arXiv:2403.08228v2 [cs.RO] 1 Aug 2024

Empowering Robot Path Planning with Large Language Models: osmAG Map Topology & Hierarchy Comprehension with LLMs

Fujing Xie*, Sören Schwertfeger*†

***Abstract*— Large Language Models (LLMs) have demonstrated great potential in robotic applications by providing essential general knowledge. Mobile robots rely on map comprehension for tasks like localization and navigation. In this paper, we explore enabling LLMs to comprehend the topology and hierarchy of Area Graph, a text-based hierarchical, topometric semantic map representation utilizing polygons to demark areas such as rooms or buildings. Our experiments demonstrate that with the right map representation, LLMs can effectively comprehend Area Graph's topology and hierarchy. After straightforward fine-tuning, the LLaMA2 models exceeded ChatGPT-3.5 in mastering these aspects. Our dataset, dataset generation code, fine-tuned LoRA adapters can be accessed at https://github.com/xiefujing/LLM-osmAG-Comprehension.**

I. Introduction

Recent years have seen a growing interest in Large Language Models (LLMs) like ChatGPT[1] and LLaMA[2]. Real-life robots often face unpredictable situations, such as a campus delivery robot blocked by a closed intersection for pipe repair, depicted in Fig. 1. Despite the construction notice being posted publicly, the robot was unaware. Integrating general knowledge with real-time data from public notice boards through LLMs could significantly improve navigation and decision-making. For this integration to be effective, the robot's 'brain' needs to grasp the map's hierarchy and topology.

In mobile robotics, maps are essential foundational knowledge. Common robotic map formats include 2D occupancy grids, 3D point clouds, and visual approaches like bag of words, but these are suboptimal for LLM integration due to LLMs' text-based processing and token limitations. We propose using osmAG[6] (Area Graph[3][4] in OpenStreetMap format) for path planning in future mobile LLM-robot systems due to its advantages:

1) osmAG is stored in text format, making it naturally readable by LLMs.
2) osmAG only stores permanent structures, ensuring long-term stability.
3) osmAG can be easily generated from 3D point clouds[5], 2D occupancy grid maps[3][4], or CAD files[6].
4) Conventional robotic localization[7] and path planning[6] algorithms based on osmAG have been developed, making LLM behavior easy to monitor and verify, thereby enhancing safety.
5) osmAG is easily visualized through JOSM and ROS's rviz, enabling intuitive human interaction with the map.

The graph nodes of osmAG represent physical areas like rooms, while edges, termed passages, are door line segments of area polygons connecting adjacent areas. For brevity, the details of osmAG are omitted in this paper. Fig. 2 presents a JOSM (Java OpenStreetMap editor) visualization of a osmAG map, as discussed in Section IV-E. The objective of this paper is to show that osmAG can be utilized by LLMs

Notice about Emergency Repair of Sewage Pipes
on the East Side of the Silk Road Canteen and Road Closures

Dear all,

Due to the emergency repair of the sewage pipe under the road on the east side of the Silk Road Canteen, the road between the east side of the Silk Road Canteen and Student Apartment Building 2 must be excavated and renovated (as marked in red in the picture below). During the repair, the road will be closed for 6 days (January 19 – January 24, 2024).

There will be noise and machinery works. Please stay away and go around the area to avoid accidental injuries.

We apologize for any inconvenience this may cause!

Fig. 1. The figure above depicts a real-life situation encountered by a 3rd-party delivery robot on our University campus, where it is blocked by an intersection closure. Below the notification sent by Office of General Services announcing this closure is shown.

*The authors are with the Key Laboratory of Intelligent Perception and Human-Machine Collaboration – ShanghaiTech University, Ministry of Education, China. {xiefj,soerensch}@shanghaitech.edu.cn

† Sören Schwertfeger is the corresponding author.

This work has been partially funded by the Shanghai Frontiers Science Center of Human-centered Artificial Intelligence. This work was also supported by the Science and Technology Commission of Shanghai Municipality (STCSM), project 22JC1410700 "Evaluation of real-time localization and mapping algorithms for intelligent robots". The experiments of this work were supported by the core facility Platform of Computer Science and Communication, SIST, ShanghaiTech University.

