical application of the multi-agent advantage function decomposition theorem [32]. This development ensures monotonic improvement and convergence to Nash equilibrium in multi-agent cooperative tasks by modeling multi-agent systems (MAS) as sequential decision problems from a microscopic time perspective. However, MAT does not explicitly tackle the CG challenge, particularly in adapting to changes in team size and robot capabilities within heterogeneous multi-robot teams, nor does it maintain independent knowledge of each robot type—key for generalizing across varied, unseen team compositions. Our work builds on MAT's foundation, aiming to extend its applicability and enhance CG capabilities in C-HMRS.

III. Methodology

A. Problem Formulation

Modeling C-HMRS as a Graph: We represent team of N heterogeneous robots as a directed edge-featured graph $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ represents the nodes corresponding to the robots in a 2D environment, and $E \subseteq V \times V$ denotes the set of directed communication links between robot pairs. For any edge $e_{i,j} = (v_i, v_j) \in E$, it signifies a communication link from robot j to robot i. We assume full communication among all robots. Each robot i is associated with an observation o_i that includes its capabilities, heterogeneity type, and environmental sensor readings. The capabilities of robot i are represented by a real-valued vector $c_i \in \mathcal{C} \subseteq \mathbb{R}_+^d$, where $\mathcal{C}$ defines the d-dimensional capability space. Heterogeneity type is denoted by $I_i \in I$, where I represents all possible heterogeneity types, indicating each robot's role within the team.

Modeling C-HMRS as MARL: We formulate C-HMRS as a MARL problem within the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) framework [33], extending it to include robot capabilities [31]. The problem is defined by a set of elements $(N, O, A, \mathcal{C}, P, R, S, \gamma, \pi)$, where N is the number of robots, O integrates the local observations of each robot into a joint observation space, expressed as $O = \prod_{i=1}^{N} O^i$. The joint state and action spaces are denoted by S and A, respectively. $\mathcal{C}$ represents the space of multi-dimensional capabilities. The state transition dynamics are captured by P, while $R : O \times A \to \mathbb{R}$ specifies the shared reward. The discount factor $\gamma \in [0, 1]$, balances immediate and future rewards. Finally, π specifies the team's joint policy, which produces each agent's action based on the current state. At each time step t, each agent $i \in \{1, \ldots, N\}$ receives observations o_i, chooses actions a_i based on its policy π^i, leading to joint observation $\mathbf{o}$ and joint actions $\mathbf{a}$ that yield shared rewards $R(\mathbf{o}, \mathbf{a})$ and transition to the next state according to $P(s'|s, \mathbf{a})$. The collective aim is to optimize a joint policy π^* that maximizes the expected cumulative reward $G = \sum_{t=0}^{L} \gamma^t R_t$, over a task horizon of L time steps.

In our study, we evaluate the model's ability to generalize to new team compositions under Dec-POMDP. Specifically, we construct a training configuration set U_{train} consisting of predetermined team compositions, characterized by specific team sizes N and capabilities C. This setup facilitates the model's learning phase. Subsequently, we assess the trained model's performance on a test configuration set U_{test}, which introduces previously unseen team compositions, marked by new $\hat{N}$ and $\hat{C}$. Our goal is to craft a generalized joint policy π^* during the training phase, capable of effectively adapting to novel team size and capability distribution, and maximize the mean discounted return. Training and generalization phases are illustrated in Fig. 1.

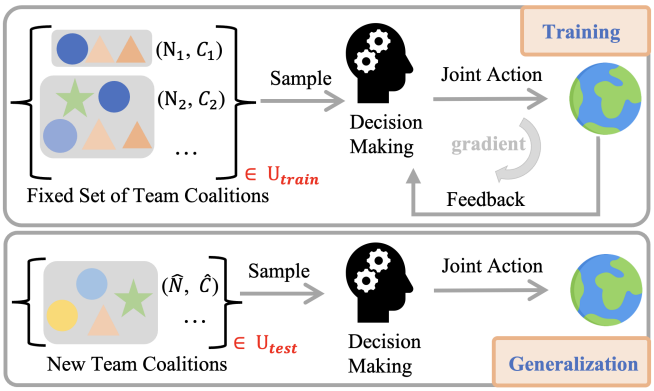

Fig. 1: During the training phase, teams characterized by $(N, \mathcal{C})$ are sampled consistently from a predefined set of team compositions within the U_{train} configuration set to train the policy via gradient updates. Each grey area encompasses a composition, identical shapes signify robots with the same heterogeneity type, while varying colors denote robots with different capabilities within that type. In the generalization phase, the policy, honed through training, is tested against new team compositions denoted by $(\hat{N}, \hat{\mathcal{C}})$, which are drawn from the U_{test} configuration set.

