Leveraging Commonsense Knowledge from Large Language Models for Task and Motion Planning

Yan Ding*
SUNY Binghamton
Binghamton, NY
yding25@binghamton.edu

Xiaohan Zhang*
SUNY Binghamton
Binghamton, NY
xzhan244@binghamton.edu

Chris Paxton
Meta AI
Pittsburgh, PA
cpaxton@meta.com

Shiqi Zhang
SUNY Binghamton
Binghamton, NY
zhangs@binghamton.edu

***Abstract*—Multi-object rearrangement is a crucial skill for service robots, and commonsense reasoning is frequently needed in this process. However, achieving commonsense arrangements requires knowledge about objects, which is hard to transfer to robots. Large language models (LLMs) are one potential source of this knowledge, but they do not naively capture information about plausible physical arrangements of the world. We propose LLM-GROP, which uses prompting to extract commonsense knowledge about semantically valid object configurations from an LLM and instantiates them with a task and motion planner in order to generalize to varying scene geometry. LLM-GROP allows us to go from natural-language commands to human-aligned object rearrangement in varied environments. Based on human evaluations, our approach achieves the highest rating while outperforming competitive baselines in terms of success rate while maintaining comparable cumulative action costs. Finally, we demonstrate a practical implementation of LLM-GROP on a mobile manipulator in real-world scenarios. Supplementary materials are available at: https://sites.google.com/view/llm-grop**

I. Introduction

Multi-object rearrangement is a critical skill for service robots to complete everyday tasks, such as setting tables, organizing bookshelves, and loading dishwashers [34, 37]. These tasks demand robots exhibit both manipulation and navigation capabilities. For example, a robot tasked with setting a dinner table might need to retrieve tableware objects like a fork or a knife from different locations and place them onto a table surrounded by chairs, as shown in Fig. 1. To complete the task, the robot needs to correctly position the tableware objects in semantically meaningful configurations (e.g., a fork is typically on the left of a knife) and efficiently navigate indoors while avoiding obstacles like chairs or humans whose locations are unknown in advance.

A variety of mobile manipulation systems have been developed for object rearrangement tasks [13, 28, 36, 15, 14, 22, 6, 35]. Most of those systems require explicit instructions, such as arranging similar colored items in a line or placing them in a specific shape on a table [13, 15, 14, 6, 35, 23]. However, user requests in the real world tend to be underspecified: there can be many different ways to set a table that are not equally preferred. How does a robot figure out a fork should be placed on the left of a plate and a knife on the right? Considerable commonsense knowledge is needed.

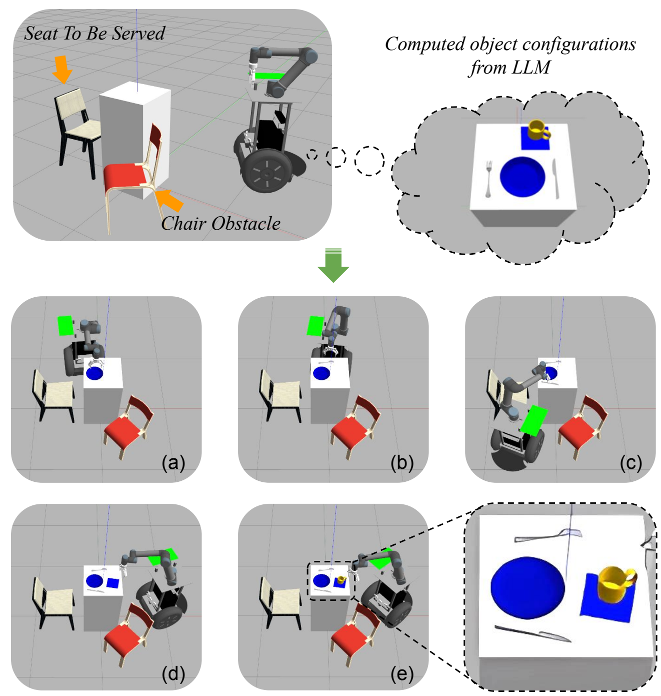

Fig. 1: A mobile manipulator is assigned the task of setting a table in a dining domain. The manipulator needs to arrange several tableware objects, including a knife, a fork, a plate, a cup mat, and a mug. These objects are available on the other tables, and there are also randomly generated obstacles (i.e., the red chair) that are not included in the pre-built map beforehand. The robot needs to compute feasible and efficient plans for rearranging the objects on the target table using both navigation and manipulation behaviors.

