Spending Semantics Wisely: Inference Delegation in Heterogeneous Multi-Robot Teams via Monte Carlo Tree Search Guided Dialogue

A Manicka Praveen, Yi Loo, Marcel Bartholomeus Prasetyo, Malika Meghjani
Singapore University of Technology and Design
{praveen_amanicka, yi_loo}@mymail.sutd.edu.sg, {marcel_prasetyo, malika_meghjani}@sutd.edu.sg

***Abstract*—Effective robot operation in complex environments demands not just the ability to perform semantic inference, but the capacity to reason about where such inference is worth spending. While selective inference strategies have shown promise for single-robot systems, extending this principle to heterogeneous multi-robot teams introduces a fundamentally different challenge. Given agents that differ in onboard compute capacity, the team must collectively reason about which robot should perform deep semantic inference, on which observations, and in what order. We reframe this as a team-level semantic delegation problem and propose a Monte Carlo Tree Search (MCTS) guided actor-critic dialogue framework in which robots jointly deliberate over semantic scene graph observations to produce compute-aware delegation plans. A deterministic scoring function provides structured feedback at each dialogue step, grounding language model reasoning in task constraints. Evaluated on a collaborative object retrieval task, our framework consistently outperforms single-round planning in both constraint satisfaction and objects retrieved, across simulation and real-world experiments, demonstrating that structured multi-agent dialogue enhances semantically-grounded resource allocation in heterogeneous robot teams.**

I. Introduction

Semantic understanding has become a cornerstone of capable robotic systems, enabling robots to reason about objects, spatial relationships, and task-relevant context in ways that purely geometric representations cannot support [6, 9, 27, 11, 21]. As robotic deployments scale toward heterogeneous multi-robot teams [14], this semantic reasoning capacity must be distributed across agents that differ not only in physical capabilities but also in their ability to perform inference. Yet while semantic reasoning frameworks have matured considerably for single-robot systems, and recent work has begun extending them to multi-robot settings through collaborative scene graph construction [3, 13], the question of how to coordinate semantic inference across agents with heterogeneous compute capabilities has received little attention.

Large foundation models have become the predominant tool for deep semantic reasoning [10], but carry significant computational cost for robots operating at the edge under strict constraints on resources, latency, and power [23]. This challenge is compounded in heterogeneous multi-robot teams, where agents differ not only in physical capabilities but also in onboard compute capacity. While multi-robot task allocation has been studied extensively [1, 5, 25] and semantic inference for single robots has advanced rapidly [9, 27], these two lines of work have largely developed in isolation. The question of how to jointly reason about what to infer semantically, and which robot should bear the cost of doing so, has not been addressed.

Efforts to reduce the computational burden of semantic inference have largely focused on model-level optimisations [17] rather than on reasoning about where inference is worth spending at a system level. One such system-level approach is explored in [21], where a coarse semantic representation is first constructed and expensive fine-grained inference is selectively triggered only in task-relevant regions, effectively treating semantic inference as a resource to be allocated rather than applied uniformly. However, extending this idea to multi-robot systems introduces a fundamentally different challenge: what was previously a single-agent decision becomes a cross-agent semantic coordination problem, where the team must collectively reason about which robot should perform deep inference, on which observations, and why.

We propose to address this gap by reimagining the bimodal inference decision as a team-level semantic reasoning problem. Rather than asking when a single robot should invoke deep inference, we ask which robot should do so, on which observations, and in what order, given a shared task objective and heterogeneous compute budgets. This reframing exposes a joint optimisation over semantic confidence, spatial efficiency, and per-robot resource constraints that cannot be resolved by independent per-robot reasoning alone.

To address this, we present a framework in which a team of heterogeneous robots jointly deliberate over semantic observations using a Monte Carlo Tree Search guided actor-critic dialogue. A deterministic scoring function evaluates candidate delegation plans at each step, providing structured feedback that guides the language models toward valid and efficient solutions. An overview of the proposed framework is shown in Fig. 1. The contributions of this work are as follows:

- A reframing of bimodal semantic inference as a multi-robot semantic reasoning problem, extending the single-agent fast/slow inference paradigm to heterogeneous robot teams.
- An MCTS-guided actor-critic dialogue mechanism for compute-aware semantic delegation, combining large language models (LLM) reasoning with deterministic plan