B. Network Structure

This subsection presents the architecture of the Capability-Oriented Multi-Agent Transformer (COMAT) model, which is delineated into two principal components: an encoder and a decoder, to construct a dynamic mapping from agents' observation inputs to their action outputs. The encoder transforms the joint observations of all agents $(o^1, \ldots, o^N)$ into a cohesive latent representation $(\hat{o}_t^1, \ldots, \hat{o}_t^N)$. This representation has a twofold function: it is utilized for estimating the state values $(\hat{V}^1, \ldots, \hat{V}^N)$, and it provides the input for decoder's sequential decision-making. The decoder, similar in [9], operates in an auto-regressive manner, using the latent representations and previously determined agents' actions to generate the next agent's action, culminating in the action sequence $(a^1, \ldots, a^N)$ for this current time step. The overall model ensures that the determination of each agent's action is contingent upon the preceding actions of agents, thus intrinsically facilitating coordination within the team. This structure is visually depicted in Fig. 2.

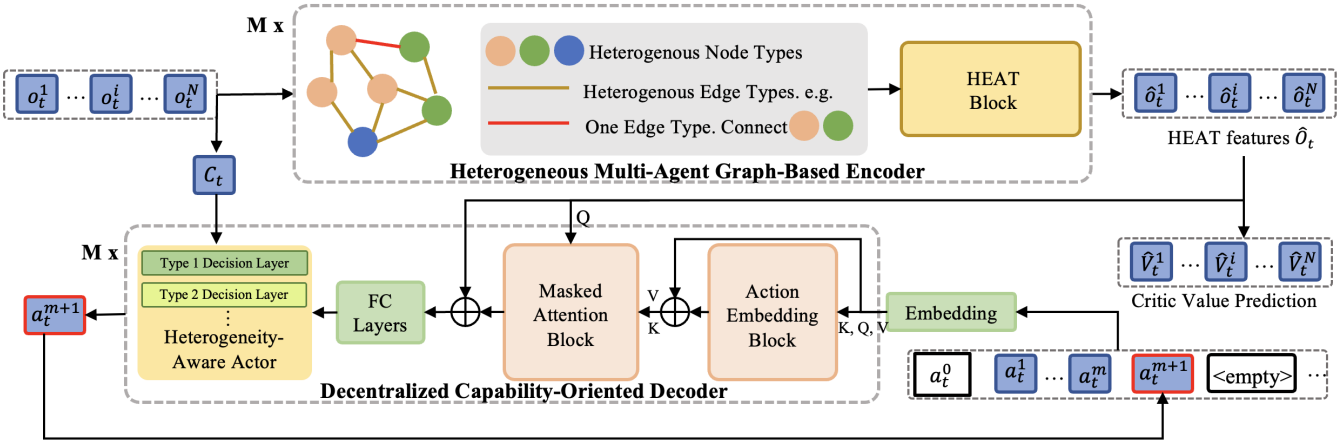

Fig. 2: Overview of the COMAT network architecture. The encoder utilizes a graph-based feature extractor to process the agents joint observations, yielding HEAT features. These features inform the critic value prediction and feed into the decentralized, capability-oriented decoder. The decoder sequentially ingests the HEAT features along with previous agents' actions to autoregressively determine the subsequent action for each agent at the current time step.

The Encoder: COMAT employs the Heterogeneous Edge-enhanced Graph ATtention network (HEAT) [34], an extension of the Graph Attention Network (GAT) [35], to capture the nuanced interactions among heterogeneous agents within team. By representing the team as a directed, edge-featured heterogeneous graph, HEAT effectively handles the varying types of agents and their interactions, which are critical in capturing the heterogeneity and dynamics within the team. In details, we utilize a series of HEAT blocks, enables the model to enrich and distill complex inter-agent relationships into interaction features $(\hat{o}_t^1, \ldots, \hat{o}_t^N)$, facilitating the subsequent decision-making process:

$$\hat{O}_t = \{\hat{o}_t^1, \ldots, \hat{o}_t^N\} = \text{HEAT}_{\text{enc}}^{\phi}(E_t, V_t), \tag{1}$$

where $\text{HEAT}_{\text{enc}}^{\phi}$ symbolizes the encoding procedure, coalescing agent interactions and heterogeneity into an integrated representation for C-HMRS.