Recent results have shown large language models (LLMs) like GPT3 [4] and ChatGPT [30] capture a great deal of this common sense knowledge [25]. In the past, researchers have equipped mobile manipulators with semantic information using machine learning methods [28, 27, 36]. Those methods require collecting training data, which limits their applicability to robots working on complex service tasks.

To equip robot planning methods with common sense for object rearrangement, we introduce LLM-GROP, standing for **L**arge **L**anguage **M**odel for **G**rounded **RO**bot Task and Motion **P**lanning, our approach that leverages commonsense knowledge for planning to complete object rearrangement tasks. LLM-GROP first uses an LLM to generate *symbolic*

* indicates equal contribution. The complete version of this paper is under review [10].

spatial relationships between objects, e.g., a fork and a knife are placed on the left and right respectively. The spatial relationships then can be grounded to different *geometric* spatial relationships whose feasibility levels are evaluated by a motion planning system, e.g., placing objects in some areas of a table can be easier than the others. Finally, the feasibility and efficiency of different task-motion plans are optimized towards maximizing long-term utility, i.e., seeking the best trade-off between motion feasibility and task-completion efficiency.

We have applied LLM-GROP to a dining room, where a mobile manipulator must set a table according to a user's instructions. A set of tableware objects are provided to the robot, where the robot's task is to compute a tabletop configuration of those objects that comply with common sense, and compute a task-motion plan to realize the configuration. To evaluate the performance of our approach, we had users rate different place settings to get a subjective evaluation. We observed improvements in user satisfaction from LLM-GROP compared with existing object rearrangement methods, while maintaining similar or lower cumulative action costs. Finally, LLM-GROP was demonstrated on a real robot.

II. Related Work

We first introduce the object rearrangement domain, then discuss methods for tabletop object arrangement that mostly rely on supervised learning methods, and finally summarize research on using large language models for planning.

A. Object Rearrangement

Rearranging objects is a critical task for service robots, and much research has focused on moving objects from one location to another and placing them in a new position. Examples include the Habitat Rearrangement Challenge [34] and the AI2-THOR Rearrangement Challenge [37]. There is rich literature on object rearrangement in robotics [13, 15, 14, 6, 35, 23, 39]. A common assumption in those methods is that a goal arrangement is part of the input, and the robot knows the exact desired positions of objects. ALFRED [32] proposed a language-based multi-step object rearrangement task, for which a number of solutions have been proposed that combine high-level skills [2, 29], and which have recently been extended to use LLMs as input [18]. However, these operate at a very coarse, discrete level, instead of making motion-level and placement decisions, and thus can't make granular decisions about common-sense object arrangements.

By contrast, our work accepts underspecified instructions from humans, such as setting a dinner table with a few provided tableware objects. LLM-GROP has the capability to do common sense object rearrangement by extracting knowledge from LLMs, and operates both on a high level and on making motion-level placement decisions.

B. Predicting Complex Object Arrangements

Object arrangement is a task that involves arranging items on a tabletop to achieve a specific functional, semantically valid goal configuration. This task requires not only the calculation of object positions but also adherence to common sense, such as placing forks to the left and knives to the right when setting a table. Previous studies in this area, such as [28, 21, 27, 36], focused on predicting complex object arrangements based on vague instructions. For instance, StructFormer [26] is a transformer-based neural network for arranging objects into semantically meaningful structures based on natural-language instructions. By comparison, our approach LLM-GROP utilizes an LLM for commonsense acquisition to avoid the need of demonstration data for computing object positions. Additionally, we optimize the feasibility and efficiency of plans for placing tableware objects.

