2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11247013

LLM-Driven Hierarchical Planning: Long-horizon Task Allocation for Multi-Robot Systems in Cross-Regional Environments

Yachao Wang[1], Yangshuo Dong[1], Yunting Yang[1], Xiang Zhang[1],
Yinchuan Wang[1], Yuhan Wang[1], Chaoqun Wang[1]✉, and Max Q.-H. Meng[2]

***Abstract*— Long-horizon composite task planning for multi-robot systems in cross-regional complex scenarios faces dual challenges: spatial-semantic comprehension of natural language described tasks and collaborative optimization of subtask allocation. To address these challenges, this paper proposes a progressive three-stage task planning framework. First, an augmented scene graph is constructed to enable large language models (LLMs) to comprehend environmental structures, thereby generating simplified Linear Temporal Logic (LTL) task sequences. Subsequently, a novel heuristic function is employed to select optimal task allocation plans. Finally, LLMs are used to generate low-level executable robot instructions based on robotic system instruction templates. We establish a long-horizon composite task dataset for experimental validation on real-world quadrupedal multi-robot systems. Experimental results demonstrate the effectiveness of our approach in resolving cross-regional composite tasks.**

I. Introduction

In recent years, multi-robot systems have gained increasing adoption in large-scale scenarios such as intelligent warehousing [1] and disaster rescue operations [2]. Efficient task planning for multi-robot systems in cross-regional multi-room environments remains a critical challenge when handling long-horizon composite tasks. For instance, in cross-room material transportation scenarios, robotic systems must execute extended sequential instructions comprising dozens of subtasks. Such tasks present dual challenges for multi-robot coordination: cross-regional spatial-semantic understanding of long-horizon tasks and coordinated planning for extensive subtask allocation.

While existing multi-robot task planning methods demonstrate capability in complex environments [3], most traditional algorithmic exhibit limitations in holistic task comprehension when instructions are fully described in natural language [4]. Although data-driven collaborative strategies can optimize task allocation, their success rates degrade significantly with increasing subtask sequences due to the curse of dimensionality in state space representation. More critically, current methods generally overlook real-world spatial constraints during cross-regional operations. When subtasks contain spatial preconditions (e.g., "push box in room-126 against the wall" requires prior entry to room-126 as shown in Fig. 1), their reasoning performance deteriorates markedly. For natural language-described composite tasks, systems must simultaneously address spatial distribution relationships among subtasks, load balancing across robots, and crucially, infer implicit conditions embedded in linguistic descriptions. For example, interpreting the subtask "get the takeout" requires prior knowledge that takeout items may not be explicitly mapped but are associated with delivery points (Fig. 1), demanding adaptive target adjustment posing significant challenges for linguistic comprehension [5].

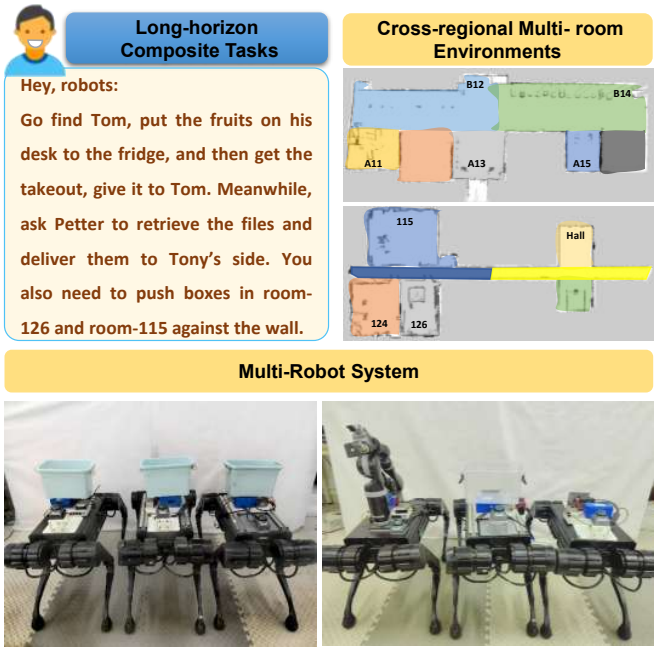


Fig. 1: Real-world experimental environments: cross-region outdoor scenario (top) and multi-room indoor setting (bottom); Exemplar of long-horizon composite tasks; Multi-robot systems validated in real-world experiments: homogeneous configuration (left) and heterogeneous configuration (right).