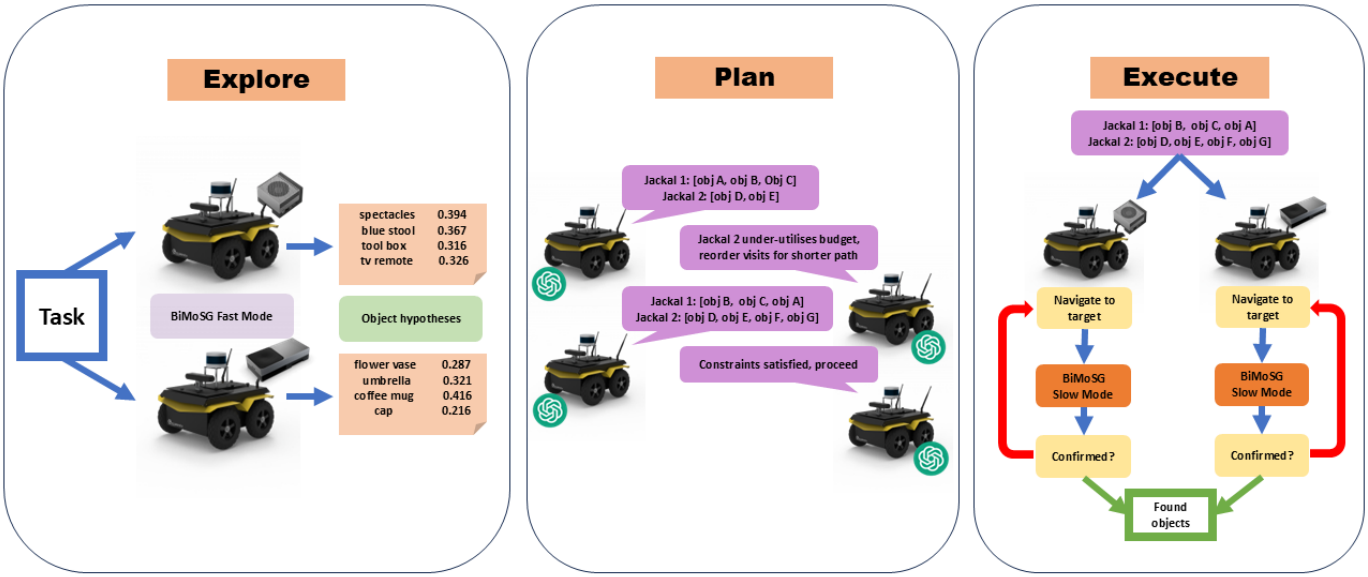

Fig. 1. System overview of our heterogeneous multi-robot framework. The pipeline consists of three phases: semantic exploration using BiMoSG fast mode, MCTS-guided actor-critic task distribution planning, and distributed execution via BiMoSG slow mode verification.

evaluation.

- Empirical validation on a collaborative object retrieval task, demonstrating that iterative MCTS dialogue consistently produces better delegation plans compared to single-round planning, in both simulation and real-world experiments.

II. Related Work

We situate our work across three related areas: semantic mapping and scene graph generation, multi-robot task allocation and coordination, and structured planning with LLMs.

A. Semantic Mapping and Scene Graphs

Scene graphs have emerged as a powerful representation for robotic semantic understanding, encoding objects, their attributes, and spatial relationships in a structured form that supports downstream reasoning and task planning. Early work on 3D scene graphs for robotics [11] demonstrated that hierarchical representations, spanning objects, rooms, and buildings, could be constructed incrementally from sensor streams, enabling robots to reason about their environment at multiple levels of abstraction. Subsequent work has extended this to open-vocabulary settings, where vision-language models allow robots to ground natural language queries against 3D scene representations without requiring a fixed object vocabulary [9, 27]. These approaches have significantly expanded the range of tasks robots can perform through semantic scene understanding [26].

While these scene graph approaches rely on fixed vocabularies or expensive inference for every observation, recent work has explored more efficient strategies. [21] proposes a bimodal inference framework for single-robot semantic scene graph generation, combining a lightweight closed-set detector with CLIP [22] for semantic similarity scoring for fast-mode operation, and selectively triggering a more powerful vision-language model only in regions likely to contain task-relevant objects. This selective usage of larger models reduces unnecessary compute while preserving semantic reasoning quality. However, this decision remains confined to a single agent, and does not account for the coordination challenges that arise when multiple robots with heterogeneous compute capabilities must collectively reason about the same task.