There is recent research for predicting complex object arrangement using web-scale diffusion models [21]. Their approach, called DALL-E-Bot, enables a robot to generate images based on a text description using DALL-E [31], and accordingly arrange objects in a tabletop scenario. Similar to DALL-E-Bot, LLM-GROP achieves zero-shot performance using pre-trained models, but it is not restricted to a single top-down view of a table. In addition, we consider the uncertainty in manipulation and navigation, and optimize efficiency and feasibility in planning.

C. Robot Planning with Large Language Models

Many LLMs have been developed in recent years, such as BERT [8], GPT-3 [4], ChatGPT [30], CodeX [5], and OPT [38]. These LLMs can encode a large amount of common sense [25] and have been applied to robot task planning [20, 16, 1, 17, 33, 9, 24]. For instance, the work of Huang et. al. showed that LLMs can be used for task planning in household domains by iteratively augmenting prompts [16]. SayCan is another approach that enabled robot planning with affordance functions to account for action feasibility, where the service requests are specified in natural language (e.g., "make breakfast") [1]. Compared with those methods, LLM-GROP optimizes both feasibility and efficiency while computing semantically valid geometric configurations.

III. The LLM-GROP Approach

The objective of this task is to rearrange multiple tableware objects, which are initially scattered at different locations, into a tabletop configuration that is semantically valid and aligns with common sense. The robot is provided with prior knowledge about table shapes and locations, and equipped with skills of loading and unloading tableware objects. There are dynamic obstacles, e.g., chairs around tables, that can only be sensed at planning time. We consider uncertainty in navigation and manipulation behaviors. For instance, the robot can fail in navigation (at planning or execution time) when its goal is too close to tables or chairs, and it can fail in manipulation when it is not close enough to the target position.

In this paper, we develop LLM-GROP that leverages LLMs to facilitate a mobile manipulator completing object rearrangement tasks. LLM-GROP consists of two key components, LLM for generating symbolic spatial relationships (Sec. III-A)

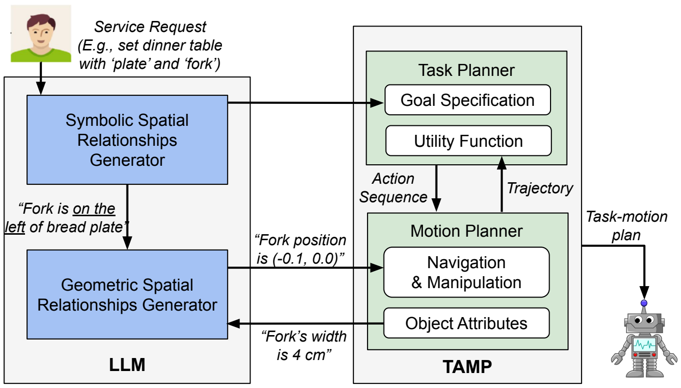

Fig. 2: LLM-GROP takes service requests from humans for setting tables and produces a task-motion plan that the robot can execute. LLM-GROP is comprised of two key components: the LLM and the Task and Motion Planner. The LLM is responsible for creating both symbolic and geometric spatial relationships between the tableware objects. This provides the necessary context for the robot to understand how the objects should be arranged on the table. The Task and Motion Planner generates the optimal plan for the robot to execute based on the information provided by the LLM.

and geometric spatial relationships (Sec. III-B) between objects, and TAMP for computing optimal task-motion plan (Sec. III-C), as shown in Fig. 2.

A. Generating Symbolic Spatial Relationships

LLMs are used to extract common sense knowledge regarding symbolic spatial relationships among objects placed on a table. This is accomplished through the utilization of a template-based prompt:

> Template 1: *The goal is to set a dining table with objects. The symbolic spatial relationship between objects includes [spatial relationships]. [examples]. What is a typical way of positioning [objects] on a table? [notes].*

where *[spatial relationships]* includes a few spatial relationships, such as *to the left of* and *on top of*. In presence of *[examples]*, the prompting becomes few-shot; when no examples are provided, it is simplified to zero-shot prompting. In practice, few-shot prompts can ensure that the LLM's response follows a predefined format, though more prompt engineering efforts are needed. *[objects]* refers to the objects to be placed on the table, such as *a plate*, *a fork*, and *knife*. To control the LLM's output, *[notes]* can be added, such as the example *"Each action should be on a separate line starting with 'Place'. The answer cannot include other objects"*. Please refer to Sec. VI in Appendix for more technical details.