To address these challenges, we propose a progressive three-stage task planning framework for multi-robot systems executing natural language-described long-horizon tasks in cross-regional environments. Our method adopts a three-stage progressive architecture: Long-horizon Task Allocation, Optimal Plan Selection, and Low-Level Command Output. First, we perform spatial topology of complex cross-regional multi-room scenarios, extracting objects in the environment and constructing predicate relationships between

[1]School of Control Science and Engineering, Shandong University, China.
[2]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, China.
This work was supported by TaiShan Youth Scholar Scheme of Shandong Province and in part by the National Natural Science Foundation of China under Grant No. 62473232.

objects and connectivity relations between rooms. The generated spatial scene graph is then interpreted by large language models (LLMs), such as GPT-4 [6], Llama2 [7], in conjunction with the task description, yielding a sequence suitable for multi-robot allocation and optimal selection. Then, we translate these simplified sequences into Linear Temporal Logic (LTL) specifications, employing our novel heuristic function module to select optimal allocation strategies from multiple candidate solutions. Finally, we implement an LLM-powered instruction generation system that converts allocation plans into executable commands through customizable prompt templates, ensuring compatibility with diverse robotic platforms. Our principal contributions include:

- We develop an LLM-enhanced framework that simultaneously processes environmental scene graphs and linguistic task descriptions to generate optimized task sequences for multi-robot systems;
- We design a novel heuristic function module for selecting optimal LTL-based multi-robot task allocation strategies. Then, we convert allocation plans into executable commands using LLMs;
- We conduct experiments in a simulation environment and designed a benchmark dataset for long-sequence composite tasks in AI2-THOR [8]. We additionally conduct real-world experiments on a quadrupedal multi-robot system.

II. Related Work

A. Multi-Robot Task Planning

In existing research, significant progress has been made in task allocation for robotic task planning. [9] [10] have adopted task decomposition methods to achieve rational and efficient allocation planning for complex tasks. However, these approaches may suffer from excessive computational costs. [11] employs an average allocation strategy , achieving good performance in single-room environments. However, in complex multi-room and partitioned spaces, Euclidean distance calculations may not be accurate, leading to imbalanced path allocations. [12] proposed a method that incorporates graphs, Boolean formulas, and budget constraints to compute feasible paths within a given budget. But it is difficult to avoid high computational demands. The study in [13] introduces an urgency-based allocation strategy. But it struggles to handle complex tasks and environments. Research such as [14] [15] [16], using LTL (Linear Temporal Logic) and automata to constrain task planning ensures logical and sequential task execution. However, these methods often require a highly structured LTL framework or involve high computational costs. Additionally, studies like [17] explore the use of deep learning to address task allocation strategies. While effective, these approaches inevitably come with the drawback of high training costs. Our approach employs LLMs and augmented scene graphs for task planning and optimization, effectively handling complex environments without incurring significant costs.

B. LLMs for Robot

LLMs have seen growing adoption in robotics research, yet their outputs often lack stability. In [18], [19], LLMs are combined with scene graphs for task planning, but they struggle in complex environments and produce unstable results. In [20], a vision-language model collaborates with an LLM for flexible task planning; however, it relies heavily on precise perception and cannot fully mitigate instability.

In [21], [22], a Python-like language is used to provide input information, helping LLMs better interpret tasks while detecting real-time environmental conditions. This boosts feasibility but depends on accurate environmental sensing, limiting its applicability in complex, open settings. In [23], LLMs adjust task planning strategies via context recognition, enhancing execution accuracy yet still grappling with output reliability. In [22], restricted prompts reduce LLM load, and an exception-handling module corrects outputs to lessen "hallucinations," although complex scenarios remain challenging. [24] employs LLMs to generate a skill list from natural language instructions, constructs a directed acyclic graph based on skill dependencies, and applies linear programming to optimize task allocation. This effectively curbs "hallucinations" and enforces correct task order but remains limited in complex settings. Unlike previous efforts, we offer abundant spatial information to the LLM for multi-robot task allocation, thereby reducing over-reliance on LLMs. We also enforce strict templates to generate low-level execution commands.

III. Methodology

As shown in Fig. 2, our method consists of three main modules: `Long-horizon Task Allocation`, through the constructed augmented scene graph, we assist the LLMs in understanding the spatial distribution of the environment, performing task decomposition and assigning tasks to multiple robots, ultimately outputting a simplified representation of the LTL task sequence. `Optimal Plan Selection`, using the reward heuristic function we designed, the optimal task planning solution is selected from the LTL sequence. `Command Output`, by replacing different templates and utilizing the reasoning capabilities of the LLMs, the high-level plan is converted into low-level execution instructions for the robots. These modules form a closed-loop integration through a cascading data processing pipeline that bridges high-level semantic understanding with low-level robotic execution. We now provide a detailed exposition of each module's methodology, structured as follows:

A. Long-horizon Task Allocation

In this section, we present a methodology that integrates natural language described long-horizon composite tasks with augmented scene graphs to facilitate task allocation via LLMs. The scene graph augmentation aims to establish an environmental representation system capable of supporting complex spatial reasoning. LLMs rely not only on its pre-trained prior knowledge, but also on the spatial relationships of objects within the task when understanding the task.