B. Multi-Robot Task Allocation and Coordination

Task allocation in multi-robot systems has been studied extensively, with classical approaches relying on auction-based mechanisms [18, 7] and optimisation formulations [8]. These methods have demonstrated effective coordination across a range of multi-robot applications, from search and rescue to warehouse automation.

More recently, LLMs have been applied to multi-robot coordination, leveraging their commonsense reasoning and natural language understanding to produce more flexible task plans. RoCo [16] equips each robot with an LLM agent that engages in dialogue to collectively negotiate task strategies and low-level motion plans, with environment feedback used to iteratively refine proposals in-context. SMART-LLM [12] takes a centralised approach, decomposing high-level instructions into subtasks and allocating them across agents through multi-stage LLM prompting. While these frameworks demonstrate the promise of language-guided coordination for multi-robot systems, our work addresses a complementary problem: the allocation of semantic inference resources across heterogeneous agents toward a shared task objective.

C. Structured Planning with Large Language Models

The rich semantic knowledge encoded within LLMs also allow them to produce semantic meaningfully plans that are

grounded in real robot environments. Most notably, SayCan [2] combined the reasoning capabilities of LLMs and learned value functions to select the appropiate skills during task execution. Code as Policies [14] and ProgPrompt [24] uses LLMs to generate high level robot plans in from of executable code. Our work extends the use of LLMs to a multi-robot setup where robots can use LLMs to perform task distribution in a decentralised manner.

In the context of multi-agent planning. CoELA [28] was proposed as framework for embodied agents to perform cooperative tasks by exchanging information via dialogue. Each agent utilizes LLMs to reason about messages to send to their teammates, and generate execution plans based on the environment as well as messages from other teammates. CoTS [29] builds upon CoELA by introducing a Monte Carlo Tree Search based collaborative planning module in the planning and communication phase, where each agent acts as either planner or critic to iteratively refine the collaborative plans via MCTS. Our work extends this to a multi-robot setup where LLMs are used to reason collectively about semantic inference delegation across agents with heterogeneous compute capabilities.

III. Methodology

A. System Overview

Our framework operates in three phases, as illustrated in Fig. 1. During the exploration phase, each robot autonomously navigates its assigned zone and builds a scene graph, together with local object hypotheses using the BiMoSG fast mode pipeline [21], producing CLIP similarity scores and estimated positions for each candidate object. These observations, together with the task objective and various constraints, are passed to the planning module. The planning phase employs an MCTS-guided actor-critic dialogue. The two LLM agents reason over shared observations without a central coordinator. The dialogue protocol is agnostic to where each agent is physically hosted, and can be realised either on a single robot or distributed across the team. The actor agent proposes a delegation plan, a deterministic scorer evaluates the plan across three components: CLIP rank quality: whether each robot selects the highest-confidence candidates in its zone; path efficiency: whether the visit ordering minimises total travel distance within each robot's distance budget; and query budget utilisation: whether each robot fully uses its allocated number of VLM queries. These three components are summed to produce a score out of 10, which is passed to the critic agent alongside the proposed plan. The critic agent provides targeted feedback to guide refinement based on the given information. This loop iterates until a valid plan is found or a preset maximum depth is reached. The resulting plan assigns each robot an ordered list of candidate objects to verify. In the execution phase, each robot navigates to its assigned candidates and triggers the BiMoSG slow mode [21] for verification, contributing confirmed objects toward the team score.

B. Bimodal Scene Graph (BiMoSG)

BiMoSG [21] generates representations of 3D objects through a relatively low computation cost without the use of large foundation models. This process, dubbed the "fast" mode is able to generate prismatic object representation of closed set objects though lower cost models. Specifically, given a overall textual goal, Mask2Former [4] to generate initial class predictions and CLIP [22] to give a likeliness score of the object toward fulfilling the goal. The robot can then decide to turn on the "slow" mode which utilized VLM to further confirm the target object. This is in contrast to other notable 3D scene graph generation works [9, 15] which directly utilized vision foundation models to generate open-set object representations but at the cost of higher computation and slower inference times.

