

EMPOWER: Embodied Multi-role Open-vocabulary Planning with Online Grounding and Execution

Francesco Argenziano¹, Michele Brienza¹, Vincenzo Suriani², Daniele Nardi¹, and Domenico D. Bloisi³

Abstract—Task planning for robots in real-life settings presents significant challenges. These challenges stem from three primary issues: the difficulty in identifying grounded sequences of steps to achieve a goal; the lack of a standardized mapping between high-level actions and low-level commands; and the challenge of maintaining low computational overhead given the limited resources of robotic hardware. We introduce EMPOWER, a framework designed for open-vocabulary online grounding and planning for embodied agents aimed at addressing these issues. By leveraging efficient pre-trained foundation models and a multi-role mechanism, EMPOWER demonstrates notable improvements in grounded planning and execution. Quantitative results highlight the effectiveness of our approach, achieving an average success rate of 0.73 across six different real-life scenarios using a TIAGo robot.

I. INTRODUCTION

A plan is a finite sequence of actions that one or more agents must execute to achieve a specific goal within a given environment. When this environment is a real-world setting, whether controlled or not, the already inherent difficulty of this task increases even more. This happens because the actions need to be *grounded* in the environment, meaning a mapping is needed between the high-level actions of the plan - described either in natural language or in some formal language - and the low-level actions that the agent can execute in the physical world, in terms of joint motions.

When those agents are real robotic systems, they typically face limitations in computational resources due to their hardware, unlike simulated agents that can run on more powerful machines. As a result, a trade-off is necessary between performance and computational overhead, which can harm the system, especially if fast execution is required. Thus, enabling an embodied agent to accurately plan and act in the world is a significant challenge in this context.

In recent years, we have witnessed the emergence of Foundation Models (FMs), i.e. machine learning models trained on a vast, *Internet-scale* amount of data adaptable for a wide range of downstream applications [1]. Although initially designed for Natural Language Processing (NLP) tasks, these models have found utility across a variety of cross-domain applications. This happened because as a consequence of being trained on such a broad amount of data, FMs have incorporated the *commonsense knowledge* which is proper

¹Department of Computer, Automation and Management Engineering, University of Rome "La Sapienza", 00181 RM Rome, Italy lastname@diag.uniroma1.it ²School of Engineering, University of Basilicata, 85100 Potenza, Italy vincenzo.suriani@unibas.it ³International University of Rome UNINT, Rome RM 00147, Italy domenico.bloisi@unint.eu

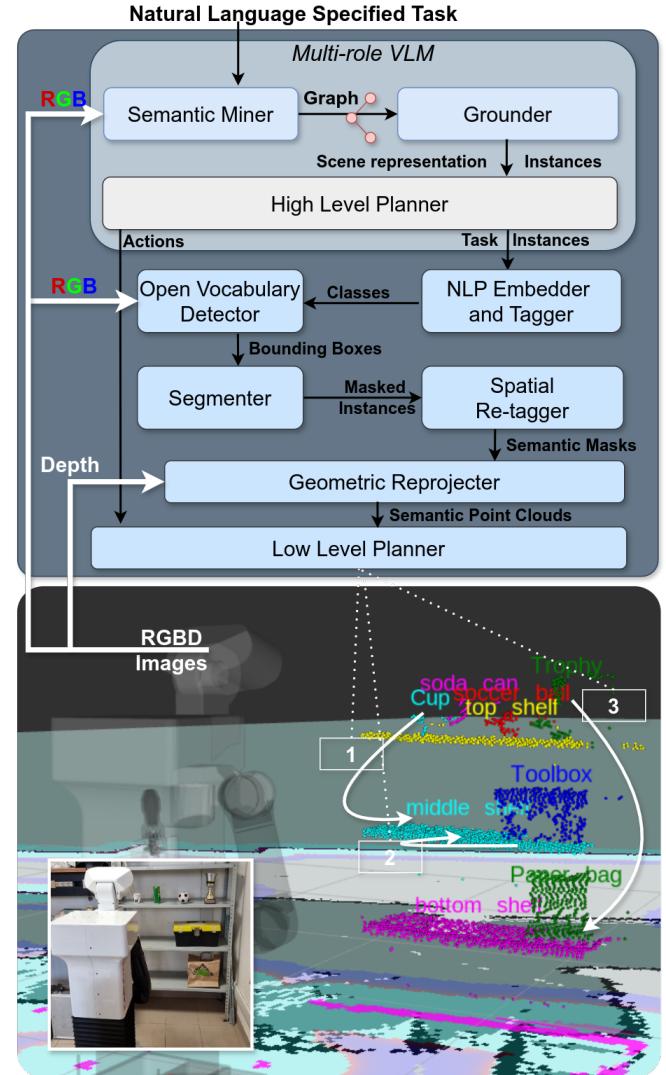


Fig. 1. The EMPOWER architecture during the execution of one of the use cases analyzed: reordering the shelf to have only 2 objects per level.

to human reasoning. This knowledge, along with *chain-of-thought* reasoning [2] and *prompt engineering* techniques [3], has shown that such models, in particular, Large Language Models (LLMs) [4], [5], [6], can function as few-shot planners [7] and zero-shot planners [8] when provided with prompts in a predefined structure. Recent research has shown that architectures involving multiple agents, each prompted with different roles to work together in a collaborative environment, outperform single-agent architectures [9], [10].

