Grounding Large Language Models for Long-horizon Robot Mission Planning in OsmAG

Zhang Jitian, Ren Wanqing, Ma Xu

A project of the 2024 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University
https://robotics.shanghaitech.edu.cn/teaching/robotics2024

***Abstract*—Large Language Models (LLMs) have demonstrated impressive language understanding abilities and have been applied to a variety of domains beyond traditional language tasks. These include dialogue systems, visual understanding, reasoning, code generation, embodied reasoning, and robot control. Building on the significant advancements in LLMs for comprehending complex natural language instructions, this project seeks to investigate the use of LLMs in robot mission planning and autonomous navigation, specifically within the information-rich and structured environment of university campuses. The approach involves integrating the text-based map representation, osmAG, with the high-level understanding capabilities of LLMs. By adding descriptive tags to osmAG and utilizing LLMs to interpret these descriptions, we aim to enhance navigational tasks. This method enables robots to handle complex, description-based queries and perform comprehensive path planning based on the map's topological structure.**

I. Introduction

"Make me a coffee and place it on my desk." The successful execution of such a seemingly straightforward command remains a daunting task for today's robots. The associated challenges permeate every aspect of robotics, encompassing navigation, perception, manipulation, and high-level task planning. However, recent advances in Large Language Models (LLMs) have led to significant progress in incorporating common sense knowledge into robotics. This enables robots to plan complex strategies for a diverse range of tasks that require substantial background knowledge and semantic comprehension.

This project focuses on integrating LLMs into robotic systems, specifically for mission planning and autonomous navigation in complex and structured environments like university campuses. We employ a text-based map representation, osmAG, which contains both spatial structural information and descriptive information that can be interpreted by LLMs. By introducing a "description" tag in osmAG and leveraging the advanced understanding and reasoning capabilities of LLMs, our system can execute complex navigational tasks based on users' natural language instructions.

Specifically, this research explores two application scenarios: description-based task planning and topology-based path planning. In the first scenario, LLMs use the provided descriptive tags to determine the best destination. In the second, LLMs utilize the topological structure in osmAG to plan paths that include multiple target locations. Both approaches demonstrate the vast potential of LLMs in interpreting spatial descriptions and executing task planning based on those descriptions.

Through this integration, we aim to develop a new type of robotic system capable of understanding complex task requirements, autonomously planning routes, and interacting with human users in a more natural and intuitive manner. This will not only advance the development of robotic technology in autonomous navigation but also open new possibilities for human-robot interaction, enabling robots to play a more significant role in education, research, and daily life.

In an era marked by technological advancements, robotics technology is becoming ever more prevalent across various sectors, enhancing our daily lives and work with new conveniences and possibilities. Inspired by our keen interest in Large Language Models and fueled by our participation in robotics courses, this project aims to apply our deep understanding of robotics principles to innovative task planning for robots.

II. Related Work

[1] Large language models (LLMs) have undergone significant expansion and have been increasingly integrated across various domains. Notably, in the realm of robot task planning, LLMs harness their advanced reasoning and language comprehension capabilities to formulate precise and efficient action plans based on natural language instructions.

[2] This paper demonstrates how ChatGPT can be used in a few-shot setting to convert natural language instructions into a sequence of executable

robot actions. The paper proposes easy-to-customize input prompts for ChatGPT that meet common requirements in practical applications, such as easy integration with robot execution systems and applicability to various environments while minimizing the impact of ChatGPT's token limit.

[3] There are some attempts of ChatGPT to robots, however, general usage of ChatGPT in robotics presents certain drawbacks, such as the inability to guarantee system stability and safety in the execution code it generates. ChatGPT's responses can vary for the same task, leading to unpredictability in outcomes. This inconsistency makes direct integration of ChatGPT into robotic manipulation loops challenging.
This paper introduces RobotGPT, an innovative decision framework designed for robotic manipulation, addressing these limitations by emphasizing stability and safety. The framework leverages ChatGPT's problem-solving capabilities while ensuring reliability through a structured prompt and a robust learning model.
Moreover, the paper outlines an effective prompt structure with a self-correction module and experiments on tasks of varying difficulty to explore ChatGPT's capabilities in robotic tasks. It proposes a novel framework where an agent, rather than executing ChatGPT-generated code directly, learns the planning strategies from ChatGPT, thus enhancing system stability. This approach does not solely rely on ChatGPT for direct action but uses it as a guide for an agent to learn and execute tasks, offering a solution to the limitations and security risks associated with direct code execution from language models.