B. Generating Geometric Spatial Relationships

After determining the symbolic spatial relationships between objects in Sec. III-A, we move on to generate their geometric configurations, where we use the following LLM template.

> Template 2: *[object A] is placed [spatial relationship] [object B]. How many centimeters [spatial relationship] [object B] should [object A] be placed?*

For instance, when we use Template 2 to generate prompt "*A dinner plate is placed to the left of a knife. How many centimeters to the left of the water cup should the bread plate be placed?*", GPT-3 produces the output "*Generally, the dinner knife should be placed about 5-7 centimeters to the right of the dinner plate.*"

To determine the positions of objects, we first choose a coordinate origin. This origin could be an object that has a clear spatial relationship to the tabletop and is located centrally. A dinner plate is a good example of such an object. We then use the recommended distances and the spatial relationships between the objects to determine the coordinates of the other objects. Specifically, we can calculate the coordinates of an object by adding or subtracting the recommended distances in the horizontal and vertical directions, respectively, from the coordinates of the coordinate origin. The LLM-guided position for the ith object is denoted as (x^i, y^i), where $i \in N$. Please refer to Sec. VII in Appendix for more technical details.

C. Computing Task-Motion Plans

After identifying feasible object configurations on the tabletop in Steps 1 and 2, the next step is to place the objects on the tabletop based on one of object configuration sequences. At the task level, the robot must decide the sequence of object placement and how to approach the table. For example, if a bread is on top of a plate, the robot must first place the plate and then the bread. The robot must also determine how to approach the table, such as from which side of the table. Once the task plan is determined, the robot must compute 2D navigation goals (denoted as loc) at the motion level that connect the task and motion levels. Subsequently, the robot plans motion trajectories for navigation and manipulation behaviors.

In the presence of dynamic obstacles, not all navigation goals (loc) are equally preferred. For instance, it might be preferable for the robot to position itself close to an object for placement rather than standing at a distance and extending its reach. A recent approach called GROP [39] was developed for computing the optimal navigation goal loc, which enabled the task-motion plan with the maximal utility for placing each object in terms of feasibility and efficiency given an object configuration (x^i_j, y^i_j), where $0 \leq j \leq M$. Therefore, for different groups of object configurations, we use GROP to compute the maximal utility value of task-motion plans and select the best one for execution. Fig. 3 shows one task-motion plan generated using LLM-GROP for a four-object rearrangement task.

IV. Experiments

In this section, we evaluate the performance of LLM-GROP using the task of rearranging tableware objects. The robot needs to compute semantically valid tabletop arrangements, plan to efficiently rearrange the objects, and realize the plan via navigation and manipulation behaviors. Please see Sec. VIII in Appendix for the experiment setup.