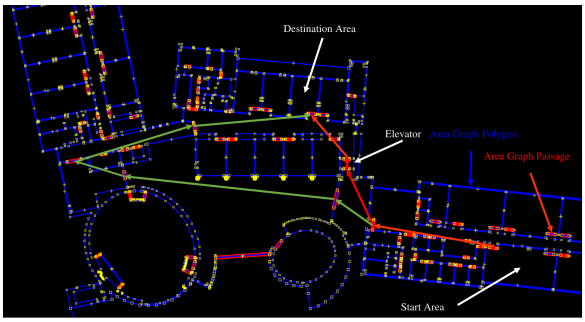

Fig. 2. osmAG visualization in JOSM displays areas in blue polygons, passages in red lines, and nodes coordinates in yellow dots. ChatGPT-4 identifies the shortest path via a two-way elevator in red. After given an elevator maintenance notice, it recommends an alternate route in green.

for applications in map-related mobile robotics tasks such as path planning.

Research on LLMs in mathematics has shown their challenges with numbers[8][9]. Considering that humans often omit metric details in path planning discussions, as noted in the notification in Fig. 1, excluding these metrics from maps does not diminish the system's effectiveness. Therefore, this paper focuses only on the topological and hierarchical properties of osmAG, disregarding its metric information.

For proprietary LLMs like ChatGPT, we adjust prompts and osmAG variants to evaluate their comprehension using datasets. For open-source LLMs, we fine-tune LLaMA2 models, achieving over 90% success rate in map comprehension tasks. Section IV-E highlights ChatGPT-4's ability to understand real-life scenarios, demonstrating the utility of LLMs with osmAG in robotics. Our contributions are as follows:

- We propose utilizing osmAG as the map representation for future mobile LLM-robot systems.
- We offer scripts to convert osmAG for LLMs and generate datasets for fine-tuning on topology and hierarchy understanding.
- We employ efficient fine-tuning to achieve better performance with our LLaMA2 model than ChatGPT-3.5 on our tasks.
- Our dataset, dataset generation scripts, and LLaMA2 adapters are made publicly available to encourage further research and collaboration in the field.

II. Related Works

A. Integrated Robotics and Navigation with LLMs and Maps

Research in robotics integrates natural language models like PaLM-E[10], which uses real-world sensor data for better decision-making, and LLMs

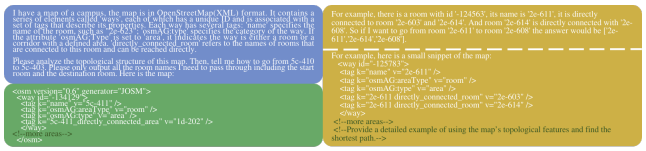

Fig. 3. Our prompts include two main elements: a task description in a blue box and the osmAG map in a green box. We offer three prompt levels: Level 1 with just the description and map, Level 2 with a simple example in the upper yellow box, and Level 3 with a detailed example in the lower yellow box.

for tasks such as code generation[11], object rearrangement[12], and motion planning[13]. Visual-language navigation systems like [14] allow robots to follow human instructions using advanced map technologies such as VLMaps[15] that combine language models with 3D point clouds. Our focus is on developing map representations that are easily understandable by both LLMs and robots, prioritizing map data to enhance navigation and decision-making in robotics.

B. Scene Graph in Robotics

Armeni et al. [16] and Hughes et al. [17] introduced Scene Graphs as RGB-D camera-based 3D representations that organize environments into layered graphs with nodes for spatial concepts from geometry to high-level semantics, and edges depicting relationships. Subsequent applications by Chen et al. [18], and others have advanced Scene Graphs for localization and planning. Unlike Scene Graphs, osmAG, derived from grid maps, LiDAR, or CAD data, is less prone to occlusion and remains reliable without frequent updates. It focuses on permanent structures and doesn't require the semantic or visual data that traditional robot navigation algorithms use to avoid obstacles.

III. Approach

This paper explores how osmAG can aid LLMs in robotic tasks like path planning. For proprietary LLMs such as ChatGPT, in Sections III-A and III-B, we explore the levels of task description in prompt and the variants of osmAG, respectively. For open-source models like LLaMA2, initial results show a low success rate of approximately 0.1 (see Table II). This indicates that merely combining prompt engineering with appropriate osmAG representation falls short of our objectives, leading us to also incorporate fine-tuning LLMs using our datasets. Detailed in Section III-C is our methodology for dataset creation, and Section III-D outlines the fine-tuning process using LoRA (Low Rank Adaptation)[19].