[4] The letter presents robust lifelong indoor LiDAR global localization and pose tracking using the hierarchical, semantic topometric Open Street Map Area Graph map representation.
[5] The article presents osmAG, a hierarchical semantic topometric map format based on the OSM XML format, designed to enable efficient path planning for mobile robots in complex indoor and outdoor environments.

[6]The paper delves into the utilization of Large Language Models (LLMs) to enhance map comprehension for mobile robots, with a focus on the osmAG map representation. The document highlights the importance of LLMs, such as ChatGPT and LLaMA, in providing extensive general knowledge capabilities that can aid in the development of intelligent robots.

The introduction sets the stage by illustrating a real-life scenario where a delivery robot faces an obstacle due to a closed intersection, emphasizing the need for robots to comprehend map hierarchy and topology for effective navigation. The authors stress the potential of LLMs to process real-time information and plan paths based on prior knowledge, thereby enabling robust decision-making in dynamic environments.

To improve LLM performance in understanding osmAG maps, the paper discusses three levels of description in the prompt, each varying in the depth of explanation and inclusion of examples. Additionally, the authors introduce osmAG variants that simplify the representation of connections between areas, making relationships more explicit for LLMs. By refining prompts, adopting effective map representations, and fine-tuning LLMs, the study aims to enhance map comprehension capabilities for robotic applications.

The methodology section outlines the creation of datasets for fine-tuning LLMs and the process of fine-tuning the LLaMA2 model to surpass the performance of ChatGPT-3.5. The use of Low Rank Adaptation (LoRA) accelerates the training process, resulting in a more efficient model for map understanding tasks. The authors emphasize the importance of prompt engineering, which involves providing a detailed task description and osmAG context to optimize LLM responses.

Furthermore, the paper discusses the limitations of token size and the decision to omit metric information from osmAG, focusing solely on connections and hierarchy for evaluating LLM proficiency. By employing hand-drawn layout templates, the study aims to assess LLMs' understanding of topological and hierarchical relationships without the need for detailed shape information.

In conclusion, the paper underscores the potential of LLMs in enhancing map comprehension for mobile robots, paving the way for improved navigation, decision-making, and adaptability in dynamic environments. By leveraging advanced language models and innovative map representations, researchers can unlock new possibilities for intelligent robotic systems in various real-world scenarios.

[7] This paper introduces a novel method of integrating OpenAI's ChatGPT with robotics to handle a wide range of tasks. By using Large Language Models like ChatGPT, it translates natural language commands into robot actions through detailed prompt engineering and a specialized functional library. The study shows ChatGPT's strengths in understanding detailed dialogues, creating code, and making decisions through interactions with users. It also showcases how ChatGPT can manage tasks involving space-time reasoning, manipulation, and flying based on textual commands and visual data.

[8]Recent advances in large language models have

highlighted their potential to encode massive amounts of semantic knowledge for long-term autonomous decision-making, positioning them as a promising solution for powering the cognitive capabilities of future home-assistant robots. Inspired by ChatGPT for Robotics, the authors propose a novel framework that integrates a large language model with visual perception and motion planning modules for robotic grasping.

Their approach uses natural language prompts, combined with visual information, to provide high-level guidance to the robot and enable accurate and efficient object grasping in unstructured environments. The perception module processes visual observations, extracts semantic information, and feeds it to the large language model with user instructions. Based on this input, the large language model makes decisions that inform the perception module to calculate the robot end-effector's grasping pose. The motion planning module then generates a trajectory for the robot to complete the action and grasp the target object.

This proposed framework can handle personalized user instructions and perform tasks more effectively in home scenarios, unlike traditional methods that focus solely on generating stable grasps. It understands imprecise language commands, allowing the robot to grasp objects without the need for fine-tuning or demonstration, significantly lowering the deployment costs and making it well-suited for real-world applications.

[9] The authors introduce SayPlan, which leverages 3D Scene Graph (3DSG) representations to handle complex, multi-room, and multi-floor environments. This is achieved through a hierarchical process that allows for efficient semantic searches within a collapsed representation of the full graph, reducing the planning horizon via classical path planning, and refining plans iteratively with feedback from a scene graph simulator.

The key contributions of the paper include:

1) Hierarchical Semantic Search: Utilizing the hierarchical nature of 3DSGs to enable LLMs to perform semantic searches for relevant subgraphs efficiently.
2) Integrated Path Planning: Combining LLMs with classical path planners to manage the navigational components of task planning.
3) Iterative Replanning: Introducing a replanning pipeline that uses feedback from a scene graph simulator to correct infeasible actions and avoid planning failures.