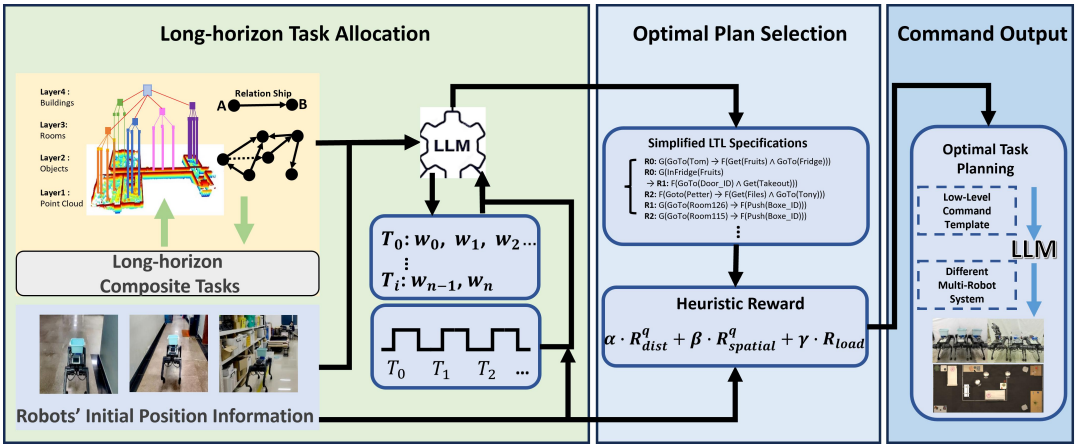


Fig. 2: System overview: firstly, construct scene graph with objects' relationship, input the topological information along with the task sequence into the LLM for task decomposition and allocation, and generate a series of simplified LTL task planning solutions for the multi-robot system; Secondly, reward calculations are performed for various task allocation sequences to select the optimal plan; Finally, depending on the multi-robot system, appropriate or predefined low-level instruction templates are selected to convert the task plan into executable instructions.

Following the methodology proposed in [25], we construct a foundational scene graph using environmental point clouds captured by 3D Lidar. We omitted the construction of the object mesh layer, thereby resulting in a four-layer scene graph. Subsequently, by employing a Scene Graph Generation (SGG) model [26] to generate inter-object predicate relationships, we enhance the object layer. (e.g., relationship: 'fruits on Tom's desk'), as illustrated in Fig. 2. In the following, we provide separate descriptions of the four-layer structure of the scene graph.

Point Cloud Layer: A 3D point cloud map constructed using Light Detection and Ranking Simultaneous Localization and Mapping (LiDAR SLAM), providing low-level navigational references for robotic systems.

Augmented Object Layer: Object instances extracted through object detection and semantic segmentation, combined with predicate relationships refined using the fast SGG model [26]. This layer encapsulates:

$$O_j = \{(x_j, y_j, z_j), \phi_j \ , \Psi_j\}, \quad \Psi_j = \{\psi_j^k\}_{k=1}^K,$$

$$\mathcal{P}_\mathcal{R} = \{P_i\}, \quad P_i = (O_j \rightarrow a_{jk} \rightarrow O_k),$$

in which O_j denotes the information contained in object j within the scene graph. $(x_j, y_j, z_j) \in \mathbb{R}^3$ denotes the 3D coordinate of the object. $\phi_j \in \mathbb{Z}^+$ represents the room ID to which object belongs. $\Psi_j = \{\psi_j^k\}_{k=1}^K$ encodes object attributes including: object category (e.g., cup, table), unique instance ID and robot interactivity flags (e.g., graspable).

$\mathcal{P}_\mathcal{R}$ denotes the set of inter-object relationships. P_i denotes the set of predicate relationship descriptors between objects. It describes the predicate relationship between object O_j and object O_k (e.g., "on", "next to").

Room Layer: The room layer showcases the connectivity relationships among adjacent rooms:

$$\mathcal{R}_\mathcal{M} = \{R_i\}, \quad R_i = \{\mathcal{O}_\mathcal{J}, E_i, \Psi_j\},$$

in which $\mathcal{O}_\mathcal{J} = \{O_j\}$ denotes the set of objects contained in R_i, $E_i = \{R_j, R_k, ...\}$ denotes the rooms which connected to R_i, Ψ_j denotes the room name attributes.

Building Layer: The top-level representation encapsulating cross-regional environmental features, including floor connectivity and structural landmarks.

Then, we employ LLMs to process natural language task descriptions and identify all required target object locations. Particularly, when task specifications implicitly describe target objects through spatial relationships rather than object name, our method localizes these objects by conducting top-down searches through hierarchical scene graphs while assigning unique identifiers to each precisely located entity in Fig. 2. For tasks involving complex spatial constraints, we implement an LLM-guided decomposition mechanism that breaks down the original task into two sequentially executable subtasks.