C. Collaborative Task Distribution via Monte Carlo Tree Search

Upon completing the exploration phase, the robots proceed to the task distribution phase. Inspired by CoTS [29], we introduce a Task Distribution Module that utilises MCTS to iteratively refine delegation plans. Two LLM agents, running GPT-4o [19, 20], an actor and a critic jointly deliberate over the shared semantic observations from both zones. The plan proposer takes in the semantic information generated by both robots and proposes an initial plan. The actor proposes an initial delegation plan, which is evaluated by a deterministic scorer and reviewed by the critic. The critic reads the score and any flagged issues, and provides targeted feedback to the actor. This initial proposal and critique serve as the root node for the planning process, which consists of the following four steps.

Selection At each iteration, the planning module starts at the root and traverses the tree based on the UCT selection policy until a leaf node is reached. The UCT policy is defined as follows:

$$UCT(c_i) = V(c_i) + w\sqrt{ln\frac{N(p_i)}{N(c_i)}} \tag{1}$$

where $N(n_i)$ is the number of visits for node c_i, $(V(c_i)$ is back-propagated value of the the subtree from c_i, w is exploration coefficient and p_i is the parent of c_i

Expansion Once a leaf node is reached, the module then expands the tree with n new child nodes. Similar to the root node, each node consists of a distribution plan generated by the proposer and a critique from the critic. Crucially, each child node's plan is a refinement of the parent node's plan based on the critique in the parent node.

Evaluation Each candidate plan is evaluated by a deterministic scorer across three components, chosen to reflect the three independent constraint axes of the problem. Query budget utilisation is scored 0–4, awarding one point per robot whose inference budget is fully expended; this component receives the highest weight because under-utilisation of the higher-compute agent is the most consequential failure mode

in heterogeneous teams. CLIP rank quality is scored 0–3, awarding one point per robot whose selected candidates correspond to the highest CLIP-similarity detections in its zone; a third point is awarded when the cross-robot ordering is also consistent with global confidence rank. Path efficiency is scored 0–3, deducting one point per robot whose cumulative visit distance exceeds its budget, with a further point deducted when the visit ordering is suboptimal relative to a nearest-neighbour baseline. The three components sum to a raw score out of 10, normalised to $r = \frac{\text{score}}{10} \in [0, 1]$ for backpropagation. A plan is considered valid if it scores $\geq 7.0/10$ and satisfies all hard constraints. The threshold of 7.0 was selected empirically: a score below 7.0 implies at least one full component has been violated, and plans scoring at or above this threshold were consistently found to meet all per-robot constraints in pilot trials. The hard constraints are: (i) no inference budget violation, meaning a robot does not exceed its allocated number of VLM queries; (ii) no distance budget violation, meaning a robot's planned path does not exceed its travel budget; (iii) no cross-zone assignment, meaning the planner does not assign a candidate to the robot operating in the opposite zone; and (iv) no duplicate targets, meaning the same candidate does not appear on more than one robot's list. The critic reads the numerical score and any flagged constraint violations, and provides targeted feedback to the actor to guide the next expansion.

Backpropagation The normalised reward r is back-propagated from the evaluated node to the root. The visitation count and value at each node along the traversal path are updated as follows:

$$N'(c_i) \leftarrow N(c_i) + 1, \quad \forall c_i \in \{c_0, \ldots, c_d\} \tag{2}$$

$$V'(c_i) \leftarrow \frac{V(c_i)N(c_i) + r}{N'(c_i)}, \quad \forall c_i \in \{c_0, \ldots, c_d\} \tag{3}$$

where d is the current depth of the tree.

Final Plan Selection The planning module terminates after a preset number of expansions. The final delegation plan is retrieved from the highest-value solved node in the tree. If no solved node exists, the node with the highest back-propagated value $V(c_i)$ is selected as a fallback.

IV. Experiments And Results

A. Experimental Setup

Experiments are conducted in an indoor laboratory environment consisting of two circular zones, each approximately 3m in diameter, separated by a physical divider. as shown in 2. Zone 1 is assigned to Jackal 1, a Clearpath Jackal equipped with an NVIDIA Jetson AGX Orin, and Zone 2 to Jackal 2, a Clearpath Jackal equipped with an NVIDIA Jetson AGX Thor. Both robots are equipped with a ZED 2 stereo camera for RGB-D perception.