The drawback of these architectures is that they increase

the already high overhead caused by LLMs, which require substantial computational resources, such as dedicated graphics cards, not only for training but also for inference. This creates a conflict with the need for online processing. However, recent advancements have led to the development of FMs that can operate on limited hardware within reasonable time frames, while maintaining high-quality performance, therefore enabling also their use in constrained applications.

In this paper, we introduce EMPOWER, a framework for task planning with robots in real-world domains (see Fig. 1). EMPOWER leverages cutting-edge FMs and cloud services to create an open-vocabulary representation of the world for efficient action grounding, and utilizes multi-role prompting techniques to generate sound plans in complex scenes. The contributions of this work are three-fold:

- 1) We propose a multi-role, single-body hierarchical sub-task division that enables complex reasoning about the scene and allows for the creation of plans that can be grounded in the environment;
- 2) We present a complete pipeline that incorporates FMs for extracting meaningful information from the world, while being resilient to the online constraint given by the limited computational resources of real robots;
- 3) We provide a flexible solution that, thanks to our novel open-vocabulary framework, grounds and deploys plans across different robotic platforms in the environment.

Quantitative results show the effectiveness of the proposed approach with a 0.73 average Success Rate (SR) across six challenging use cases performed with a TIAGo robot¹.

The remainder of the paper is organized as follows. The subsequent Section II provides an overview of recent methods for robot planning. Section III presents the details of our approach. Qualitative and quantitative experimental results are presented in Section IV. Finally, conclusions are drawn in Section VI.

Videos of the TIAGo robot's execution, code, and additional material are publicly available at <https://lab-roccoco-sapienza.github.io/empower/>.

II. RELATED WORK

In this section, we review the most relevant literature on the topic. We first discuss the applications of LLMs in the field of robotics (Section II-A). Next, we examine the benefits of prompt engineering, particularly in multi-role and multi-agent scenarios (Section II-B). Finally, we analyze the use of open-vocabulary detectors in complex scenarios (Section II-C).

A. LLMs and robotics

Large Language Models (LLMs) [4], [5], [11], [6], and Visual Language Models (VLMs) [12], [13], [14] have emerged as some of the most significant achievements in Artificial Intelligence (AI) in recent years. Their integration with robotics has rapidly advanced, driven by the benefits

these models offer. FMs have started being applied in various areas, including navigation [15], Human-Robot Interaction [16], conversation [17], collaboration [18], and planning.

A milestone in this field is the SayCan framework [19], which demonstrated how LLMs can serve as a planning component for embodied agents. In this framework, the robot acts as the *hands and eyes* of the LLM, grounding actions to solve daily tasks in real-world environments. Subsequent works relaxed the assumption of relying solely on textual inputs, incorporating feedback from the environment either through images [20] or custom data structures [21].

Correctly prompting the LLM has also shown how LLMs can be used both as few-shot [4] and zero-shot planners [8], where minimal or no training is sufficient to outperform heavily trained models and generate reliable plans for various tasks in real-world scenarios. This is largely due to the commonsense knowledge embedded in FMs. Many other works followed this research trend: Dorbala et al. [22] demonstrated how robots can locate objects in a scene even when they are described using natural language (*can you find a "Cat-Shaped" mug?*), while Yang et al. [23] illustrated how despite the *bag-of-words* phenomenon [24] present in CLIP-based VLMs, an efficient vision-language-driven agent can be designed for visual navigation tasks.

Another approach involves designing specific transformer architectures [25] for robotics. The Robotics Transformers family [26], [27], [28] has developed models with zero-shot generalization capabilities for new tasks, with ongoing efforts to create versatile robot policies that can be applied across different robots and environments.

However, combining FMs and robots presents some challenges, particularly the need for substantial computational resources[29]. Many robots have limited hardware, often lacking GPUs, which makes deploying LLM- or VLM-based applications difficult without external computational support. This limitation restricts the deployment of these systems to controlled setups if fast, online execution is required.

Despite these challenges, recent advances in transformer-based architectures and FMs demonstrate that it is possible to run these models autonomously on robots while maintaining state-of-the-art performance.