Every subtask is simplified such that each subtask sequence $T_k \in \mathcal{T} = \{T_1, T_2, ...T_i\}$ consists of k unique target points $w \in \mathcal{W} = \{w_1, w_2, ..., w_n\}$:

$$T_k = (w_1, w_2, ...w_k).$$

After obtaining a simplified and unambiguous task representation, we utilize LLMs to allocate these structured sequential tasks within multi-robot systems. This hierarchical approach is expected to enhance the interpretation of implicit spatial references while maintaining precise object-level localization through scene graph reasoning, thus aiming to bridge high-level task understanding with distributed robotic execution.

Prior to task allocation, the multi-robot system provides each agent's available capabilities along with randomized initial coordinates, which are comprehensively represented through a unified descriptor encompassing $A_i\{(x_0, y_0, z_0), C_i, \phi_0\}$, in which (x_0, y_0, z_0) represents the

robot's initial coordinates, C_i denotes the robot's available capabilities, and ϕ_0 indicates the robot's initial room affiliation. Crucially, the allocation process must rigorously address the strong temporal logic constraints between tasks and their potential concurrency requirements. To this end, we employ a simplified LTL formalism for task decomposition and assignment, generating optimized sub-task sequences for individual robots while ensuring temporal constraint satisfaction. We provide input instructions to the LLMs that integrate scene graph information and robot-specific data, requesting the LLMs to generate n candidate LTL sequence plan:

$$LTL_n = Prompt\{A_i\{(x_0, y_0, z_0), C_i, \phi_0\}, \mathcal{S}_\mathcal{G}\}, \tag{1}$$

in which $\mathcal{S}_\mathcal{G} = \{\mathcal{R}_\mathcal{M}, \mathcal{P}_\mathcal{R}\}$ denotes the scene graph of the whole environment. The results of the task allocation derived, as illustrated in Fig. 2, demonstrate the effective mapping of complex mission requirements to executable robot behaviors through this formal verification-based approach.

B. Optimal Plan Selection

Upon obtaining the LTL specifications, two heuristic approaches are employed for multi-robot task allocation scheme selection. The first leverages LLMs following the approach in [27] that originally applied LLM-based heuristics to single-robot system. We extend it through integration of multi-robot coordination information to enhance task planning decisions.

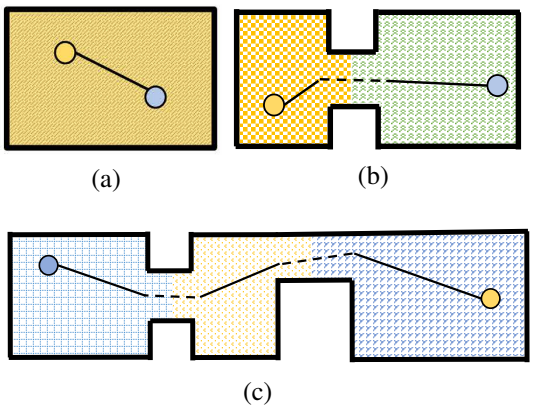


Fig. 3: The relationship between the rooms of adjacent target objects: (a) within the same room; (b) in different rooms but directly connected; (c) in different rooms and not directly connected.

The second approach is function-based heuristics. We have designed a task optimization heuristic suitable for multi-robot systems. The heuristic function integrates three types of rewards to select the optimal plan, which is described as follows:

1) Distance Reward: For each LTL planning scheme, the movement path assigned to robot R_q is denoted as $T_q = (w_1^q, \ldots, w_{K_q}^q)$. The normalized movement cost is the computed as an evaluation for the distance reward.

$$R_{\text{dist}}^q = 1 - \frac{\sum_{k=1}^{K_q-1} \|w_{k+1}^q - w_k^q\|_2}{D_{\max}}, \tag{2}$$

in which $D_{\max} = \max_{\pi\in\Pi} \sum_{k=1}^{K_q-1} \|p_{k+1} - p_k\|_2$ represents the maximum path length in the plan set. However, this distance reward only considers the straight-line distance between points and does not account for the spatial distribution. Therefore, we introduce the second reward named the spatial reward.

2) Spatial Reward: Based upon the **Room Layer** of the scene graph in Sec. A, we define three types of spatial relationships for objects executing sequential tasks in proximity, as shown in Fig. 3: (a) both objects are in the same room; (b) the objects are in different rooms, but the rooms are directly interconnected; (c) the objects are in different rooms and the rooms are not directly interconnected. In the third relationship, the connectivity between non-interconnected rooms in the scene graph allows for the calculation of the number of rooms through which the objects are connected, which impacts the computation of the spatial reward.