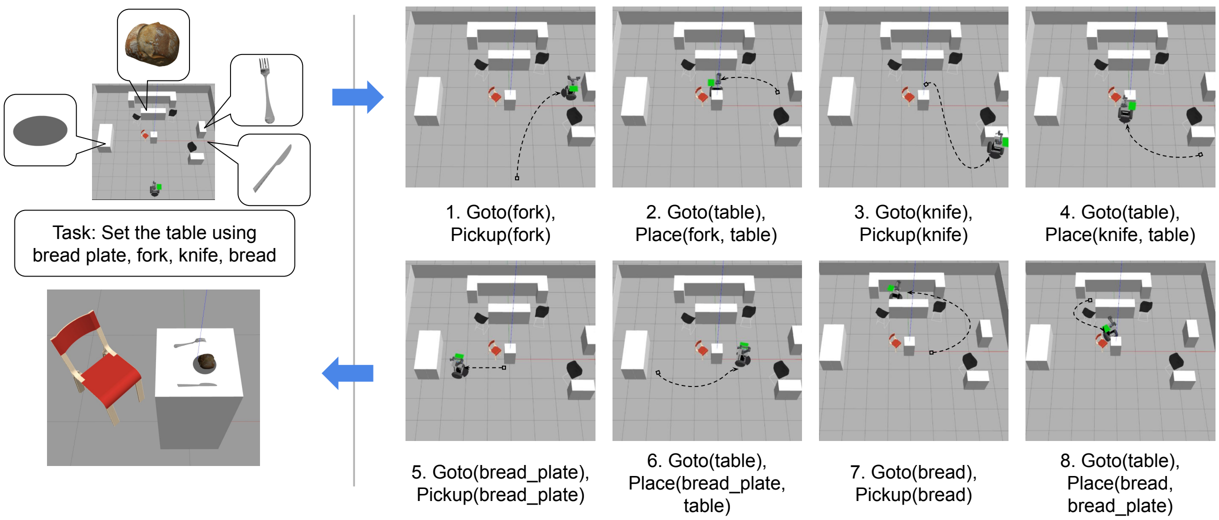

Fig. 3: An illustrative example of LLM-GROP showing the robot navigation trajectories (dashed lines) as applied to the task of "set the table with a bread plate, a fork, a knife, and a bread." LLM-GROP is able to adapt to complex environments, using commonsense extracted from GPT-3 to generate efficient (i.e., minimize the overall navigation cost) and feasible (i.e., select an available side of the table to unload) pick-and-place motion plans for the robot.

Baselines: LLM-GROP is evaluated by comparing its performance to three baselines, where the first baseline is the weakest.

- Task Planning with Random Arrangement (TPRA): This baseline uses a task planner to sequence navigation and manipulation behaviors, while it randomly selects standing positions next to the target table and randomly places objects in no-collision positions on the table.
- LLM-based Arrangement and Task Planning (LATP): It can predict object arrangements using LLMs and perform task planning. It uniformly samples standing positions around the table for manipulating objects.
- GROP [39]: It considers plan efficiency and feasibility for task-motion planning, and lacks the capability of computing semantically valid arrangements.

LLM-GROP vs. Baselines: Fig. 4 shows the key findings of our experiments, which compares the performance of LLM-GROP to the three other baseline approaches. The x-axis indicates the time each method takes to complete a single task, while the y-axis indicates the corresponding user rating (see Sec. IX in Appendix for the rating guidelines). The results demonstrate that our LLM-GROP achieves the highest user rating and the shortest execution time compared to the other approaches. While GROP proves to be as efficient as our approach, it receives a significantly lower rating score. By contrast, TPRA and LATP both receive lower user ratings than our LLM-GROP. They also display poor efficiency. This is because they lack the navigation capabilities to efficiently navigate through complex environments. For instance, when their navigation goals are located within an obstacle area, they struggle to adjust their trajectory, leading to longer task completion times. More quantitative results are presented in Sec. X in Appendix.

In addition, we fully implemented and tested LLM-GROP

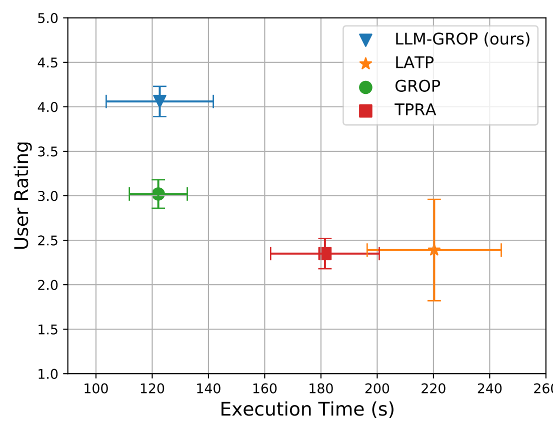

Fig. 4: Overall performance of LLM-GROP as compared to three baselines based on mean values and standard errors of user ratings and robot execution time for all tableware object arrangement tasks.

on real robot hardware (Sec. XI in Appendix).