B. Multi-role prompting

Designing prompts to follow specific patterns has proven to be very effective, leading to better performance compared to free-form text (*prompt engineering*) [3]. Notably, Wei et al. [2] showed that asking an LLM to explicitly describe its reasoning process significantly improves the quality of the results, a technique known as *chain-of-thought* reasoning. Building on this concept, other prompting strategies have been developed, such as *tree-of-thoughts* [30], which is particularly relevant in planning applications where multi-step reasoning is required. Despite these advancements, LLMs can still suffer from issues such as hallucinations or other undesirable outputs. A highly effective solution to this problem is leveraging multi-agent and multi-role systems. By following the "*divide et impera*" paradigm, breaking down

¹<https://pal-robotics.com/>

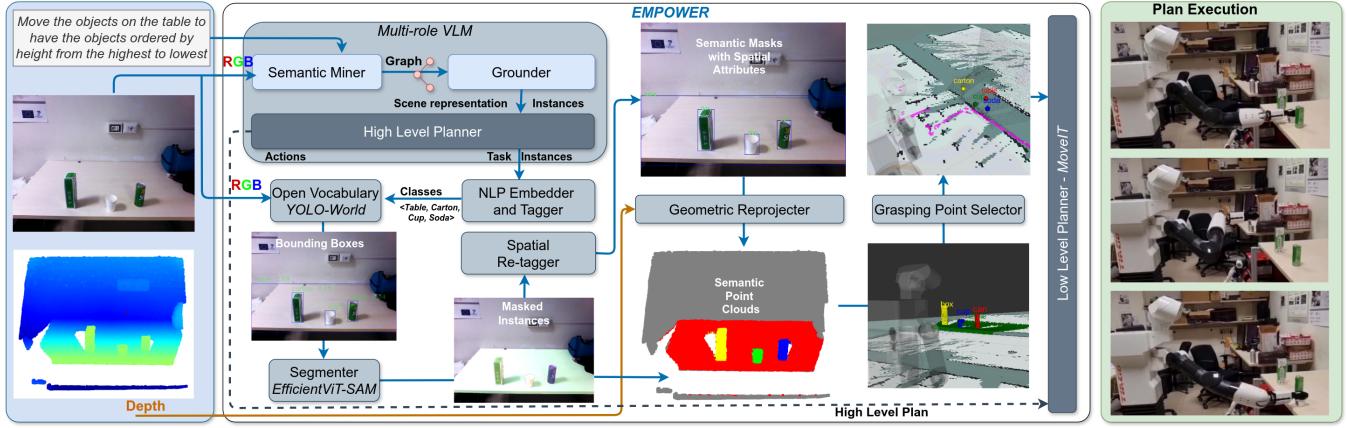


Fig. 2. Complete architecture of EMPOWER, from the task description to the execution of the plan in the world. The RGB image is used to extract a graph of the scene as long as the final plan and the object labels. These labels are then grounded via an NLP pipeline and reprojected onto the point clouds extracted from the depth image of the robot. Lastly, reference points are computed from these point clouds to facilitate actions in the world. Use case illustrated: order the objects on the table from the highest to the lowest.

a primary task into several sub-tasks that are then processed by multiple agents with distinct roles has produced more robust results compared to single-agent, single-role systems [31], [32], [33], [9], [34], [35], [10]. However, designing frameworks with multiple roles and agents increases the system complexity, making them less suitable for constrained or embedded applications. In our work, we demonstrate that the benefits gained from a multi-role architecture outweigh the additional complexity added to the system. Our system remains simple enough for fast execution on real robots in real-world settings. To the best of our knowledge, this is the first deployment of a multi-role LLM planner on a real robot.

C. Open-vocabulary systems

Recent advancements in open-vocabulary systems have contributed to bridging the semantic gap between textual descriptions and visual content, freeing models from the limitations of a fixed, closed vocabulary. Transformers have enabled open-vocabulary architectures in various tasks, including image segmentation [36], [37], [38], object detection [39], [40], [41] and scene parsing [42], [43], [44]. However, this level of generality comes at a cost: these models often require significant computational resources even during inference, making them less suitable for embedded applications. In our work, we use on-the-edge models like YOLO-World [45] and EfficientViT-SAM [46] to achieve an open-vocabulary architecture that does not compromise the overall system's performance.

Table I compares several state-of-the-art architectures, highlighting the presence of some key components: system embodiment in a robot (Embodied), visual assistance of the system (V.A.), open-vocabulary capabilities (O.V.), online execution (O.), and the presence of a multi-role subdivision (M.R.). To the best of our knowledge, our system is the first to integrate all of these components.