If the relationship between objects is (a), the distance reward R_{dist}^q remains unchanged. If the relationship between objects is (b) or (c), R_{dist}^q is updated to $R_{\text{dist}}^q \times \alpha(o_j, o_k)$, where α is a dynamic parameter. This parameter is related to the distance to the entrance of the room and the number of intervening rooms, and is defined as follows (assuming that o_j is located in r_m, o_k is located in r_n; d_m is the door of r_m, d_n is the door of r_n):

$$\alpha(o_j, o_k) = \begin{cases} 1 & \text{(a)} \\ 1 + \omega \frac{|\overrightarrow{o_j o_k} \times \overrightarrow{o_j d_m}|}{\|\overrightarrow{o_j o_k}\| \|o_j - o_k\|_2} & \text{(b)} \\ (b) \times (1+\lambda)^{N-1} & \text{(c)}, \end{cases} \tag{3}$$

in which ω serves as a tuning parameter, taking values from 0.3 to 0.6. N represents the number of rooms that must be traversed from o_j to o_k, excluding r_m and r_n. λ is a hyper-parameter that takes values ranging from 0.2 to 0.5, depending on the specific scenario.

The spatial reward is then expressed as:

$$R_{\text{spatial}}^q = \frac{1}{\alpha(o_j, o_k)} \times R_{\text{dist}}^q. \tag{4}$$

3) Load Balancing Reward: When the robot system is heterogeneous, it is necessary to consider whether the robot's capabilities align with the assigned tasks; if they do not match, the corresponding planning scheme is disregarded. When the robot system is homogeneous, we propose a stability measure based on statistical dispersion for modeling, and define the load balancing reward as:

$$R_{\text{load}} = \frac{1}{1 + \sqrt{\frac{1}{N}\sum_{i=1}^{N}(|s_i| - \mu)^2}}, \tag{5}$$

in which $S = \{s_1, s_2 ..., s_n\}$ represents the task allocation state of the robots, s_i represents the task set of robot i, N

represents the number of robots in this system. Mean μ is defined as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}|s_i|. \tag{6}$$

Finally, the overall reward $\alpha R^q_{\text{dist}} + \beta R^q_{\text{spatial}} + \gamma R_{\text{load}}$ for each LTL plan is calculated (the parameters α, β, and γ assume values within the intervals 0.1 to 0.2, 0.5 to 0.6 and 0.3 to 0.4, respectively), and the optimal task planning strategy is selected for the multi-robot system to perform the tasks.

C. Command Output

After obtaining the optimized task allocation scheme, we perform task minimization decomposition for each robot. We extend the task decomposition template from [5] to accommodate a broader range of multi-robot systems. For instance, when executing tasks in the real world, we obtain low-level instructions required for different robots to perform various tasks. These instructions are then packaged into template forms and input into the LLMs for learning. The LLMs subsequently generate minimized commands suitable for robot task execution. By modifying the low-level instructions within the template, this approach can be applied to general multi-robot systems.

IV. Experiments And Results

A. Simulation Experiments

We conduct comprehensive simulation experiments utilizing the AI2THOR [8] framework, an embodied artificial intelligence platform. Within this environment, RobotTHOR serves as a modular indoor simulation domain comprising 3-7 interconnected room configurations, each equipped with 36 parameterized action primitives for robotic task execution. Typical environmental setting is shown in Fig. 4. This experimental setup enables the acquisition of multimodal sensory data, including depth maps from egocentric robotic perspectives, object-centric semantic metadata, and continuous pose trajectories, thereby establishing quantifiable metrics for systematic performance evaluation.

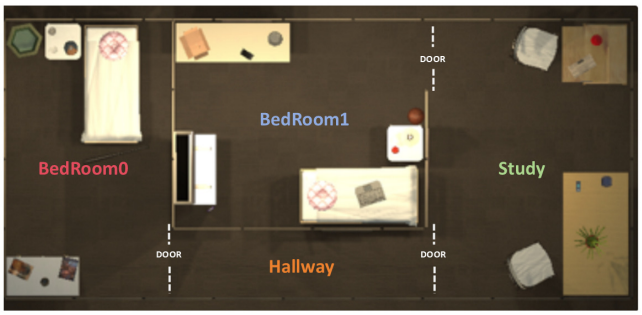


Fig. 4: An example of a multi-room environment in the AI2THOR simulation. The image highlights virtual doors and room names as defined in our setup.

To systematically evaluate the quantitative benchmarking of fundamental metrics between our methodology and baseline approaches, we established a multimodal dataset encompassing multi-relational scene graphs and long-horizon composite task sequences within this multi-room environment. The data set comprises five distinct scene graph instantiations, including the foundational and augmented scene graph. We name each room based on its structure and the objects stored in it. Notably, the absence of architectural door boundaries in AI2THOR necessitates the implementation of virtual transitional interfaces. We set the door and room names as Fig. 4 These are operationalized through coordinate calibration of robotic navigation trajectories, enabling precise spatial mapping of pseudo-doorway positions within the scene graph's ontological object hierarchy. All synthetic portals were algorithmically constrained to persistent open states to emulate frictionless inter-room navigation.