Fig. 4. The osmAG map representation and two of its variants. The osmAG enclosed in blue box represents the original format that uses 'passage' to describe connections between areas. In the yellow box we illustrate Variant 1 of osmAG, which introduces a tag with a key set to "connected_area" and a value corresponding to the area connected via the passage. Variant 2 of osmAG displayed in the green box, modifies Variant 1 by replacing "connected_area" with "current area name_directly_connected_room".

A. Prompt Engineering

The prompt is structured into two parts: the first provides a complete task description, including the map format and task details, while the second part offers osmAG context. We explored three levels of detail in the task description (see Fig. 3): all include basic osmAG and task explanations. Level 1 has no example, Level 2 includes a simple example, and Level 3 provides a detailed example with an illustrative map and detailed answer. Our aim is to enhance LLM performance by using examples to better clarify the task, leveraging in-context learning principles [20].

B. osmAG Variants

As shown in Fig. 4 and referenced in [3], the original Area Graph uses 'passage' to connect different areas, with 'passage' and 'area' under separate 'way' tags in the osmAG XML, complicating connection understanding for LLMs. We developed a script to create osmAG variants that integrate connection details directly within the 'area' tags. Variant 1 adds a 'connected_area' key directly to the 'area', specifying its connected area. Variant 2 further specifies the current area's name and its direct connections.

C. Datasets

In order to test and fine-tune the ability of LLMs to understand the topology and hierarchy of osmAG, we need to construct specific datasets with language instructions, osmAG, and ground truth. We generate topological and hierarchical datasets specifically to enhance and evaluate the LLMs' capabilities in these areas. Additionally, a general knowledge dataset is created to evaluate whether the model retains its general knowledge capabilities after fine-tuning.

Due to the limitation of token size (LLaMA2 supports up to 4096 tokens) and our decision to omit the metric information of osmAG, the specific shapes of areas become irrelevant. Instead, only the information regarding connections and hierarchy remains pertinent. Consequently, employing hand-drawn layout templates is sufficient for evaluating the LLMs' proficiency in understanding topological and hierarchical relationships.

1) Topological Datasets: Map templates shown in Fig. 5 are handcrafted using JOSM, with 'area' and 'passage' defined per osmAG standards in [6]. Labels like '1d-201' indicate a room's location by zone, floor, wing, and room number. We employ a script to randomize these attributes to create varied maps from a single template. Room numbers are shuffled to ensure paths rely on map information, not sequence. For each map, we ask the LLM to find a path (sequence of room names) between each two rooms as one item of the dataset. Dataset 1, using the 'normal' layout Template (a) from Fig. 5, contains 440 entries. Dataset 2, mixing Templates (a), (b), and (c), includes 12,520 entries aimed for training, with 440 reserved for testing. Dataset 3, derived from an Template (d) that is not used in the training dataset, aim to evaluate the model's generalization ability. The number of rooms in templates is limited by token limits, preventing larger designs. However, leaf rooms with a single door are unnecessary for path planning unless they are the terminal rooms, so they can be omitted beforehand. Thus, despite the template's limited number of rooms, it remains highly relevant for real-world, large environment applications.

Based on experiment in Sections IV-A and IV-B, we utilize task description Level 3 and osmAG Variant 2 in prompts. Our datasets' ground truth is sequential room numbers from standard path planning algorithms. For circular paths with equal-length alternatives, both are included.

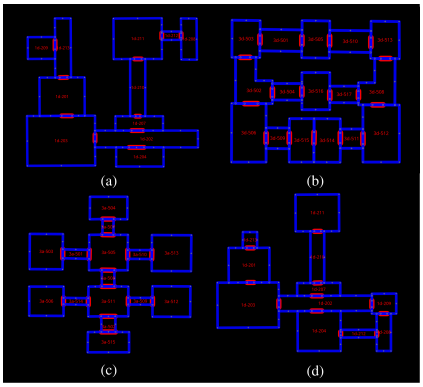

Fig. 5. Hand-drawn map layout templates created using JOSM, with areas depicted as blue polygons, passages represented by red lines, and room names in red. Templates (a), (b), and (c) were utilized to generate datasets for fine-tuning the LLaMA2 model. Template (d), on the other hand, was exclusively used in the test dataset to assess the LLM's capacity to adapt to unseen layouts.