The HEAT Block: Within the HEAT block, the initial input $(o_t^1, \ldots, o_t^N)$ is restructured into node features $V_{feature}$ and edge features $E_{feature}$, alongside their respective types V_{type} and E_{type}. Node features encapsulate dynamic agent attributes such as location, velocity and sensor data, while node types convey the heterogeneity of agents, including task-oriented roles and capabilities denoted by c. Edge features encode interaction metrics like relative distances and shared information through communication, with edge types marking the direction of interaction according to agent types.

Given an edge from node j to i, represented as $e_{ij} = [e_{ij}^{\text{feature}} \| e_{ij}^{\text{type}}]$, the fusion of edge features and types alongside node data is denoted by $\tilde{\mathbf{f}}_{ij} = [e_{ij} \| \mathbf{V}_j]$. This vector characterizes node j's information from the viewpoint of node i. Leveraging the GAT's architecture, the concatenated vectors $\tilde{\mathbf{f}}$ are shared among nodes via a K-head attention mechanism, which enrich the representation of node relationships and enhance training stability. For node i, the attention coefficient α_{ij}^k dictates the significance of $\tilde{\mathbf{f}}_{ij}$ on $V_{feature}^i$ within the k-th attention head, computed as:

$$\alpha_{ij}^k = \frac{\exp(\text{LeakyReLU}(\hat{\mathbf{a}}[V_{feature}^i \| \tilde{\mathbf{f}}_{ij}]))}{\sum_{s \in N} \exp(\text{LeakyReLU}(\hat{\mathbf{a}}[V_{feature}^i \| \tilde{\mathbf{f}}_{is}]))}, \tag{2}$$

where $\hat{\mathbf{a}}$ is a single-layer feed-forward neural network, followed by LeakyReLU non-linearity and softmax normalization. The updated feature for node i at time t, $\hat{o}_t^i$, is then derived from the attentively weighted summation of features from all heads and neighbors:

$$\hat{o}_t^i = \frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N} \alpha_{ij}^k W_h [e_{ij} \| V_{feature}^j]. \tag{3}$$

This culminates in a rich node feature captured inter-agent interactions and heterogeneity, ready for further subsequent steps. The critic value $\hat{V}_t^i$ is then estimated from $\hat{o}_t^i$, with multi-layer perceptron (MLP) is established as the critic network to estimate the value function, i.e., $\hat{V}_t^i = \text{mlp}(\hat{o}_t^i)$ which is used to guide the policy learning process.

The Decoder: Our decoder takes the idea that agents make decisions sequentially [9]. Given the latent representation $(\hat{o}_t^1, \ldots, \hat{o}_t^N)$ at time t from the encoder and the sequence of preceding actions $(a_t^0, a_t^1, ..., a_t^{m-1})$, the decoder aims to generate the current agent m's action a_t^m, setting the stage for subsequent decisions. This auto-regressive mechanism ensures that each agent's decision is informed by prior teammates' actions, fostering effective team coordination for maximal reward optimization.

The decoding process begins with a_t^0, represented by a zero vector that denotes the start. The sequence of prior actions up to agent $m-1$ is fed through an action embedding block, comprising a K-head attention module and two fully connected layers, parameterized by θ^1, which yields the action embedding. Subsequently, these embedding and the latent representations $(\hat{o}_t^1, \ldots, \hat{o}_t^N)$ are input into a K-head attention module parameterized by θ^2 that operates with masking to exclude latent representation beyond agent m.

This procedure results in an aggregated feature set $h_t^{0:m}$, tailored for decision-making. The attention mechanism restricts the query to $\hat{o}_t^{0:m}$, while the keys and values are derived from the actions $\hat{a_t}^{0:m-1}$:

$$\hat{a_t}^{0:m-1} = f_{\theta^1}(a_t^{0:m-1}), \quad 1 \leq m \leq N, \tag{4}$$

$$h_t^{0:m} = f_{\theta^2}(\hat{a_t}^{0:m-1} | \hat{o}_t^{0:m}), \quad 1 \leq m \leq N. \tag{5}$$

This process uniquely tailors each agent's decision to the cumulative actions of its predecessors, intuitively sharing of common knowledge among heterogeneous agents.