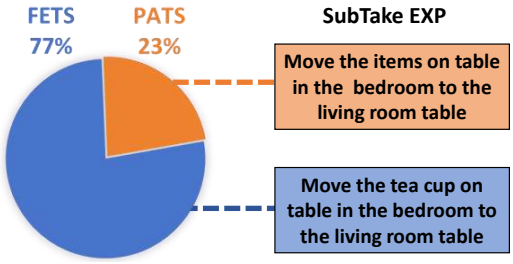


Fig. 5: The proportions of FETS and PATS in the total task set (with 27 FETS tasks and 8 PATS tasks) and the examples of their respective subtasks.

To ensure experimental rigor, each environmental configuration was paired with seven longitudinal task sequences containing two formally defined composite task typologies, as shown in Fig. 5:

Fully Explicit Target Subtasks (FETS): Complete specification of all mission-critical objects throughout hierarchical action sequences;

Partially Ambiguous Target Subtasks (PATS): Strategic omission of 1-2 objective specifications requiring contextual disambiguation.

We performed systematic ablation and comparative studies on the benchmark dataset. This rigorous evaluation enabled us to establish quantitative reliability metrics for the proposed methodology, and develop essential technical prerequisites for real-world robot experiments.

1) Ablation Study: To systematically evaluate the impact of scene graph representations, we conducted ablation studies across all five scenarios for long-horizon composite tasks, particularly examining their effects on cross-room decomposition tasks. Three experimental groups with varying environmental descriptions fed to the LLMs for task allocation and decomposition were designed:

- **Augmented Scene Graph (A-SG):** Utilizing relationship augments scene graphs as environmental descriptors.

- **Foundational Scene Graph (F-SG):** Employing non-augmented scene graphs for spatial representation.
- **No Scene Graph (N-SG):** Direct processing of object-level data streams without scene graph abstraction.

To isolate the effects of scene graph integration, we intentionally disabled the task allocation optimization from III-B during these ablation studies. Instead, the initial LTL sequences generated III-A were randomly assigned to multiple robots, thereby producing minimized decomposition instructions through this constrained pipeline. We established executable minimized commands for all task sequences in the dataset as ground truth. The executable minimized commands composed of robot actions and targets. These commands standardize robot identifiers and are independent of task allocation. The quality of generated instructions is evaluated by comparing command functions and object ID parameters with the ground truth, calculating the percentage of executable instructions.

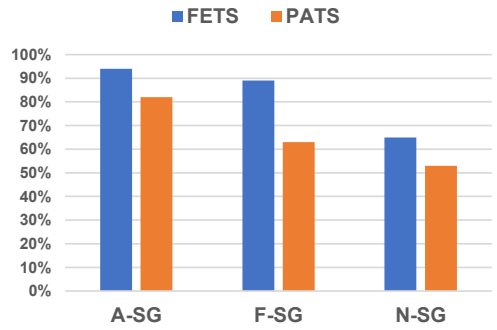


Fig. 6: An ablation study was conducted on the scene graph across various task types.

Experiment result as shown in Fig. 6, on the horizontal axis, we denote whether a scene graph was employed, and if so, the specific type of scene graph utilized. The vertical axis indicates the percentage of generated commands that can be executed. Analysis of the experimental results demonstrates that both FETS and PATS task sequences achieve the highest percentage of executable instructions when augmented scene graphs are employed for environmental representation. Notably, the impact of scene graph augmentation is less pronounced in FETS-form tasks. This phenomenon arises because the hierarchical topological structure of scene graphs facilitates LLMs comprehension of cross-room commands, particularly through our supplementary room-level annotations absent in the original dataset. The scene graphs effectively disambiguate identical objects across different rooms, whereas direct object-level data streams fail to resolve referential ambiguities between object names and environmental IDs, resulting in lower instruction executability. This discrepancy becomes more evident in PATS tasks, where certain subtasks lack explicit object references and instead require spatial reasoning combined with LLMs commonsense knowledge. Without scene graph abstraction, the generated LTL sequences exhibit over-reliance on commonsense priors rather than grounded spatial information. However, our experiments reveal that using augmented scene graphs still cannot enable PATS tasks to achieve the same percentage of executable instructions as PETS tasks. Moreover, as the number of "ambiguous" objects increases in the scenarios, the performance degrades significantly. This issue may require improved prompt engineering to better describe the scene graph structure to the LLMs for resolution.

2) Comparative Study: To assess the effectiveness of our approach in selecting task allocation plans for robots, we quantify the spatial efficiency metric for comparison. The calculation method is outlined as follows:

TABLE I: Cross-Scene Performance Evaluation

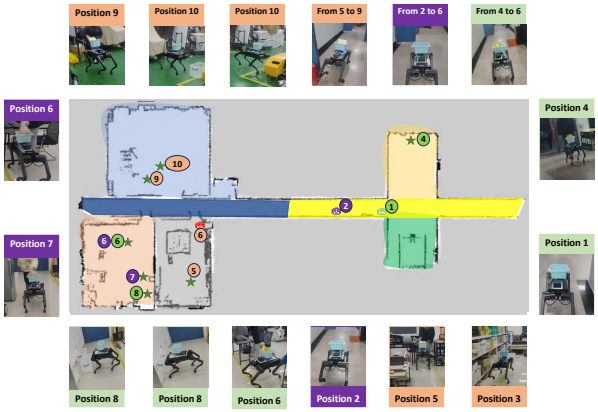
Method	Scene0 (4 rooms)	Scene1 (3 rooms)	Scene2 (3 rooms)	Scene3 (5 rooms)	Scene4 (6 rooms)
Ours	**0.175**	**0.255**	**0.196**	0.159	**0.194**
LLM	0.147	0.231	0.176	0.138	0.175
Baseline[5]	0.142	0.205	0.133	**0.177**	0.109