TABLE I
COMPARISONS BETWEEN DIFFERENT STATE-OF-THE-ART ARCHITECTURES AND OUR ARCHITECTURE

| Architecture | Embodied | V.A. | O.V. | O. | M.R. |
|-----------------------|----------|------|------|----|------|
| Huang et al. [8] | ✗ | ✗ | ✗ | - | ✗ |
| SayCan [19] | ✓ | ✗ | ✗ | - | ✗ |
| LLM-Planner [7] | ✗ | ✓ | ✗ | ✗ | ✗ |
| SayPlan [21] | ✓ | ✓ | ✗ | ✗ | ✗ |
| LLM-Grounder [23] | ✗ | ✓ | ✓ | ✗ | ✗ |
| LGX [22] | ✓ | ✓ | ✓ | - | ✗ |
| EMPOWER (ours) | ✓ | ✓ | ✓ | ✓ | ✓ |

III. METHODOLOGY

In this section, we describe the architecture of our system. Section III-A details the planner component, which utilizes a multi-role subdivision of tasks. Section III-B explains the NLP processing technique we used for grounding concepts in the scene and the open-vocabulary tools we adopted. Section III-C describes how we map the high-level grounded plan obtained in previous steps to low-level actions executable by the robot.

A. Multi-role planner

The multi-role planner module consists of three agents, chosen accordingly to [10]: a *Semantic-Knowledge Miner Agent* (SMK) that is specialized in obtaining semantic relationships between objects in the scene; a *Grounded-Knowledge Miner Agent* (GMK) that is focused on describing the environment by grounding the objects relevant to the task; a *Planner Agent* (P) which is responsible for generating the final sequence of actions needed to complete the task. For our implementation, the SMK and GMK roles are performed by GPT-4V [12], while the Planner role is handled by [11]. We selected these models due to their effectiveness in preliminary tests and their ability to perform efficiently without overloading the robotic system.

The agents work together to devise plans executable by a

TIAGo robot.

Fig. 2 illustrates the interaction between these models to produce the output plan.

SMK. Takes as input an image I from the robot’s RGB-D camera, the description of the task that we want to achieve d , and it outputs a semantic representation of the scene as a set of triples: $S = \langle head, relation, tail \rangle$, where the *relation* denotes the semantic connection between the *head* and the *tail* objects. This representation helps improve the robot’s knowledge, similar to an instantiated Knowledge Graph of the scene [47], and assists in grounding objects perceived by the camera.

GMK. Receives the triples from *SMK*, along with the image and task description. It produces a compact representation of the scene and suggests optimizations for the plan, such as renaming objects to ensure a unique representation. These unique names are used by the open-vocabulary grounder as identifiers.

P. Uses the task description and the output from *GMK* to generate the sequence of steps needed to accomplish the task. The planner is prompted to adhere to a predefined vocabulary of actions, facilitating the translation of the plan into low-level actions directly executable by the robot. More details on this aspect are in Section III-C. The planning process is designed to adhere to the Markov assumption, where each action’s outcome depends solely on the current state, determined through the LLM’s chain-of-thought reasoning from the initial state.

B. Open-vocabulary grounding

The *open-vocabulary grounding* component enables semantic grounding and discrimination of the objects necessary to achieve the desired task. In fact, given that multiple instances of the same object type might be present, it is crucial to distinguish their locations and roles w.r.t. the task.

The *SMK* agent’s response includes a graph of objects and their spatial relations. This information helps identify instances of objects and their grounding.

To determine object classes seen by the robot, we apply a Part-of-Speech (POS) tagging technique to extract nouns, filtering out adjectives that describe objects. We have chosen the nouns that respect the condition of central nouns, nominal subjects, or direct objects of the sentence. For instance, for an object described as a *recycling bin for paper*, we only need to know that the object is a *bin* to detect it: at this stage, every other information is not relevant. Therefore, the POS-tagger allows us to obtain for the object only the label that we need (*bin*). We use spaCy [48] as our POS-tagger of choice. Objects not previously encountered are added to a list that contains all the classes needed to solve the task (Algorithm 1). We use Word2Vec similarity embeddings to check whether an object is already in the list. The use of semantic similarity is necessary because several words can describe the same object in the scene. To compute the semantic similarity between two objects, we cross-check each word of the object obtained by the POS-tagger.

Algorithm 1 Classify Objects in Graph

Require: $Graph = \{\langle head, relation, tail \rangle, \dots\}$

Ensure: $Classes$

```

1: function ALREADYSEEN(object)
2:   for object_seen in Classes do
3:     if similarity(object_seen, object) <  $\tau$  then
4:       Classes.append(object)
5:     end if
6:   end for
7: end function
8: Classes  $\leftarrow []$ 
9: for relation in Graph do
10:   alreadySeen(POS_tagger(head))
11:   alreadySeen(POS_tagger(tail))
12: end for

```

$$\mathcal{E} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M \frac{w_i \cdot w_j}{\|w_i\| \|w_j\|}, N = |W_1| \text{ and } M = |W_2| \quad (1)$$

In Eq. 1, w_i and w_j represent the elements in W_1 and W_2 , which are the lists that contain the nouns given by POS-tagger that describe the two objects that are being compared. For every W_1 and W_2 pair, if the similarity score \mathcal{E} between two objects exceeds a threshold $\tau = 0.708$, the object is considered already detected. We have chosen this τ value using the approach presented in [49], where they obtain thresholds that vary according to the dimensionality of the embedding vector. This approach ensures the classes that YOLO-World can use during inference, which will then provide the bounding boxes of the objects: $\{b_i\} = YOLO\text{-World}(I, \{l_i\})$. The automatic procedure of selecting the classes to be recognized makes our architecture open-vocabulary. These bounding boxes serve two purposes: i. locating multiple instances of the same object using spatial relations; ii. generating binary masks for each object using EfficientViT-SAM $m_i = EfficientViT\text{-SAM}(b_i)$. Lastly, these masks are reprojected onto the point cloud from the depth image to create masked point clouds for each scene object.