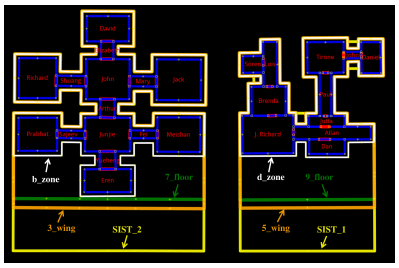

Fig. 6. This image displays two osmAG maps from Dataset 2, each randomly assigned a 'owner' tag per room (first names shown for clarity). The maps are structured by zone, floor, and wing, with buildings 'SIST_1' or 'SIST_2' to establish a hierarchical tree using 'parent' tags. Details for the left map are: SIST_2, 3_wing, 7_floor, b_zone.

2) Hierarchical Datasets: As shown in Fig. 6, we randomly selected two osmAG from Dataset 2, assigning each room a random 'owner' tag. We structured each map with 'parent' tags linking rooms to zones, floors, and wings, and assigned buildings 'SIST_1' or 'SIST_2' to establish a hierarchy. This setup supports queries asking the LLM to locate individuals by directing towards the correct building. We created 1056 such queries for training Dataset 4. An example of this map visualized by JOSM is in Fig. 6. The LLM must navigate hierarchy tree to correctly identify the target building.

3) General Knowledge Dataset: LoRA fine-tuning is not completely immune to catastrophic forgetting [21]. To verify that general knowledge capabilities remain intact, we created a small Dataset 5 with 20 general questions, like "Who wrote 'Hamlet'?", to test the model after fine-tuning.

D. Fine-tuning

We utilize Meta's LLaMA2 model[2], a decoder-only transformer, as our base model. The models are fine-tuned using datasets from Section III-C with the LoRA (Low Rank Adaptation)[19] method, which updates additive low-rank matrices in neural layers while keeping the original weights frozen, reducing the number of trainable parameters and allowing efficient use of computational resources. Fine-tuning targets topological tasks with Dataset 2 and hierarchical tasks with Dataset 4. The LoRA hyperparameters are as follows: the rank is set to 8, and target modules of LoRA are set to "q_proj" and "v_proj". These two modules are the query and value matrices in the self-attention mechanism of the transformer architecture [19], [25]. The learning rate is set to 5e-5 and we opted for a cosine learning rate scheduler. The fine-tuning process is illustrated in Figure 7. The LLaMA2-7B and LLaMA2-13B models were fine-tuned for topological and hierarchical tasks using 4×NVIDIA A40 and 4×A100 GPUs, respectively. Topological tasks took 6.6 hours for LLaMA2-7B and 5.5 hours for LLaMA2-13B, both running 2.5 epochs with a dataset of 12,520 entries. Hierarchical tasks required 2 hours for LLaMA2-7B and 1.5 hours for LLaMA2-13B, each undergoing 7.76 epochs with 1,056 entries.

TABLE I
COMPARISON CHATGPT-3.5&4'S SUCCESS RATE ON DIFFERENT PROMPT LEVELS AND OSMAG VARIANTS

Task Description Level	Original osmAG		osmAG Variant 1		osmAG Variant 2	
	Chat GPT3.5	Chat GPT4	Chat GPT3.5	Chat GPT4	Chat GPT3.5	Chat GPT4
Level 1	**0.54**	0.85	0.50	0.95	0.69	0.95
Level 2	0.42	**0.87**	0.45	**0.97**	**0.70**	0.95
Level 3	0.49	**0.87**	**0.57**	0.96	0.69	**0.96**

IV. EXPERIMENTS

Since LLMs can only process textual input, comparing osmAG with occupancy grid maps or point clouds is impractical. Regarding the 3D Scene Graph, it primarily focuses on semantic representation within a 3D scene. We assert that semantic information is unnecessary for mobile robot path planning between specified start and end rooms, and therefore, do not include comparisons with Scene Graph.

For topological tests, matching ground truth is considered successful, while correctly identifying buildings signifies success in hierarchical tests. Model performances are measured by success rates on the test dataset. Sections IV-A and IV-B detail experiments on ChatGPT-3.5 and ChatGPT-4 regarding prompt engineering and osmAG variants. Sections IV-C and IV-D compare pre and post-fine-tuning performance of LLaMA2 models. Section IV-E and Fig. 2 demonstrate ChatGPT-4 using real osmAG for live path planning and adjustments.