In the last of the decoding process, we introduce a decentralized decision layer, stratified according to agent type, which operates as a set of decentralized actors. This design is conducive to individualized decision-making, allowing for the distinctive attributes and roles of each agent type to be reflected in the decisions made. Inputs to this stage encompass the composite representation $h_t^{0:m}$, coupled with the capability vector $\mathcal{C}_m$, which encapsulates the particular capabilities and the role an agent plays in achieving the collective goals of the team. This allows for an advanced integration of global situational awareness and individual agent capabilities. By adopting this approach, the model adeptly manages to balance between the efficient sharing of common knowledge across all agents and the preservation of heterogeneity-specific information, underpinning capability generalization.

C. Experiment Design

We conduct detailed experiments on [23], an open framework which enables hardware experimentation in the Robotarium [36], a well-established multi-robot test bed.

Environments: We utilize following two C-HMRS tasks:

- *Heterogeneous Material Transport (HMT):* A team of robots are tasked with transporting two zones of materials to designated location. The team is comprised of two types of robots: N fast robots and M slow robots. And the robots capability refers to their respective speed and payload capacity. Faster robots can transport less payload. For each episode, the initial location for all robots are randomly initialized within the purple area, and two zone's material amount are set according to robots number and capabilities. Locations and remaining material amount are appended to the observation of each robot. The team will be penalized for collision and time goes, and rewarded for material unloaded.
- *Predator Capture Prey (PCP):* A team of heterogeneous robots are tasked with capturing K prey robots. The team is comprised of two types of robots: N sensing robots and M capturing robots. All robots starting location are randomly initialized. Both sensing and capturing robots have there location appended to their observation. The capability of robots refers to their respective sensing range and capturing range, which is also appended to their local observation. If the prey is within the sensing range of the sensing robot, the location of the prey is appended to the observation of the sensing robot. Capture robots will never be know the location of the prey, unless they cooperate with the sensing robots. The team will be rewarded if capture robots are within the capturing range of the prey.

Both environments are designed to terminate either successfully complete the task or exceed the maximum time limit. A visualization of the two environments is shown in Fig. 3.

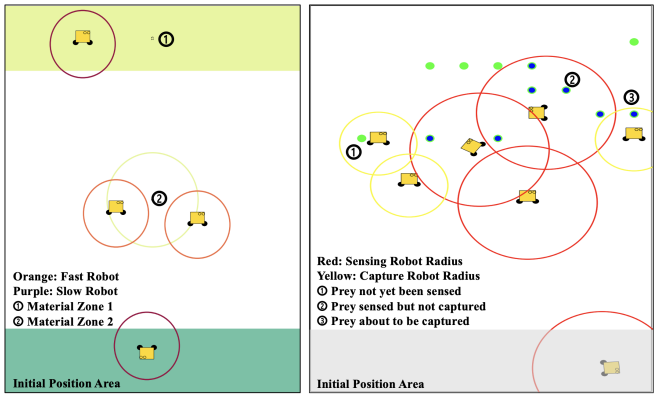

Fig. 3: LEFT is HMT scenario: The yellow areas represent the two zones of materials, the orange and purple circles represent the robots with different speed and payload; RIGHT is the PCP scenario: The grey area shows the starting location of the robots, The red and yellow circles represent corresponds robot type with sense radius, the green dots represent prey robots, point 1, 2, 3 indicates different prey state.

D. Training Procedure

Combinatorial Generalization Design: We aim to evaluate the CG capabilities of our model by examining its performance across varied team compositions, focusing on two key aspects: team size and capability changes. For each experimental environment, we employ a systematic approach to sample a fixed set of team compositions from a range of team size and capability configurations for training purposes.

For the training phase in both environments, we sample a total of 20 coalitions. The team sizes are randomly chosen from the set $\{3, 5, 7\}$, enabling an examination of the model's adaptability to various team sizes. In the PCP environment, the capabilities, i.e. the effective range, of both sensing and capture robots are established by sampling from uniform distributions. Specifically, sensing robots' capabilities are sampled from $U(0.3, 0.55)$, and capture robots' capabilities are sampled from $U(0.35, 0.45)$. In MT scenario, the capabilities, i.e. speed and payload, for both fast and slow robots are determined through a similar sampling process from uniform distributions. While speed and payload capacity are inversely related, the speed of faster robots is sampled from $U(0.19, 0.45)$, with their payload capacity ranging from $U(5, 12)$. Conversely, the speed of slower robots is sampled from $U(0.10, 0.15)$, with their payload capacity extending from $U(16, 22)$. This deliberate sampling strategy is designed to expose the model to a wide array of

team compositions during training, thereby enhancing its CG capabilities within C-HMRS.