C. Plan Actuator

The *Plan Actuator* component enables the robot to execute in the world the actions specified in the high-level plan. At this stage, we perform two key tasks: i. we *refine* high-level actions by mapping them with low-level primitives that the agent can perform; ii. we *ground* these actions, namely, we determine which objects the robot should interact with based on the output of the previous module. Since we are using a TIAGo robot with a gripper end-effector, we have identified a set of 5 low-level primitives directly executable by the robot: 1. NAVIGATE \langle to a location \rangle , 2. GRAB \langle an object \rangle , 3. DROP \langle an object \rangle , 4. PULL \langle an object and move back \rangle , 5. PUSH \langle an object with the base \rangle . All possible action names that can appear in the plan’s description are mapped to one of these 5 primitives so that at lower levels we have more control over what action is going

| Task Description: Throw away the objects in the corresponding recycling bin | | | | |
|---|-----------------|----------------------|------------------------|--|
| Object World Detection | Spatial Tagging | Semantic Point cloud | Geometric Reprojection | Plan |
| | | | | GRAB crumpled paper, DROP crumpled paper into recycling bin for paper, GRAB can, DROP can into recycling bin for plastic and metal |
| Task Description: Exit the room | | | | |
| Object World Detection | Spatial Tagging | Semantic Point cloud | Geometric Reprojection | Plan |
| | | | | GRAB trash can, DROP trash bin right to box, PUSH box away from the door, PULL door |
| Task Description: Order the shelf to have 2 objects per level | | | | |
| Object World Detection | Spatial Tagging | Semantic Point cloud | Geometric Reprojection | Plan |
| | | | | GRAB trophy, DROP trophy middle shelf, GRAB paper cup, DROP paper cup bottom shelf |
| Task Description: Give me the green jacket from the clothing rack | | | | |
| Object World Detection | Spatial Tagging | Semantic Point cloud | Geometric Reprojection | Plan |
| | | | | GRAB green jacket, DROP green jacket right to you |
| Task Description: Move the objects on the table to have the objects ordered by height from the highest to lowest | | | | |
| Object World Detection | Spatial Tagging | Semantic Point cloud | Geometric Reprojection | Plan |
| | | | | GRAB cup, DROP cup right to can |
| Task Description: Move the objects on the shelf in order to have for each level of the shelf only the objects made of the same material | | | | |
| Object World Detection | Spatial Tagging | Semantic Point cloud | Geometric Reprojection | Plan |
| | | | | GRAB paper cup, DROP paper cup bottom shelf, GRAB plastic ball, DROP plastic ball middle shelf, GRAB metal trophy, DROP metal trophy top shelf |

TABLE II

RESULTS OF EACH MODULE ON ALL THE SIX EXPERIMENTAL USE CASES. FIRST, THE OBJECTS IN THE WORLD ARE DETECTED THANKS TO THE OPEN-VOCABULARY DETECTOR. THEN, THE SPATIAL TAGGING NECESSARY FOR GROUNDING IS PERFORMED. A SEMANTIC POINT CLOUD IS THEN GENERATED FROM THE GROUNDING OUTPUT, AND IT IS USED TO POPULATE THE RVIZ SCENE. ON THE RIGHT, THE FINAL PLAN IS SHOWN. TO OBSERVE THE FULL EXECUTION OF THE PLANS, PLEASE REFER TO PROJECT'S WEBSITE

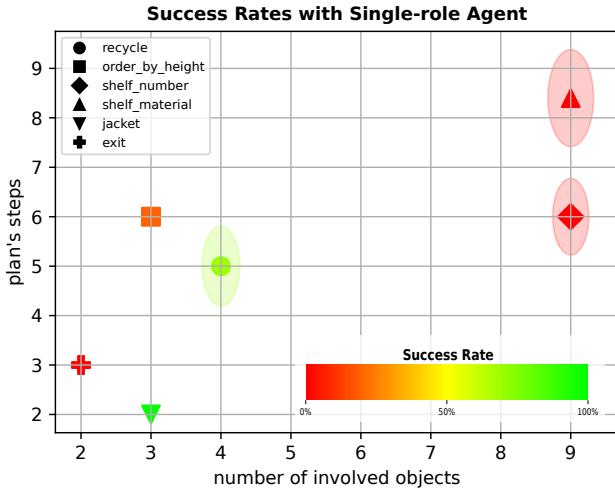


Fig. 3. Success Rates of the experiments in the single-role setup. Each marker represents one particular use case. The deviation from the mean on the number of the steps of the plans over 10 trials is also shown.