For each composite task sequence, we compute the sum of straight-line distances between the target points of each subtask as a baseline, denoted as L_B. Subsequently, by applying our method's heuristic selection, LLM-based heuristic selection, and the baseline selection, we obtain different multi-robot task allocation plans. These plans are subjected to instruction decomposition in simulation, followed by manual correction of unexecutable instructions to ensure that each instruction is executable. During the execution of each subtask by a robot, we sample and record the coordinates from the previous target point (or starting point) to the next target point and compute the sum of the distances between these coordinates, denoted as L_P. The efficiency is then quantified using the ratio L_B/L_P.

The experimental results are presented in Table I. Scenes 0 - 4 represent the simulation environments in the dataset, each encompassing varying numbers of rooms. From the experiments conducted in five different multi-room scenarios, it can be concluded that our method generally achieves the most efficient task planning solutions. However, in Scene 3, our method performs worse than the baseline. Upon analysis, we find that although Scene 3 consists of five different rooms, most of these rooms are directly connected, and most of the objects that can interact with the robots are concentrated in just two fixed rooms. The baseline's task allocation is mostly focused on a single robot, which increases the frequency of a robot performing tasks in a single room. This corresponds to the baseline's method of multi-robot task allocation in a single room. Thus, we conclude that our task allocation plan is more efficient for multi-robot interaction tasks executed across rooms, compared to tasks performed in a single room or between directly connected rooms.

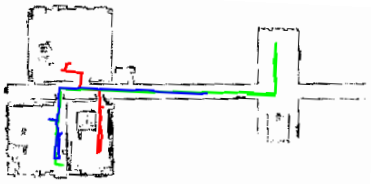
B. Real World Experiments

We tested our approach with three legged robots in both indoor and outdoor real-world environments, using the localization method proposed in [28], and compared it with the



(a) Leg-footed multi-robot systems executing long-sequence composite tasks in real-world environments.

(b) Scene Graph + LLM Task Planning



(c) Our Method Task Planning

Fig. 7: Results of real-world experiments: (a) illustrates a schematic representation of the target point locations for each subtask in the real-world experiments, along with the actual performance of the quadrupedal robot during task execution; (b) displays the task plan generated via LLMs heuristic guidance, while (c) illustrates the task plan produced using our heuristic function.

LLMs allocation scheme. The indoor real-world environment consisted of six rooms. The task setup was as follows: *Go find Tom, put the fruits on his desk into the fridge (1), and then get the takeout and give it to Tom (2). Meanwhile, find Petter to retrieve the files and deliver them to Tony (3). Additionally, push boxes in rooms 126 and 115 against the wall (4, 5). This long-sequence composite task is composed of five simple tasks, each of which can be further broken down into multiple sub-tasks* (e.g., task (1) can be split into: ① Go find Tom; ② Get the fruits; ③ Go to the fridge).

After collecting the environmental map and establishing the coordinate system, we randomly selected three coordinate points as the initial positions for the three robots. The initial positions of the three robots and the target areas in the task sequence are shown in Fig. 7(a). Fig. 7(b) shows the scene graph with the task planning selected by the LLMs, while Fig. 7(c) displays the task planning selected by our method. Our approach ensures balanced task distribution while minimizing the number of cross-room tasks assigned to the robots.

From the task planning routes of both schemes, it can be observed that in our approach, Robot 2 completes all tasks by traveling from the starting point to target point 6, and then to target point 8, with all tasks only crossing two rooms. In contrast, in the LLM-based planning, after completing the task at target point 8, Robot 2 enters another room to perform the task at target point 10, rather than assigning the task at target point 10 to Robot 3, which is already working on a task at target point 9.

V. Conclusion

We propose a method for optimizing the allocation of long-sequence composite tasks in complex cross-room/region scenarios for multi-robot systems. Using the spatial knowledge embedded in the augmented scene graph and the prior knowledge from LLMs, we decompose and allocate tasks into a series of simplified LTL task sequences. Additionally, we introduce a heuristic function for computing the optimal allocation plan. By modifying the underlying command templates of different systems, we generate executable commands for multi-robot operations. We conducted experiments in both simulated and real-world environments, which validate the effectiveness of our approach.

References

[1] Ali Bolu and Ömer Korçak. Adaptive task planning for multi-robot smart warehouse. *Ieee Access*, 9:27346–27358, 2021.