In the generalization phase, we challenge our model with novel team compositions that were not part of the training set. For each environment, a total of 20 new coalitions are created for testing purposes. This phase introduces a broader and different range of team sizes by sampling the number of robots from the set $\{2, 4, 6, 8\}$. This selection strategy ensures that the model is tested against both smaller and larger teams than those encountered during the training phase. For the PCP, the capabilities assigned to both sensing and capturing robots are modified to $U(0.2, 0.4)$ for sensing radius and $U(0.25, 0.4)$ for capturing radius, reflecting a different range of potential abilities. Similarly, in HMT environment, the capabilities, i,e. speed and payload are adjusted. The speed and payload for faster and slow robots are sampled from $[U(0.15, 0.35),\ U(7, 15)]$ and $[U(0.08, 0.18),\ U(13, 28)]$ respectively. These adjustments in the sampling parameters for both environments are crucial for conducting a thorough evaluation of the model's ability to generalize across different team compositions with team size and capability changes.

Loss Function: The encoder network, characterized by parameter ϕ, aims to learn effective joint observation representations for value approximation during the training phase. The objective is to minimize the empirical Bellman error, described by

$$L_{\text{Encoder}}(\phi) = \frac{1}{BN}\sum_{n=1}^{N}\sum_{i=1}^{B}\Big[R(\mathbf{o}_i, \mathbf{a}_i) + \gamma V_{\overline{\phi}}(\mathbf{o}_i'^{(n)}) - V_{\phi}(\mathbf{o}_i^{(n)})\Big]^2, \tag{6}$$

where $\mathbf{o}_i'$ represents the joint observation at the next timestep. $V_{\overline{\phi}}$ is the target network's parameters for the critic, as referenced in [37]. This target network is non-differentiable and updated every few epochs. The discount factor is denoted by γ, B is the batch size, and N is the number of agents. The decoder network, with parameters denoted by θ, is trained concurrently to minimize the PPO-based loss [25]:

$$L_{\text{Decoder}}(\theta) = \left[\frac{1}{BN}\sum_{n=1}^{N}\sum_{i=1}^{B}\min\left(r_i^{(n)}(\theta)\hat{A}_i^{(n)}, \text{clip}\left(r_i^{(n)}(\theta), 1\pm\epsilon\right)\hat{A}_i^{(n)}\right)\right] + \sigma\left[\frac{1}{BN}\sum_{i=1}^{B}\sum_{n=1}^{N}S[\pi_\theta(\hat{\mathbf{o}}_i^{(n)})]\right], \tag{7}$$

where $r_i^{(n)}(\theta)$ is the importance sampling term, indicating the probability ratio between the new and old policy:

$$r_i^{(n)}(\theta) = \frac{\pi_\theta(\mathbf{a}_i^{(n)}|\mathbf{o}_i^{(n)}, \mathbf{a}_i^{(n-1)})}{\pi_{\theta_{old}}(\mathbf{a}_i^{(n)}|\mathbf{o}_i^{(n)}, \mathbf{a}_i^{(n-1)})}. \tag{8}$$

The clip function is used to prevent the policy from changing too much at each update. $\hat{A}_i^{(n)}$ is joint advantage estimate of batch i and agent n. One can take $\hat{V}_t = \frac{1}{N}\sum_{n=1}^{N}V_\phi(o_t^n)$ as an estimate of the current joint value function and obtain temporal difference error: $\delta_t = R_t + \gamma\hat{V}_{t+1} - \hat{V}_t$. Then the joint advantage estimate $\hat{A}_t$ can be determined using Generalized Advantage Estimation (GAE) [38]:

$$\hat{A}_t = \sum_{l=0}^{T-t}(\gamma\lambda)^l\delta_{t+l}, \tag{9}$$

where λ is the GAE parameter. T is the episode length. The entropy term in the decoder loss, weighted by the coefficient σ, is used to promote policy exploration.