V. Conclusion and Future Work

To summarize, we propose LLM-GROP, which demonstrates how we can extract semantic information from LLMs and use it as a way to make commonsense, semantically valid decisions about object placements as a part of a task and motion planner - letting us execute multi-step tasks in complex environments in response to natural-language commands. In the future, we may take more information from methods like M0M [7], in order to perform grasping and manipulation of fully unknown objects in unknown scenes, and expand to a wider set of placement problems.

REFERENCES

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can and not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR, 2022.

[3] Valérie Boor, Mark H Overmars, and A Frank Van Der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 2, pages 1018–1023. IEEE, 1999.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[6] Sang Hun Cheong, Brian Y Cho, Jinhwi Lee, ChangHwan Kim, and Changjoo Nam. Where to relocate?: Object rearrangement inside cluttered and confined environments for robotic manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7791–7797. IEEE, 2020.

[7] Aidan Curtis, Xiaolin Fang, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Caelan Reed Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1940–1946. IEEE, 2022.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Yan Ding, Xiaohan Zhang, Saeid Amiri, Nieqing Cao, Hao Yang, Chad Esselink, and Shiqi Zhang. Robot task planning and situation handling in open worlds. *arXiv preprint arXiv:2210.01287*, 2022.

[10] Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. Task and motion planning with large language models for object rearrangement. *arXiv preprint arXiv:2303.06247*, 2023.

[11] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Sven Thiele. A user's guide to gringo, clasp, clingo, and iclingo. 2008.

[12] Walter R Gilks and Pascal Wild. Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2): 337–348, 1992.

[13] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. Semantically grounded object matching for robust robotic scene rearrangement. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11138–11144. IEEE, 2022.

[14] Jiayuan Gu, Devendra Singh Chaplot, Hao Su, and Jitendra Malik. Multi-skill mobile manipulation for object rearrangement. *arXiv preprint arXiv:2209.02778*, 2022.

[15] Eric Huang, Zhenzhong Jia, and Matthew T Mason. Large-scale multi-object rearrangement. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 211–218. IEEE, 2019.

[16] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *Thirty-ninth International Conference on Machine Learning*, 2022.

[17] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.

[18] Yuki Inoue and Hiroki Ohashi. Prompter: Utilizing large language model prompting for a data efficient embodied instruction following. *arXiv preprint arXiv:2211.03267*, 2022.

[19] Yu-qian Jiang, Shi-qi Zhang, Piyush Khandelwal, and Peter Stone. Task planning in robotics: an empirical comparison of pddl-and asp-based systems. *Frontiers of Information Technology & Electronic Engineering*, 20(3): 363–373, 2019.

[20] Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot, and Harsh Agrawal. Housekeep: Tidying virtual households using commonsense reasoning. *arXiv preprint arXiv:2205.10712*, 2022.

[21] Ivan Kapelyukh, Vitalis Vosylius, and Edward Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *arXiv preprint arXiv:2210.02438*, 2022.

[22] Jennifer E King, Marco Cognetti, and Siddhartha S Srinivasa. Rearrangement planning using object-centric and robot-centric action spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3940–3947. IEEE, 2016.

[23] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[24] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi

Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[26] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. *arXiv preprint arXiv:2110.10189*, 2021.

[27] Weiyu Liu, Tucker Hermans, Sonia Chernova, and Chris Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv preprint arXiv:2211.04604*, 2022.

[28] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6322–6329. IEEE, 2022.

[29] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.

[30] OpenAI. Chatgpt. Accessed: 2023-02-08, 2023. URL https://openai.com/blog/chatgpt/. cit. on pp. 1, 16.

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[32] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[33] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

[34] Andrew Szot, Karmesh Yadav, Alex Clegg, Vincent-Pierre Berges, Aaron Gokaslan, Angel Chang, Manolis Savva, Zsolt Kira, and Dhruv Batra. Habitat rearrangement challenge 2022. https://aihabitat.org/challenge/rearrange_2022, 2022.