[2] Jorge Pena Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 8:191617–191643, 2020.

[3] Luke Antonyshyn, Jefferson Silveira, Sidney Givigi, and Joshua Marshall. Multiple mobile robot task and motion planning: A survey. *ACM Computing Surveys*, 55(10):1–35, 2023.

[4] Shintaro Ishikawa and Komei Sugiura. Target-dependent uniter: a transformer-based multimodal language comprehension model for domestic service robots. *IEEE Robotics and Automation Letters*, 6(4):8401–8408, 2021.

[5] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12140–12147. IEEE, 2024.

[6] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[7] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[8] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai, 2022.

[9] Yuchen Liu, Luigi Palmieri, Sebastian Koch, Ilche Georgievski, and Marco Aiello. Delta: Decomposed efficient long-term robot task planning using large language models. *CoRR*, 2024.

[10] Zhirong Luan, Yujun Lai, Rundong Huang, Shuanghao Bai, Yuedi Zhang, Haoran Zhang, and Qian Wang. Enhancing robot task planning and execution through multi-layer large language models. *Sensors*, 24(5):1687, 2024.

[11] Elango Murugappan, Nachiappan Subramanian, Shams Rahman, Mark Goh, and Hing Kai Chan. Performance analysis of clustering methods for balanced multi-robot task allocations. *International Journal of Production Research*, 60(14):4576–4591, 2022.

[12] Frank Imeson and Stephen L Smith. Multi-robot task planning and sequencing using the sat-tsp language. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5397–5402. IEEE, 2015.

[13] Oscar Valero, Javier Antich, Antoni Tauler-Rosselló, José Guerrero, Juan-José Miñana, and Alberto Ortiz. Multi-robot task allocation methods: A fuzzy optimization approach. *Information Sciences*, 648:119508, 2023.

[14] Philipp Schillinger, Mathias Bürger, and Dimos V Dimarogonas. Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems. *The international journal of robotics research*, 37(7):818–838, 2018.

[15] Xusheng Luo and Michael M Zavlanos. Temporal logic task allocation in heterogeneous multirobot systems. *IEEE Transactions on Robotics*, 38(6):3602–3621, 2022.

[16] Feifei Huang, Xiang Yin, and Shaoyuan Li. Failure-robust multi-robot tasks planning under linear temporal logic specifications. In *2022 13th Asian Control Conference (ASCC)*, pages 1052–1059. IEEE, 2022.

[17] Aakriti Agrawal, Senthil Hariharan, Amrit Singh Bedi, and Dinesh Manocha. Dc-mrta: Decentralized multi-robot task allocation and navigation in complex environments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11711–11718. IEEE, 2022.

[18] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *Conference on Robot Learning*, pages 23–72. PMLR, 2023.

[19] Yuchen Liu, Luigi Palmieri, Sebastian Koch, Ilche Georgievski, and Marco Aiello. Towards human awareness in robot task planning with large language models. *CoRR*, 2024.

[20] Zhirong Luan, Yujun Lai, Rundong Huang, Shuanghao Bai, Yuedi Zhang, Haoran Zhang, and Qian Wang. Enhancing robot task planning and execution through multi-layer large language models. *Sensors*, 24(5):1687, 2024.

[21] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

[22] Ruoyu Wang, Zhipeng Yang, Zinan Zhao, Xinyan Tong, Zhi Hong, and Kun Qian. Llm-based robot task planning with exceptional handling for general purpose service robots. In *2024 43rd Chinese Control Conference (CCC)*, pages 4439–4444. IEEE, 2024.

[23] Sthithpragya Gupta, Kunpeng Yao, Loïc Niederhauser, and Aude Billard. Action contextualization: Adaptive task planning and action tuning using large language models. *IEEE Robotics and Automation Letters*, 2024.

[24] Kazuma Obata, Tatsuya Aoki, Takato Horii, Tadahiro Taniguchi, and Takayuki Nagai. Lip-llm: Integrating linear programming and dependency graph with large language models for multi-robot task planning. *IEEE Robotics and Automation Letters*, 2024.

[25] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289*, 2020.

[26] Maëlic Neau, Paulo E. Santos, Anne-Gwenn Bosser, and Cédric Buche. React: Real-time efficiency and accuracy compromise for tradeoffs in scene graph generation, 2024.

[27] Zhirui Dai, Arash Asgharivaskasi, Thai Duong, Shusen Lin, Maria-Elizabeth Tzes, George Pappas, and Nikolay Atanasov. Optimal scene graph planning with large language model guidance. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14062–14069. IEEE, 2024.

[28] Yinchuan Wang, Bin Ren, Xiang Zhang, Pengyu Wang, Chaoqun Wang, Rui Song, Yibin Li, and Max Q-H Meng. Rolo-slam: rotation-optimized lidar-only slam in uneven terrain with ground vehicle. *Journal of Field Robotics*, 42(3):880–902, 2025.