However, it is important to note that, despite ChatGPT-4's high success rate, all its responses are verbose. This verbosity persists even when specifying concise outputs in the prompt, like 'only output the room numbers'. While this chattiness may be acceptable in human interactions, it poses challenges for traditional robotic applications to utilize the LLMs' responses. Nevertheless, we counted those answers as correct, if the room numbers matched the ground truth.

A. Prompt Engineering Experiment

As discussed in Section III-A, we tested ChatGPT-3.5 (gpt-3.5-turbo-0125) and ChatGPT-4 (gpt-4-0125-preview)'s understanding of osmAG path planning

using three prompt levels. The results, shown in Table I, reveal that including an example in the prompt does not always enhance performance. However, we opted for Level 3 prompts in our training dataset for topological tasks due to its marginally better results.

B. osmAG Variant Experiment

The osmAG variants for this experiment are detailed in Section III-B. We compared original osmAG and two of its variants with different prompt levels on ChatGPT-3.5 and ChatGPT-4. According to the results summarized in Table I, osmAG Variant 2 outperforms the others on both models. Therefore, we have selected this variant as the preferred map representation for LLMs and used it in our fine-tuning dataset.

C. Topological & Hierarchical Understanding Experiment Without Fine-tuning

We tested the LLaMA2-7B and LLaMA2-13B models on Datasets 1-5 to evaluate their map understanding. For comparison, we also used ChatGPT-3.5 and ChatGPT-4's APIs on these datasets. The results, shown in Table II, reveal a 0.1 success rate for the LLaMA2 models in topological tasks without fine-tuning, which is impractical. ChatGPT-3.5 achieved about a 0.5 success rate, still insufficient for real-world use. In hierarchical tasks, LLaMA2-13B achieved a 0.55 success rate, better than LLaMA2-7B's 0.19, but not yet practical. ChatGPT-3.5 reached a 0.66 success rate, still below practical deployment standards. ChatGPT-4 showed high success rates in both tasks.

D. Fine-tuning Experiment

After fine-tuning the LLaMA2 models as outlined in Section III-D, we assessed their performance on Datasets 1-5, observing significant improvements detailed in Table II.

For topological tasks, the fine-tuned LLaMA2-7B and LLaMA2-13B models exceeded ChatGPT-3.5's performance, with LLaMA2-13B also outperforming ChatGPT-4 on Dataset 2. LLaMA2-7B achieved over a 0.9 success rate on Datasets 1-2 with templates used in fine-tuning but dropped to 0.89 on new layouts. LLaMA2-13B showed stronger generalization with a 0.97 success rate on new layouts.

During testing, besides the fixed prompt from training, we also use random prompts to evaluate generalization. The results, shown in Table II, reveal a performance drop for LLaMA2-7B under varied prompts, while LLaMA2-13B maintained high performance, suggesting its suitability for unpredictable

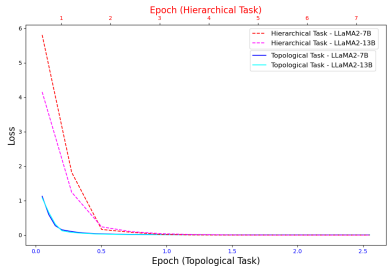

Fig. 7. Fine-tuning LLaMA2-7B and LLaMA2-13B on Topological and Hierarchical Tasks

interactions. Conversely, LLaMA2-7B is more suited for scenarios with consistent prompts, particularly when computational resources are limited.

In hierarchical tasks, after fine-tuning, both the LLaMA2-7B and LLaMA2-13B models achieve a success rate of 1, both models also generalize well on unseen prompts, which makes them totally practical.

E. Real-life Experiment

Here we perform an experiment to emulate the real-world situation of a robot blocked by a construction site, as show in Fig. 1. As depicted in Fig. 2, we send a request to the ChatGPT-4 API, providing a osmAG map with Prompt Level 3 to query a path planning from the start room to the destination room. The osmAG is converted into Variant 2, and any leaf areas with single door are removed to conserve tokens. ChatGPT-4 then returns a path, highlighted with red lines in the image, via a two-door elevator (identified by a semantic tag indicating it is an elevator). Upon introducing an email regarding elevator maintenance, ChatGPT-4 adjusts the path, adding a detour to bypass the unavailable elevator.