The effectiveness of SayPlan is demonstrated through experiments in large-scale environments, including scenarios with up to three floors and 36 rooms, showing its capability to execute long-horizon task plans from natural language instructions. The approach is validated with real robot demonstrations, emphasizing its practical applicability in real-world settings.

III. System Design

A. System Description

We download the html file of faculty website and shorten it into a brief text file, then start a conversation with LLM by providing a text file that containing information of professors, such as offices labs and so on. Secondly we teach LLM some basic functions that we can use, go-to(), wait(), hear(), stop-hear() as well as parameters of them.

It become easy to program a simple routine as we don't have to care about the navigation details. We simply need to extract "passage" number of the particular room in osmag map and put it into go-to() function then we can go to that room carefree.

And here we can set up an example of cakes delivery task.

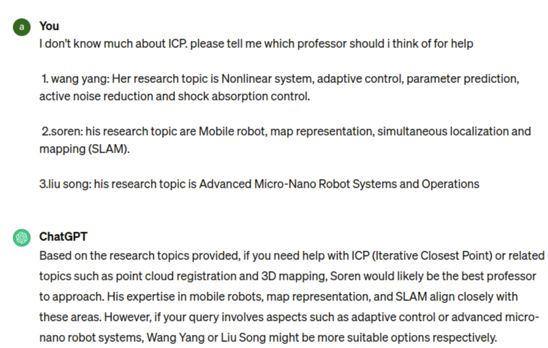
1) Long task planning based LLM: We want to add a new tag named "description" on osmAG, as show in Fig 1. It is some information about this area. This can be understood by LLM and do some reasoning based on this. For example, I can take to LLM "I don't know much about ICP. Please find a professor to explain it to me. " Then the robot can take me to soren office.

```
<way id='-117360' action='modify' visible='true'>
  <nd ref='-149988' />
  <nd ref='-149989' />
  <tag k='height' v='3.2' />
  <tag k='indoor' v='room' />
  <tag k='level' v='2' />
  <tag k='name' v='E1d-13' />
  <tag k='osmAG:areaType' v='room' />
  <tag k='osmAG:type' v='area' />
  <tag k='description' v='This is soren office, his research topic are Mobile robot,
                           map representation, simultaneous localization and mapping (SLAM).' />
</way>
```

Figure 1. osmAG

Its workflow is like this:

1) LLM acts as a high-level planner. Because it can have a good comprehension of the osmAG and description gived by us. As show in Fig 8, LLM can inferre that I need to go to your office instead of others'. Then let LLM return the ID of your office in osmAG.


You
I don't know much about ICP. please tell me which professor should i think of for help

1. wang yang: Her research topic is Nonlinear system, adaptive control, parameter prediction, active noise reduction and shock absorption control.

2.soren: his research topic are Mobile robot, map representation, simultaneous localization and mapping (SLAM).

3.liu song: his research topic is Advanced Micro-Nano Robot Systems and Operations

ChatGPT
Based on the research topics provided, if you need help with ICP (Iterative Closest Point) or related topics such as point cloud registration and 3D mapping, Soren would likely be the best professor to approach. His expertise in mobile robots, map representation, and SLAM align closely with these areas. However, if your query involves aspects such as adaptive control or advanced micro-nano robot systems, Wang Yang or Liu Song might be more suitable options respectively.

Figure 2. ChatGPT

2) Execute a low-level path planner. Give a path from my current location to soren office based on the ID obtained in the previous step.
3) Finally let the robot execute the path and take me to the office

B. System Evaluation

1) Scenario and Testing: We first provide the large language model with a specific scenario to test its application in mission planning. We assign it a role, assuming it is a robot with the task of delivering ten cakes to the professors on the second floor of the Star Center. The robot can execute action commands including *go to*, *wait*, *hear*, *stop hear*, and *done*. During the delivery process, it needs to adjust its delivery plan based on sounds from the environment. We will play the role of people interacting with it in the environment. The large language model needs to understand our instructions and make decisions. Each time it executes the *hear* command, the robot will stop generating and wait for people to input commands via the keyboard, thus training it to replan its tasks.

Example Experimental Steps:

1) The robot receives the command and then go to the professor's office or labs.
2) After arriving at the designated rooms, the robot uses *speak()* to inform the people in the room what it came for.
3) Then it executes the *wait()* command and *hear* command for instructions from people. It may change its planning based on human requirements. If no one in the room it will automatically go to the next place.
4) Based on the conversation, it can decide if the task has been done and return to its starting point by executing the *done* command and generates the log and conversations during the mission.