Evaluation Details: For our experiments, we train each model for $4,000,000$ timesteps, utilizing a batch size of 32 and three distinct random seeds for each policy. This approach aims to underscore the generalization capabilities of our model. We investigate the effects of integrating COMAT with established baselines like QMIX, MAPPO, and Vanilla MAT for both scenarios. For those baselines that are limited to fix number of agents, they are trained with the maximum number eight of agents necessarily, and dead masked the action of redundant agents without affecting the policy update. Similarly, each agent's observation is augmented with capabilities vector $\mathcal{C}$ and heterogeneity type I. Parameter sharing is applied to all baselines for same type of agents to ensure a fair comparison. All algorithms are trained on 16 core 32 threads 3.3 GHz CPU, 32 GB RAM, NVIDIA RTX4090 and 64-bit Ubuntu system.

We assess the models performance based on several key metrics: average return, average episode length, and task-specific indicators such as completion rate and collision times. These evaluations are conducted on generalization team compositions set, designed to challenge the models with Out-of-Distribution team capabilities and variable team sizes, thereby testing their CG efficacy.

IV. RESULTS AND DISCUSSION

A. Performance on Training Set

This section presents the results of the training phase for both PCP and HMT scenarios. In Figs. 4, we present the

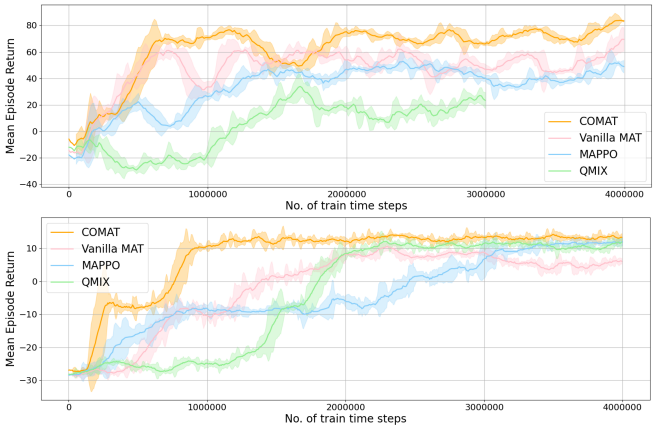

Fig. 4: Mean episode return of different algorithms for both environments under training configuration set U_{train}. Top: PCP task. Bottom: HMT task.

mean episode returns across time steps for various trained

policies within the PCP and HMT training team compositions, respectively. Initially, both MAPPO and QMIX exhibit fluctuations in return, due to the varying training team compositions upon environment resets, posing challenges for critic in learning accurate value functions effectively within heterogeneous teams. In contrast, COMAT demonstrates superior adaptability by recognizing the heterogeneity of agents, thereby facilitating more stable coordination based on individual capabilities. Notably, COMAT and Vanilla MAT achieve similar convergence times, surpassing the QMIX and MAPPO baselines in efficiency.

Table I shows more detailed results of the training phase with other task-specific metrics for both tasks. Collision refers to the average robot collides times with others in each episode, and completion refers to the percentage of episodes that all prey are captured or all materials are delivered within the time limit. The performance disparity between COMAT and MAT baseline underscores the efficacy of integrated graph-based encoder and capability-oriented decentralized decoder. This configuration proves particularly adept at managing cooperation in teams with varying compositions.

TABLE I: Detailed metrics for train team compositions.

Task	Metric	MAPPO	QMIX	MAT	COMAT
PCP	Mean Return	45.43	24.10	70.12	**82.47**
PCP	Episode Steps	79.00	79.70	70.95	**65.61**
PCP	Completion	87.9%	85.6%	91.6%	**92.8%**
PCP	Collision	0.00	0.00	0.00	0.00
HMT	Mean Return	11.70	11.70	8.32	**12.41**
HMT	Episode Steps	70.00	70.60	72.30	68.75
HMT	Completion	97.7%	97.2%	91.1%	97.7%
HMT	Collision	0.00	0.00	1.00	0.00

B. Performance on Generalization Set

In the generalization phase, we evaluated the trained policies against new team compositions sampled from the test configuration set U_{test}. These compositions featured varied capability distributions and team sizes across both scenarios. Fig. 5 further elucidates the mean episode return of different algorithms within these generalization coalitions. The findings underscore COMAT's superior performance over baseline algorithms during the generalization phase. Notably, COMAT demonstrates a significantly reduced performance disparity between the training and testing phases. This is highlighted by the bar height difference between the red and orange series. Specifically, COMAT policy retained 88% of their performance in the PCP scenario and 95% in the HMT scenario, underscoring their CG ability to generalize effectively to novel team compositions.