We conduct experiments in two settings: real-world trials using the physical robot platforms described above, and simulation trials using a synthetically generated corpus that mirrors the real-world setup but at larger scale with controlled observation noise, allowing us to assess planner performance under more idealised conditions.

Fig. 2. Zone 1 is highlighted in green, and Zone 2 is highlighted in orange.

Each zone contains a set of distractor objects arranged around the perimeter, among which four target objects are placed as shown in 3 The target objects in Zone 1 are a fire extinguisher, blue stool, toolbox, and television remote. The target objects in Zone 2 are a flower vase, cap, umbrella, and mug. Objects are selected to span a range of detection difficulties. Some are visually distinctive and reliably detected by the fast mode pipeline, while others are smaller or visually similar to distractors as shown in 4, making them harder to localise with high confidence.

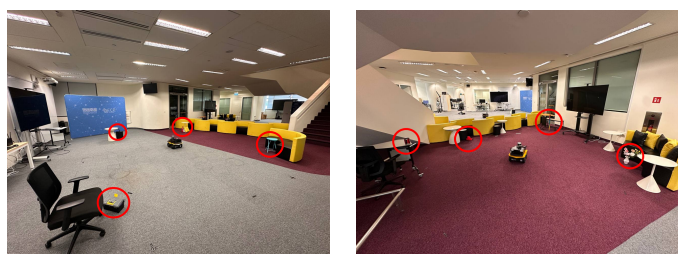

Fig. 3. Experimental zones with target objects circled. Left: Zone 1 (Jackal 1). Right: Zone 2 (Jackal 2).

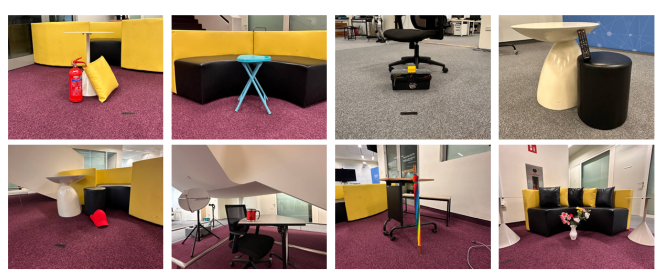

Fig. 4. Target objects used in real-world experiments. Top row: Zone 1 objects (fire extinguisher, blue stool, toolbox, television remote). Bottom row: Zone 2 objects (flower vase, cap, umbrella, mug).

Each robot is allocated an inference budget reflecting its onboard compute capacity. Jackal 1, with lower compute, is permitted at most 2 deep inference queries per trial, while Jackal 2, with higher compute, is permitted at most 3. This asymmetry is intentional to mirror realistic heterogeneous deployments where more capable agents can shoulder a greater share of expensive inference. A distance budget of 13m per robot per trial is additionally imposed, requiring the planner to consider not only which objects to query but also the order in which they are visited. Together, these constraints encourage the planning module to produce spatially efficient and resource-aware delegation plans rather than naive selections. While the full execution pipeline including physical navigation and VLM-based verification is implemented and operational, we opt to evaluate plan quality against manually verified ground truth rather than physically executing each trial. This allows us to isolate and evaluate the planning mechanism independently of execution noise and physical variability across trials.

B. Evaluation Metric

The final delegation plan is evaluated against a ground truth set of correctly detectable objects. For real-world trials, this is established by manual inspection of each robot's scene graph against the physical environment. For simulation trials, ground truth is known by construction from the synthetic data generation process.

For each trial, we compute the maximum number of ground truth objects reachable within the inference and distance budgets across all possible visit orderings, which varies per trial depending on the scan observations. A plan is considered successful if the number of confirmed objects equals this maximum, indicating that the delegation plan made optimal use of the available scan observations. Additionally, a plan is considered valid if it meets all of the following conditions: each robot's query budget is fully utilised, no robot exceeds its distance budget, no object is assigned to the wrong zone, no object appears on both robots' lists, and the overall plan score meets the validity threshold of $\geq$ 7.0/10. We report mean number of objects found, mission success rate, and plan validity across all trials, comparing the initial one-shot plan against the final MCTS-refined plan.