2) Definition of Success Criteria:

- The robot successfully receives user commands and reacts appropriately.
- The robot is able to replan tasks based on sound inputs from the environment and complete the delivery tasks correctly.
- In the simulated environment, the robot successfully completes the delivery of ten cakes.

C. Experiments Result

- Background setup by telling room number format, room information and basic functions.

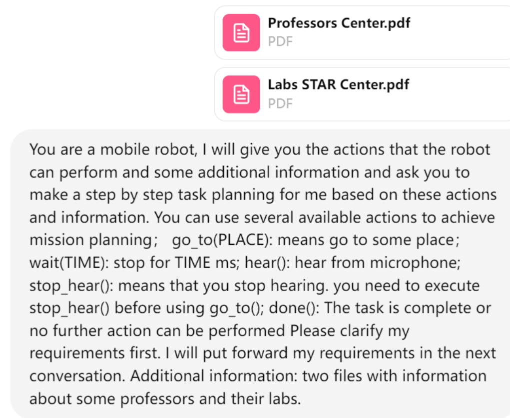


Figure 3. Teach ChatGPT Rules and Provide Rooms Info

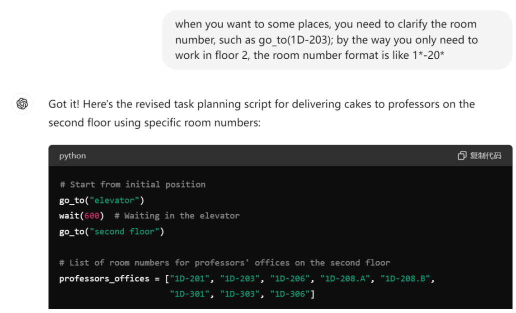


Figure 4. Let GPT remember Room Numbers

- Scenario: Deliver the mail to Prof. Jiahao Chen, as well as 9 pieces of pizza to him. Then deliver some fried chicken and potato chips to professors' office rooms in floor 2.