to be executed. To determine which objects the robot should interact with, we utilize the masked point clouds generated in the previous module. For each masked point cloud, we compute a representative point, the centroid. These centroids serve as the candidate locations of the grounded objects within the environment and act as the *grasping points* for the robot’s actions. Finally, we use the *MoveIt* [50] motion planning framework to deploy actions, setting the grasping point as the reference goal for the robot’s end-effector.

IV. EXPERIMENTAL RESULTS

This section details the experiments conducted in our laboratory using the robot.

A. Hardware specification

Our framework operates on the TIAGo robot, which is well-suited for indoor manipulation tasks due to its 7 degrees-of-freedom arm. The robot is equipped with a 640x480 RGB-D camera and uses a Parallel Gripper end-effector for enhanced grabbing capabilities. It is powered by 16GB of RAM, an Intel i7 processor, 512 GB of storage, and notably, it lacks a GPU, meaning all computations are carried out on the CPU.

We designed six representative use cases to evaluate the LLMs’ planning capabilities for indoor tasks involving multi-step reasoning and manipulation. These use cases are: *sort object by their height*, *grab a jacket on the coat rack*, *throw the objects in the right recycle bins*, *order the shelf to have 2 objects per level*, *order the shelf depending on the objects’ material*, *exit the room*.

B. Qualitative Results

We conducted 10 runs for each use case. Table II shows representative results from these runs. In particular, it includes the results of the *grounding* components, the semantic point cloud, their reprojections in the scene, and the plans

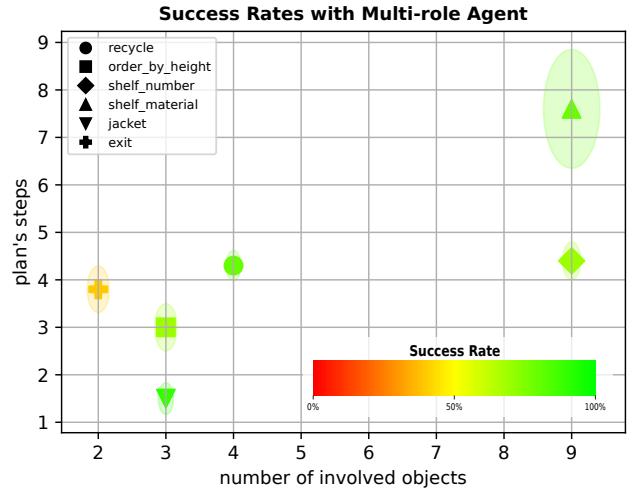


Fig. 4. Success Rates of the experiments in the multi-role setup. It is possible to see how the Success Rates are higher w.r.t. the ones of the single-role setting of Fig 3.

generated by the multi-role planner. Qualitatively, the bounding boxes and binary masks demonstrate high precision in object detection. This precision enables accurate calculation of grasping points from the semantic point cloud, ensuring that the robot interacts effectively with the intended objects and executes the plans smoothly.

C. Quantitative Results

To evaluate our system quantitatively, we used the Success Rate (SR) metric, which measures the percentage of successful plans across 10 runs. A plan is deemed successful if the goal is achieved at any point and is not undone by subsequent actions; on the contrary, it is considered unsuccessful if subsequent actions nullify the achieved goal. In Figs. 3, 4 display the results. Our comparison study shows that the multi-role architecture achieves an average SR of 73%, compared to 34% for the single-role architecture, highlighting the effectiveness of the multi-role approach for complex reasoning tasks. An analysis of the average number of steps per plan is also shown in Fig. 5, and it indicates that plans generated by the multi-role setup are generally shorter and more accurate.

D. Temporal Analysis

State-of-the-art systems for open-vocabulary object detection and phrase grounding, such as GLIP [51], require CUDA acceleration and thus cannot be used on systems without GPU. Our pipeline, however, provides comparable qualitative performance on CPU-only robots and operates efficiently on embodied systems.