C. Baseline

We compare our MCTS-guided dialogue against a single-round planning baseline, in which the actor agent proposes a delegation plan given the same semantic observations, task objective, and constraints, but without any subsequent scoring, critic feedback, or iterative refinement. This corresponds to the initial plan produced at the root node of the MCTS tree before any expansion. All other conditions such as the LLM, prompt structure, and input observations remain identical, ensuring that any difference in plan quality is attributable solely to the iterative MCTS dialogue.

D. Results

Table I presents the per-trial results across 10 real-world trials. A key finding is that one-shot planning consistently under-utilises the Jackal 2's higher inference budget in all 10 trials, the initial plan assigned Jackal 2 only 2 objects instead of its full budget of 3, resulting in 0/10 constraint-satisfying plans. This suggests that without iterative feedback, the actor agent defaults to conservative allocations that fail to exploit the heterogeneous compute capabilities of the team. MCTS dialogue reliably corrects this: after refinement, all 10 final plans satisfy the full set of constraints, demonstrating that the critic consistently identifies and addresses budget under-utilisation regardless of trial conditions.

TABLE I
PER-TRIAL OBJECT RETRIEVAL RESULTS. OBJECTS: NUMBER OF OBJECTS FOUND. VALID: WHETHER THE PLAN SATISFIED ALL CONSTRAINTS. MAX: MAXIMUM REACHABLE GIVEN SCAN OBSERVATIONS.

	Initial (One-Shot)		**Final (MCTS)**		
Trial	**Objects**	**Valid**	**Objects**	**Valid**	**Max**
1	2	×	4	✓	4
2	3	×	3	✓	4
3	2	×	4	✓	5
4	2	×	2	✓	3
5	3	×	3	✓	4
6	2	×	3	✓	3
7	2	×	2	✓	3
8	2	×	4	✓	4
9	4	×	4	✓	4
10	2	×	2	✓	4
Mean	**2.4**	**0/10**	**3.1**	**10/10**	**3.8**

Beyond constraint satisfaction, MCTS-refined plans improved mean objects found from 2.4 to 3.1 out of a mean maximum of 3.8. In trials where improvement was observed (trials 1, 3, 6, 8), the revised delegation plan directed Jackal 2 to query a third object that was correctly localised in the scan, directly contributing to the higher team score. In trials where no improvement was observed despite a valid final plan (trials 2, 4, 5, 7, 10), the limiting factor was scan noise rather than plan quality.

TABLE II
MCTS SCORE CONVERGENCE ACROSS 10 REAL-WORLD TRIALS.

Planning step	**Mean score (/10)**	**Constraint-satisfied**
Initial (one-shot)	8.98	0/10
Expand 1	9.63	7/10
Expand 2	9.75	7/10
Expand 3	9.75	7/10
Expand 4	9.81	8/10
Expand 5+	9.81	8/10

Table II shows the convergence of MCTS planning scores across 10 trials. The most significant improvement occurs at the first expansion, where constraint-satisfied plans jump from 0/10 to 7/10 and mean score increases from 8.98 to 9.63. Beyond expand 1, gains diminish rapidly. Constraint satisfaction reaches 8/10 at expand 4 and plateaus thereafter,

with mean score stabilising at 9.81. This suggests that for problems of this scale, 4 to 5 MCTS expansions are sufficient to produce near-optimal delegation plans, and further expansions yield negligible improvement. The remaining 2/10 trials that never achieve constraint satisfaction correspond to cases where scan noise produced candidate positions that made it geometrically infeasible to visit 3 objects within the distance budget, representing a hard limit imposed by the environment rather than the planner.

TABLE III
SUMMARY OF SIMULATION EXPERIMENT RESULTS.

Metric	Initial (one-shot)	Final (MCTS)
Mean objects found	4.80	6.40
Mean max possible	7.75	
Mean improvement	+1.60	
Retrieval efficiency	61.9%	82.6%
Valid plan	0/20	17/20

Table III presents the results of the simulation experiments, conducted on a scaled-up setting with 10 objects per zone and asymmetric query budgets of 4 and 8 for Jackal 1 and Jackal 2 respectively, reflecting a more pronounced degree of compute heterogeneity. Despite the increased combinatorial complexity, MCTS-refined plans consistently outperform one-shot planning, improving mean objects found from 4.80 to 6.40 and retrieval efficiency from 61.9% to 82.6%. Constraint satisfaction improves from 0/20 to 17/20, consistent with the real-world finding that one-shot planning reliably under-utilises the higher-compute agent's query budget. The remaining 3 trials that failed to satisfy constraints correspond to cases where the spatial distribution of candidate objects made it geometrically difficult to visit the full query budget within the distance budget, representing a hard limit imposed by the environment rather than the planner.