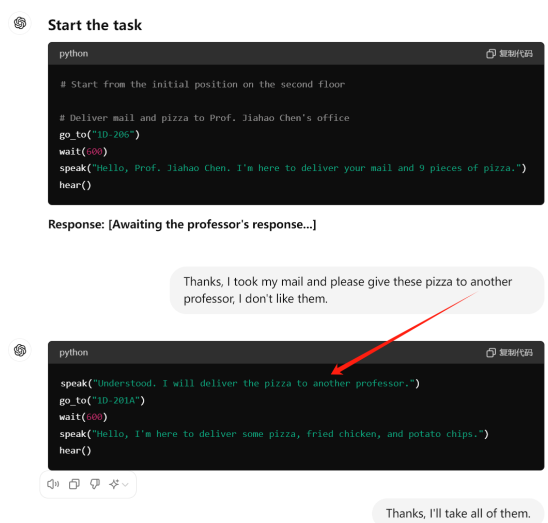


Figure 5. Assign a Mission

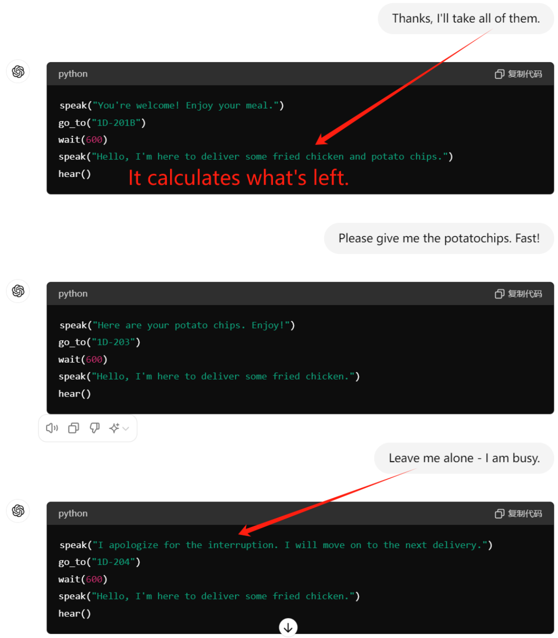


Figure 6. Planning based on requirements

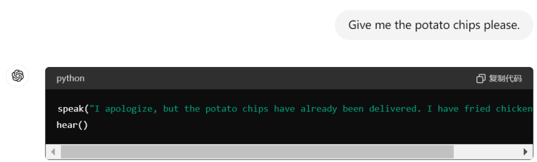


Figure 7. It remembers objects information

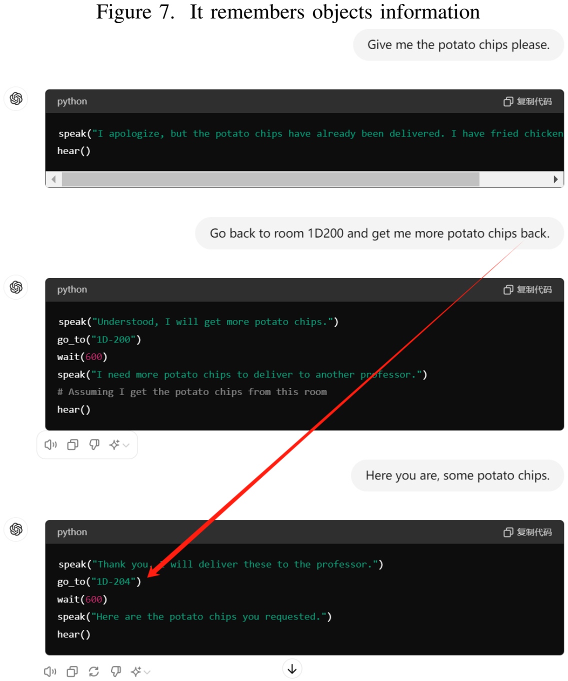


Figure 8. It remembers core missions

IV. Evaluation

Since er cannot operate on a real robot, we evaluate it by counting how many times it forgets room and professor information or goes to a wrong door in a test:

Task	Food 1	Food 2	Food 3	Mail and Food
Forget Info	0	1	1	1
Wrong Door	0	1	0	0

It sometimes forgets names of professors, if you let it goes to someone's room, it may go to the wrong door. If it remembers then it hardly ever goes a wrong way.

V. Conclusions

In this project, we successfully utilized a large language model to achieve mission planning and tested its capability as a delivery robot in a simulated environment. By assigning the robot the task of delivering ten cakes to professors' offices on the second floor of the Star Center, we validated the large language model's ability to interact and make decisions in complex environments.

During the experiments, the large language model correctly understood and executed user commands such as *go to*, *wait*, *hear*, *stop hear*, and *done*, and it replanned tasks based on received sound inputs. The model demonstrated the ability to stop at each task node, accept new instructions, and adjust the task plan accordingly. In the simulated environment, the robot successfully completed the delivery of ten cakes, proving the effectiveness and flexibility of the large language model.

Our test results indicate that the large language model can accurately receive user commands and respond appropriately. Additionally, it can replan tasks through interaction with users and successfully execute them. This showcases the potential of large language models in real-world applications, especially in tasks requiring dynamic plan adjustments and responses to environmental changes.

Future work can focus on the following areas:

1) **Enhancing Planning Capabilities:** Design more scenarios for the large language model to practice, thereby improving its planning and decision-making abilities in more complex tasks.
2) **Implementation on Physical Robots:** Deploy our mission planning system onto actual robots.

However we also encounters with some problems:

1) The first problem is that LLM can not handle with such large information like osmAG map or processing unminified web page html file. We need some pre-process so that LLM can better comprehend better and remember well.
2) The second problem is that we haven't received the navigation API yet, so we don't have a chance to try out on a real robot.

References

[1] J. Wang, Z. Wu, Y. Li, H. Jiang, P. Shu, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang, Y. Yao, X. Liu, H. Zhao, Z. Liu, H. Dai, L. Zhao, B. Ge, X. Li, T. Liu, and S. Zhang, "Large language models for robotics: Opportunities, challenges, and perspectives," 2024.

[2] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Chatgpt empowered long-step robot control in various environments: A case application," *IEEE Access*, vol. 11, p. 95060–95078, 2023. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2023.3310935

[3] Y. Jin, D. Li, Y. A, J. Shi, P. Hao, F. Sun, J. Zhang, and B. Fang, "Robotgpt: Robot manipulation learning from chatgpt," 2023.

[4] F. Xie and S. Schwertfeger, "Robust lifelong indoor lidar localization using the area graph," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 531–538, 2024.

[5] D. Feng, C. Li, Y. Zhang, C. Yu, and S. Schwertfeger, "osmag: Hierarchical semantic topometric area graph maps in the osm format for mobile robotics," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2023, pp. 1–7.

[6] F. Xie and S. Schwertfeger, "Empowering robotics with large language models: osmag map comprehension with llms," 2024.

[7] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," 2023.

[8] J. Liao, H. Zhang, H. Qian, Q. Meng, Y. Sun, Y. Sun, W. Song, S. Zhu, and J. J. Gu, "Decision-making in robotic grasping with large language models," in *International Conference on Intelligent Robotics and Applications*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264307891

[9] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=wMpOMO0Ss7a