We conducted a temporal analysis of our framework, examining the average execution times for the NLP pipeline, YOLO-World, and EfficientViT-SAM. The NLP pipeline takes on average in 0.8 seconds, YOLO-World takes 0.7 seconds, and EfficientViT-SAM processes one mask in 0.9 seconds. Thus, the total time for a full execution ranges from

TABLE III

PLANS' SUCCESS RATES IN SINGLE AND MULTI-ROLE ARCHITECTURES

| Task | Single-role | Multi-role |
|-----------------|-------------|-------------|
| recycle | 0.7 | 0.8 |
| order_by_height | 0.2 | 0.7 |
| shelf_number | 0.0 | 0.8 |
| shelf_material | 0.0 | 0.8 |
| jacket | 1.0 | 0.9 |
| exit | 0.0 | 0.4 |
| Average SR | 0.32 | 0.73 |

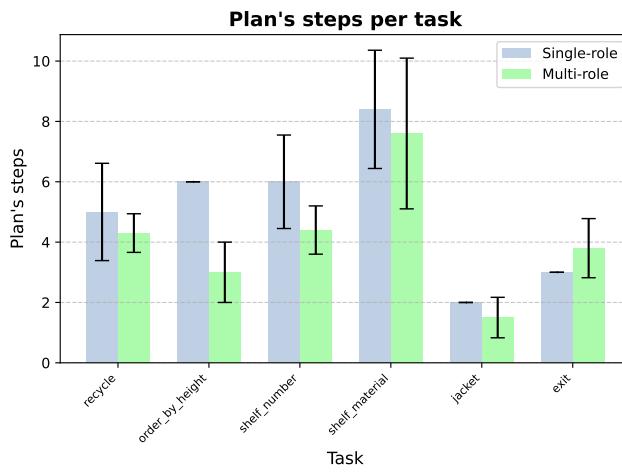


Fig. 5. Comparison of average number of steps per plan between the multi-role and the single-role setup. We can see how the plans obtained by the multi-role are shorter on average than the ones of the single-role.

2.7 to 8.1 seconds, depending on the task. Therefore, the average execution time would take about 9.6 seconds in the worst case, plus the GPT API call time, which is reasonable for real-world planning tasks.

V. DISCUSSION

This paper offers contributions both in engineering and research, advancing robot embodiment and highlighting how improved spatial and semantic awareness can enhance planning, especially in unfamiliar environments.

From an engineering perspective, the EMPOWER framework integrates pre-trained models with a multi-role planning structure, enabling real-time, open-vocabulary grounding and planning on systems like the TIAGo robot. This approach demonstrates practical, CPU-based solutions for complex tasks, making it suitable to a wide range of robotic platforms.

Research-wise, this work takes a first step toward more advanced robot embodiment, where robots gain a deeper semantic understanding of their environment. The multi-role architecture supports online execution, showing that enhanced awareness helps robots adapt and plan effectively, even in new settings. This shift towards open-vocabulary planning aids in better handling unexpected scenarios.

This paper underscores the value of combining spatial and semantic awareness in robotics, and we hope to pave the way for future developments in autonomous, intelligent systems.

VI. CONCLUSIONS

In this paper, we introduced EMPOWER, a framework designed for open-vocabulary, online grounding, and planning in embodied agents, specifically aimed at addressing real-world robotic planning challenges. By leveraging efficient pre-trained foundation models within a multi-role structure, we developed a multi-role planning system that, to the best of our knowledge, is the first to support online execution in real-time settings.

Our approach enables embodied agents to plan effectively in an open-vocabulary environment, utilizing semantic understanding derived from NLP techniques applied to elements within the scene.

We evaluated our framework at multiple levels, first demonstrating the enhanced high-level planning capabilities provided by our multi-role VLM-based architecture compared to a single-role approach. We then showcased the system's overall planning capabilities, from high-level strategy to low-level execution, by testing the Success Rate of a TIAGo robot performing tasks in real environments.

The evaluation was conducted across six real-life scenarios, including tasks that require interpretative skills, which are particularly challenging for robots. For example, sorting and recycling waste by material type involves distinguishing between objects with similar appearances but different semantic meanings.

Quantitative results highlight the effectiveness of our approach, achieving an average Success Rate of 0.73 across the selected challenging use cases performed by the TIAGo robot.

ACKNOWLEDGEMENTS

This work has been carried out while Francesco Argenziano and Michele Brienza were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome.

This work has been partially supported by PNRR MUR project PE0000013-FAIR.

The research reported in the paper was partially supported by the project "Tech4You (ECS00000009) - Spoke 6", under the NRRP MUR program funded by the NextGenerationEU.

REFERENCES

- [1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.
- [2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
- [3] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, “Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–21.
- [4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020.