C. Real-world Experiments

Fig. 6 shows the real-world experiments of COMAT policy in PCP and HMT scenarios. When the team size is four, which is selected from the test configuration set U_{test}, COMAT is able to effectively coordinate based on each agents' capability and complete the task. More details are available in the supplementary video.

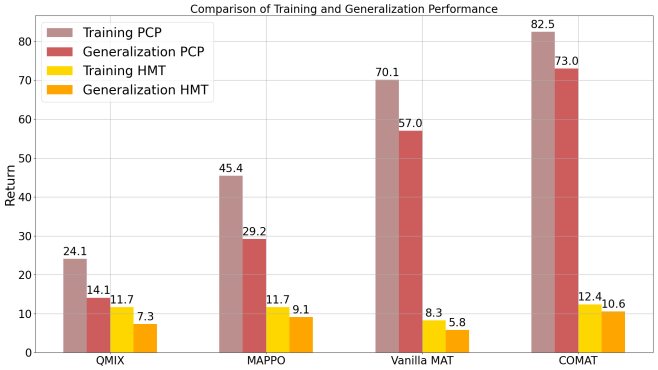

Fig. 5: Training and generalization return comparison of different algorithms in both scenarios.

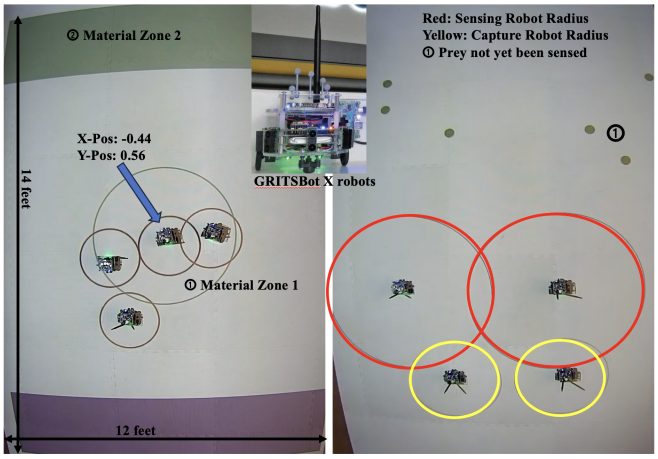

Fig. 6: LEFT is the real-world experiment of HMT scenario: All materials in zone 2 finished unload at timestep 43, left with all four robots gather in zone 1 to unload. RIGHT is the real-world experiment of PCP scenario: Two sensing robots and two capture robots (four robots) are searching for the prey illustrated in green dots. All preys' status remain unsensed and uncaptured at timestep five.

V. CONCLUSION AND FUTURE WORK

In conclusion, our study introduces COMAT, a method leverages transformer architecture, designed to address the challenges of generalizing learned policies to varying heterogeneous multi-robot team compositions. By incorporating graph-based representation extractor alongside decentralized, capability-oriented decoder, COMAT has distinctly showcased its proficiency in adapting to variations in team sizes and individual robot capabilities. The model's effectiveness was validated through rigorous evaluations on two heterogeneous multi-robot tasks in both simulated and physical environments, where COMAT consistently outperformed existing MARL approaches according to task-specific metrics.
Notwithstanding its contributions, the current implementation of COMAT encounters certain limitations, notably the prerequisite of quantifiable capability vectors and its constrained generalization to scenarios introducing new types of heterogeneity. Future work will aim to address these chal-

lenges by enhancing the framework's generalization capabilities to accommodate heterogeneous multi-robot teams with implicitly defined capabilities. An envisioned improvement includes augmenting each robot's decision-making process with predictive insights into their teammates decisions.

REFERENCES

[1] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy *et al.*, "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60–71, 2013.

[2] H. Guo, Z. Liu, R. Shi, W.-Y. Yau, and D. Rus, "Cross-entropy regularized policy gradient for multirobot nonadversarial moving target search," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2569–2584, 2023.

[3] E. R. Boroson and N. Ayanian, "3d keypoint repeatability for heterogeneous multi-robot slam," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6337–6343.

[4] L. Zhang, Y. Sun, A. Barth, and O. Ma, "Decentralized control of multi-robot system in cooperative object transportation using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 184 109–184 119, 2020.

[5] Y. Zhou, H. Hu, Y. Liu, S.-W. Lin, and Z. Ding, "A distributed approach to robust control of multi-robot systems," *Automatica*, vol. 98, pp. 1–13, 2018.