E. Discussion

The results across both real-world and simulation experiments point to a consistent finding: one-shot LLM planning reliably under-utilises the higher-compute agent's query budget, producing constraint-violating plans in all trials regardless of problem scale. This suggests that without iterative feedback, LLMs default to conservative allocations that fail to exploit heterogeneous compute capabilities. This limitation is not obvious from the raw objects-found metric alone, but becomes clear when constraint satisfaction is tracked explicitly. MCTS dialogue addresses this directly, with the critic consistently identifying and correcting budget under-utilisation in the first expansion. The similarity in retrieval efficiency improvement across real-world ($\sim$18%) and simulation ($\sim$21%) settings further suggests that this correction mechanism generalises beyond the specific object set and environment used in physical experiments.

In the real-world experiments, the gap between final and maximum possible objects found is partly attributable to upstream perception quality. The exploration phase initialises the planner using CLIP similarity scores computed over closed-set detections. When detections are incorrect, the planner operates on imperfect observations. Note that detection errors (incorrect class predictions) and localisation errors (inaccurate position estimates) are distinct failure modes, and the gap between final and maximum reachable objects reflects the former rather than the latter. In trials where no improvement was observed despite a valid final plan, the remaining gap is largely attributable to the tightness of the combined query and distance constraints. With a fixed budget, the planner cannot visit all high-confidence candidates, and in some spatial configurations no delegation strategy can recover the remaining ground truth objects. These cases represent the fundamental limits of budget-constrained planning rather than a shortcoming of the MCTS dialogue itself. The planner produces the best feasible plan given the constraints, but the constraints themselves prevent a perfect outcome.

This work demonstrates that iterative multi-agent dialogue is beneficial for task distribution in heterogeneous robot teams. The consistent improvement in constraint satisfaction and objects found across both real-world and simulation experiments validates the core premise that structured LLM dialogue, grounded by a deterministic scorer, can produce better delegation plans than single-round planning. Two directions stand out as particularly promising for future work.First, incorporating detection uncertainty estimates from the exploration phase into the scoring function, so the planner can weight candidate selections by detection reliability in addition to CLIP similarity rank. Second, extending the framework to support online re-delegation, where the team revises its semantic inference strategy mid-execution as new observations arrive, rather than committing to a fixed plan after exploration. Both directions extend the core premise of this work that inference, treated as a finite resource, benefits from structured collective reasoning rather than independent per-robot decisions.

V. CONCLUSION

We presented a framework for heterogeneous multi-robot task delegation that reimagines selective bimodal inference as a team-level semantic reasoning problem. By introducing an MCTS-guided actor-critic dialogue grounded by a deterministic scorer, we enable a pair of robots with differing compute capabilities to jointly deliberate over semantic observations and produce constraint-satisfying delegation plans. Experiments across both real-world and simulation settings demonstrate that iterative MCTS dialogue consistently outperforms single-round planning in constraint satisfaction and objects retrieved, validating the premise that structured multi-agent dialogue is an effective mechanism for semantically-grounded inference delegation in heterogeneous robot teams. Beyond the specific task studied here, we believe the core insight that semantic inference is a resource to be reasoned about collectively, not applied uniformly points toward a broader design principle for multi-robot systems operating under real-world compute constraints. Across both settings, MCTS dialogue improved retrieval efficiency by approximately 18–21% over

single-round planning, suggesting that structured collective reasoning over inference resources is a practical and effective design principle for heterogeneous robot teams.

Acknowledgments

This work was supported by TL@SUTD Seed Research Programme, which encourages innovative research in data and information exploitation.

References

[1] Haris Aziz, Hau Chan, Ágnes Cseh, Bo Li, Fahimeh Ramezani, and Chenhao Wang. Multi-robot task allocation – complexity and approximation, 2021. URL https://arxiv.org/abs/2103.12370.

[2] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pages 287–318. PMLR, 2023.

[3] Yun Chang, Nathan Hughes, Aaron Ray, and Luca Carlone. Hydra-multi: Collaborative online construction of 3d scene graphs with multi-robot teams, 2023. URL https://arxiv.org/abs/2304.13487.