V. Conclusion and Discussion

In the rapidly evolving field of AI, LLMs are increasingly used to enhance robotic intelligence, though integrating them remains a key research area. This paper introduces osmAG, a map representation suited for LLM-robot systems, interpretable by LLMs, compatible with robotic algorithms, and understandable by humans. For proprietary models like ChatGPT, we provide datasets to evaluate the model's comprehension of osmAG, along with osmAG variants to improve performance. For open-source models such as LLaMA2, we supply datasets, dataset generation methods, and fine-tuned adapters for comprehensive testing. We recognize that in real robotics, path length is crucial, but current token limitations and the LLM's difficulty with math mean we cannot ensure optimal paths. However, we are

TABLE II

COMPARISON OF SUCCESS RATES OF LLMS ON TOPOLOGICAL (T), HIERARCHICAL (H) AND GENERAL (G) TASKS

	LLaMA2-7B	LLaMA2-13B	Fine-tuned LLaMA2-7B (with unseen prompt)	Fine-tuned LLaMA2-13B (with unseen prompt)	ChatGPT-3.5	ChatGPT-4.0
Dataset 1 (T)	0.10	0.12	0.99 (0.78)	0.98 (0.91)	0.54	0.99
Dataset 2 (T)	0.05	0.066	0.94 (0.60)	0.95 (0.94)	0.50	0.89
Dataset 3 (T)	0.11	0.14	0.89 (0.75)	0.97 (0.92)	0.53	0.96
Dataset 4 (H)	0.19	0.55	1.0 (0.98)	0.99 (0.98)	0.66	0.99
Dataset 5 (G)	0.95	0.95	0.95	0.95	1	1

exploring closer integration of LLMs with traditional algorithms like A* [6] to address these issues.

Traditional robotics has been explored for decades, but integrating it with LLMs is a new frontier. osmAG aims to accelerate this integration, facilitating a map representation that aligns with LLMs, robotic systems, and human operators.

REFERENCES

[1] OpenAI, "OpenAI: Introducing ChatGPT," https://openai.com/blog/chatgpt, 2022.

[2] 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.

[3] J. Hou, Y. Yuan, and S. Schwertfeger, "Area graph: Generation of topological maps using the voronoi diagram," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 509–515.

[4] J. Hou, Y. Yuan, Z. He, and S. Schwertfeger, "Matching maps based on the area graph," *Intelligent Service Robotics*, 2022.

[5] Z. He, H. Sun, J. Hou, Y. Ha, and S. Schwertfeger, "Hierarchical topometric representation of 3d robotic maps," *Autonomous Robots*, vol. 45, no. 5, pp. 755–771, 2021.

[6] 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)*. IEEE, 2023, https://arxiv.org/pdf/2309.04791.pdf.

[7] 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, 2023.

[8] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo *et al.*, "Solving quantitative reasoning problems with language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3843–3857, 2022.

[9] R. T. McCoy, S. Yao, D. Friedman, M. Hardy, and T. L. Griffiths, "Embers of autoregression: Understanding large language models through the problem they are trained to solve," *arXiv preprint arXiv:2309.13638*, 2023.

[10] 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.

[11] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *Microsoft Auton. Syst. Robot. Res*, vol. 2, p. 20, 2023.

[12] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *arXiv preprint arXiv:2303.06247*, 2023.

[13] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting robot plans with natural language feedback," *arXiv preprint arXiv:2204.05186*, 2022.

[14] X. Wang, W. Wang, J. Shao, and Y. Yang, "Lana: A language-capable navigator for instruction following and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 048–19 058.

[15] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 608–10 615.

[16] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5664–5673.

[17] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022.

[18] J. Chen, D. Barath, I. Armeni, M. Pollefeys, and H. Blum, "" where am i?" scene retrieval with language," *arXiv preprint arXiv:2404.14565*, 2024.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[20] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[21] Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma, "Investigating the catastrophic forgetting in multimodal large language models," *arXiv preprint arXiv:2309.10313*, 2023.

[22] hiyouga, "Llama factory," https://github.com/hiyouga/LLaMA-Factory, 2023.

[23] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3505–3506.

[24] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "Zero: Memory optimizations toward training trillion parameter models," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020, pp. 1–16.

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