[6] E. Jensen, M. Franklin, S. Lahr, and M. Gini, "Sustainable multi-robot patrol of an open polyline," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4792–4797.

[7] M. Hüttenrauch, A. Šošić, and G. Neumann, "Deep reinforcement learning for swarm systems," *Journal of Machine Learning Research*, vol. 20, no. 54, pp. 1–31, 2019.

[8] R. S. Sutton, A. G. Barto *et al.*, "Introduction to reinforcement learning. vol. 135," 1998.

[9] M. Wen, J. Kuba, R. Lin, W. Zhang, Y. Wen, J. Wang, and Y. Yang, "Multi-agent reinforcement learning is a sequence modeling problem," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 509–16 521, 2022.

[10] D. Wang, F. Zhong, M. Wen, M. Li, Y. Peng, T. Li, and Y. Yang, "Romat: Role-based multi-agent transformer for generalizable heterogeneous cooperation," *Neural Networks*, p. 106129, 2024.

[11] M. Bettini, A. Shankar, and A. Prorok, "Heterogeneous multi-robot reinforcement learning," *arXiv preprint arXiv:2301.07137*, 2023.

[12] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–31, 2019.

[13] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo, "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 943–967, 2007.

[14] A. Mahajan, M. Samvelyan, T. Gupta, B. Ellis, M. Sun, T. Rocktäschel, and S. Whiteson, "Generalization in cooperative multi-agent systems," *arXiv preprint arXiv:2202.00104*, 2022.

[15] A. Agarwal, S. Kumar, and K. Sycara, "Learning transferable cooperative behavior in multi-agent teams," *arXiv preprint arXiv:1906.01202*, 2019.

[16] H. Jia, Y. Zhao, Y. Zhai, B. Ding, H. Wang, and Q. Wu, "Crmrl: Collaborative relationship meta reinforcement learning for effectively adapting to type changes in multi-robotic system," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 362–11 369, 2022.

[17] B. Liu, Q. Liu, P. Stone, A. Garg, Y. Zhu, and A. Anandkumar, "Coach-player multi-agent reinforcement learning for dynamic team composition," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6860–6870.

[18] W. Sheng, H. Guo, W.-Y. Yau, and Y. Zhou, "Pd-fac: Probability density factorized multi-agent distributional reinforcement learning for multi-robot reliable search," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8869–8876, 2022.

[19] X. Zhao, R. C. Fetecau, and M. Chen, "Efficient domain coverage for vehicles with second-order dynamics via multi-agent reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5614–5621.

[20] J. Orr and A. Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: a survey," *Sensors*, vol. 23, no. 7, p. 3625, 2023.

[21] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[22] J. Chen, F. Deng, Y. Gao, J. Hu, X. Guo, G. Liang, and T. L. Lam, "Multirobolearn: An open-source framework for multi-robot deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–6.

[23] R. J. Torbati, S. Lohiya, S. Singh, M. S. Nigam, and H. Ravichandar, "Marbler: An open platform for standardized evaluation of multi-robot reinforcement learning algorithms," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2023, pp. 57–63.

[24] 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," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7234–7284, 2020.

[25] 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," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] S. Hu, F. Zhu, X. Chang, and X. Liang, "Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers," *arXiv preprint arXiv:2101.08001*, 2021.

[28] J. Chai, W. Li, Y. Zhu, D. Zhao, Z. Ma, K. Sun, and J. Ding, "Unmas: Multiagent reinforcement learning for unshaped cooperative scenarios," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 4, pp. 2093–2104, 2021.

[29] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*. Springer, 2017, pp. 66–83.

[30] F. Christianos, G. Papoudakis, M. A. Rahman, and S. V. Albrecht, "Scaling multi-agent reinforcement learning with selective parameter sharing," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1989–1998.

[31] P. Howell, M. Rudolph, R. Torbati, K. Fu, and H. Ravichandar, "Generalization of heterogeneous multi-robot policies via awareness and communication of capabilities," *arXiv preprint arXiv:2401.13127*, 2024.

[32] J. G. Kuba, M. Wen, L. Meng, H. Zhang, D. Mguni, J. Wang, Y. Yang *et al.*, "Settling the variance of multi-agent policy gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 458–13 470, 2021.

[33] F. A. Oliehoek, C. Amato *et al.*, *A concise introduction to decentralized POMDPs*. Springer, 2016, vol. 1.

[34] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9554–9567, 2022.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[36] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[37] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[38] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.