- [5] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [6] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [7] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, “Llm-planner: Few-shot grounded planning for embodied agents with large language models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2998–3009.
- [8] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 9118–9147.
- [9] Y. Talebirad and A. Nadiri, “Multi-agent collaboration: Harnessing the power of intelligent llm agents,” *arXiv preprint arXiv:2306.03314*, 2023.
- [10] M. Brienza, F. Argenziano, V. Suriani, D. D. Bloisi, and D. Nardi, “Multi-agent planning using visual language models,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.05478>
- [11] OpenAI, “Gpt-4 technical report,” 2024.
- [12] ———, “Gpt-4v(ision) system card,” 2023.
- [13] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” 2023.
- [14] G. Team, “Gemini: A family of highly capable multimodal models,” 2023.
- [15] D. Shah, B. Osiński, S. Levine *et al.*, “Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action,” in *Conference on robot learning*. PMLR, 2023, pp. 492–504.
- [16] B. Zhang and H. Soh, “Large language models as zero-shot human models for human-robot interaction,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7961–7968.
- [17] L. Nwankwo and E. Rueckert, “The conversation is the command: Interacting with real-world autonomous robot through natural language,” 2024.
- [18] Z. Luan, Y. Lai, R. Huang, X. Lan, L. Chen, and B. Chen, “Automatic robotic development through collaborative framework by large language models,” 2024.
- [19] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.
- [20] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, “Palm-e: An embodied multimodal language model,” *arXiv preprint arXiv:2303.03378*, 2023.
- [21] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenberhauf, “Sayplan: Grounding large language models using 3d scene graphs for scalable task planning,” *arXiv preprint arXiv:2307.06135*, 2023.
- [22] V. S. Dobrala, J. F. Mullen Jr, and D. Manocha, “Can an embodied agent find your” cat-shaped mug”? llm-guided exploration for zero-shot object navigation,” *arXiv preprint arXiv:2303.03480*, 2023.
- [23] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. Chai, “Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent,” *arXiv preprint arXiv:2309.12311*, 2023.
- [24] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, “When and why vision-language models behave like bags-of-words, and what to do about it?” in *The Eleventh International Conference on Learning Representations*, 2022.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.
- [26] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [27] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.
- [28] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models,” *arXiv preprint arXiv:2310.08864*, 2023.
- [29] L. Bermudez, ““overview of embodied artificial intelligence,” 2021.
- [30] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, “Tree of thoughts: Deliberate problem solving with large language models,” 2023.
- [31] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang, “Autogen: Enabling next-gen llm applications via multi-agent conversation framework,” *arXiv preprint arXiv:2308.08155*, 2023.
- [32] S. Rasal, “Llm harmony: Multi-agent communication for problem solving,” *arXiv preprint arXiv:2401.01312*, 2024.
- [33] M. Shanahan, K. McDonell, and L. Reynolds, “Role play with large language models,” *Nature*, vol. 623, no. 7987, pp. 493–498, 2023.
- [34] E. Musumeci, M. Brienza, V. Suriani, D. Nardi, and D. D. Bloisi, “Llm based multi-agent generation of semi-structured documents from semantic templates in the public administration domain,” 2024.
- [35] Z. Luan, Y. Lai, R. Huang, S. Bai, Y. Zhang, H. Zhang, and Q. Wang, “Enhancing robot task planning and execution through multi-layer large language models,” *Sensors*, vol. 24, no. 5, p. 1687, 2024.
- [36] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, “Scaling open-vocabulary image segmentation with image-level labels,” in *European Conference on Computer Vision*. Springer, 2022, pp. 540–557.
- [37] J. Qin, J. Wu, P. Yan, M. Li, R. Yuxi, X. Xiao, Y. Wang, R. Wang, S. Wen, X. Pan *et al.*, “Freeseg: Unified, universal and open-vocabulary image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19446–19455.
- [38] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, “Open-vocabulary semantic segmentation with mask-adapted clip,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7061–7070.
- [39] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, “Open-vocabulary object detection using captions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 393–14 402.
- [40] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, “Open-vocabulary object detection via vision and language knowledge distillation,” *arXiv preprint arXiv:2104.13921*, 2021.
- [41] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, “Simple open-vocabulary object detection,” in *European Conference on Computer Vision*. Springer, 2022, pp. 728–755.
- [42] R. Ding, J. Yang, C. Xue, W. Zhang, S. Bai, and X. Qi, “Pla: Language-driven open-vocabulary 3d scene understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7010–7019.
- [43] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa *et al.*, “Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning,” *arXiv preprint arXiv:2309.16650*, 2023.
- [44] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li, “Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation,” *arXiv preprint arXiv:2402.15487*, 2024.
- [45] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, “Yolo-world: Real-time open-vocabulary object detection,” 2024.
- [46] Z. Zhang, H. Cai, and S. Han, “Efficientvit-sam: Accelerated segment anything model without performance loss,” 2024.
- [47] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier *et al.*, “Knowledge graphs,” *ACM Computing Surveys (Csur)*, vol. 54, no. 4, pp. 1–37, 2021.
- [48] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” *To appear*, 2018.
- [49] N. Rekabsaz, M. Lupu, and A. Hanbury, “Exploration of a threshold for similarity based on uncertainty in word embedding,” 04 2017, pp. 396–409.
- [50] D. Coleman, I. Sucan, S. Chitta, and N. Correll, “Reducing the barrier to entry of complex robotic software: a moveit! case study,” *arXiv preprint arXiv:1404.3785*, 2014.
- [51] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, “Grounded language-image pre-training,” 2022.