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation, 2022. URL https://arxiv.org/abs/2112.01527.

[5] Weiheng Dai, Utkarsh Rai, Jimmy Chiun, Yuhong Cao, and Guillaume Sartoretti. Heterogeneous multi-robot task allocation and scheduling via reinforcement learning. *IEEE Robotics and Automation Letters*, PP:1–8, 03 2025. doi: 10.1109/LRA.2025.3534682.

[6] Sourav Garg, Niko Sünderhauf, Feras Dayoub, Douglas Morrison, Akansel Cosgun, Gustavo Carneiro, Qi Wu, Tat-Jun Chin, Ian Reid, Stephen Gould, Peter Corke, and Michael Milford. Semantics for robotic mapping, perception and interaction: A survey. *Foundations and Trends® in Robotics*, 8(1-2):1–224, December 2020. ISSN 1935-8261. doi: 10.1561/2300000059. URL http://dx.doi.org/10.1561/2300000059.

[7] Brian P Gerkey and Maja J Mataric. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[8] Brian P Gerkey and Maja J Mataric. Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *2003 IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3862–3868. IEEE, 2003.

[9] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[10] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, Shibo Zhao, Shayegan Omidshafiei, Dong-Ki Kim, Ali akbar Aghamohammadi, Katia Sycara, Matthew Johnson-Roberson, Dhruv Batra, Xiaolong Wang, Sebastian Scherer, Chen Wang, Zsolt Kira, Fei Xia, and Yonatan Bisk. Toward general-purpose robots via foundation models: A survey and meta-analysis, 2024. URL https://arxiv.org/abs/2312.08782.

[11] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. In *Robotics: Science and Systems*, 2022.

[12] Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models, 2024. URL https://arxiv.org/abs/2309.10062.

[13] Yirum Kim, Jaewoo Kim, and Ue-Hwan Kim. Ma3dsg: Multi-agent 3d scene graph generation for large-scale indoor environments, 2026. URL https://arxiv.org/abs/2602.04152.

[14] Peihan Li, Zijian An, Shams Abrar, and Lifeng Zhou. Large language models for multi-robot systems: A survey, 2026. URL https://arxiv.org/abs/2502.03814.

[15] Dominic Maggio, Yun Chang, Nathan Hughes, Matthew Trang, Dan Griffith, Carlyn Dougherty, Eric Cristofalo, Lukas Schmid, and Luca Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *IEEE Robotics and Automation Letters*, 2024.

[16] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models, 2023. URL https://arxiv.org/abs/2307.04738.

[17] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

[18] Maitreyi Nanjanath and Maria Gini. Dynamic task allocation for robots via auctions. In *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2781–2786. IEEE, 2006.

[19] OpenAI. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

[20] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[21] Marcel Bartholomeus Prasetyo, Shrutika Vishal Thengane, A Manicka Praveen, Yi Loo, and Malika Meghjani. Seeing fast and slow: Bimodal 3d scene graphs for open-set tasks, 2026. Manuscript under review.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al.

Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[23] Ahmed Sharshar, Latif U. Khan, Waseem Ullah, and Mohsen Guizani. Vision-language models for edge networks: A comprehensive survey. *IEEE Internet of Things Journal*, 12(16):32701–32724, August 2025. ISSN 2372-2541. doi: 10.1109/jiot.2025.3579032. URL http://dx.doi.org/10.1109/JIOT.2025.3579032.

[24] 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.

[25] 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. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2023.119508. URL https://www.sciencedirect.com/science/article/pii/S0020025523010939.

[26] Beichen Wen, Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. 3d scene generation: A survey, 2025. URL https://arxiv.org/abs/2505.05474.

[27] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *Robotics: Science and Systems XX*, RSS2024. Robotics: Science and Systems Foundation, July 2024. doi: 10.15607/rss.2024.xx.077. URL http://dx.doi.org/10.15607/RSS.2024.XX.077.

[28] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *International Conference on Learning Representations*, volume 2024, pages 19373–19401, 2024.

[29] Lizheng Zu, Lin Lin, Song Fu, Na Zhao, and Pan Zhou. Collaborative tree search for enhancing embodied multi-agent collaboration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29513–29522, 2025.