[35] Vasileios Vasilopoulos, Yiannis Kantaros, George J Pappas, and Daniel E Koditschek. Reactive planning for mobile manipulation tasks in unexplored semantic environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6385–6392. IEEE, 2021.

[36] Qiuhong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. Lego-net: Learning regular rearrangements of objects in rooms. *arXiv preprint arXiv:2301.09629*, 2023.

[37] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[38] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[39] Xiaohan Zhang, Yifeng Zhu, Yan Ding, Yuke Zhu, Peter Stone, and Shiqi Zhang. Visually grounded task and motion planning for mobile manipulation. *arXiv preprint arXiv:2202.10667*, 2022.

APPENDIX

VI. HANDLING LOGICAL ERRORS FOR SYMBOLIC SPATIAL RELATIONSHIPS GENERATION

LLMs are generally reliable in demonstrating common sense, but there may be times when they produce contradictory results. To prevent **logical errors**, a logical reasoning-based approach has been developed to evaluate the consistency of generated candidates with explicit symbolic constraints. This approach is implemented on answer set programming (ASP), which is a declarative programming language that expresses a problem as a set of logical rules and constraints [11]. ASP enables recursive reasoning, where rules and constraints can be defined in terms of other rules and constraints, providing a modular approach to problem-solving [19]. ASP is particularly useful for determining whether sets of rules and constraints are true or false in a given context.

The approach involves defining spatial relationships, their transitions, and rules for detecting conflicts. These rules are created by human experts and serve to ensure that the generated context is logical and feasible. One such rule is `:- below(X,Y),right(X,Y)`, which states that object `X` cannot be both "below" and "to the right of" object `Y` at the same time. This rule ensures that the resulting arrangement of objects is physically possible. An instance of identifying a logical error is provided. For example, an LLM may generate instructions for arranging objects as follows:

1) Place fruit bowl in the center of table.
2) *Place butter knife above and to the right of fruit bowl.*
3) *Place dinner fork to the left of butter knife.*
4) Place dinner knife to the right of butter knife.
5) *Place fruit bowl to the right of dinner fork.*
6) Place water cup below and to the left of dinner knife.

There are **logical inconsistencies** in the italic lines: Steps 2 and 3 suggest placing the *fruit bowl* below the *dinner fork*, while Step 5 suggests placing the *fruit bowl* to the right of the *dinner fork*. This contradicts the established rule and results in no feasible solutions.

VII. ADAPTIVE SAMPLING FOR GEOMETRIC SPATIAL RELATIONSHIPS GENERATION

Solely relying on the response of the LLMs is not practical as they do not account for object attributes such as shape and size, including tables constraints. To address this limitation, we have designed an adaptive sampling-based method that incorporates object attributes after obtaining the recommended object positions. Specifically, our approach involves sequencing the sampling of each object's position using a 2D Gaussian sampling technique [3], with (x^i, y^i) as the mean vector, and the covariance matrix describing the probability density function's shape.

The resulting distribution is an ellipse centered at (x^i, y^i) with the major and minor axes determined by the covariance matrix. However, we do not blindly accept all of the sampling results; instead, we apply multiple rules to determine their acceptability, inspired by rejection sampling [12]. These rules include verifying that the sampled geometric positions adhere to symbolic relationships at a high level, avoiding object overlap, and ensuring that objects remain within the table boundary. For example, if the bounding box of an object position falls outside the detected table bounds, we reject that sample. The bounding box of objects and the table are computed based on their respective properties, such as size or shape. After multiple rounds of sampling, we can obtain M object configuration sequences.

TABLE I: Objects that are involved in our object rearrangement tasks for evaluation, where tasks 1-5 include three objects, tasks 6 and 7 include four objects, and task 8 includes five objects.

Task #ID	Objects
1	Dinner Plate, Dinner Fork, Dinner Knife
2	Bread Plate, Water Cup, Bread
3	Mug, Bread Plate, Mug Mat
4	Fruit Bowl, Mug, Strawberry
5	Mug, Dinner plate, Mug Lid
6	Dinner Plate, Dinner Fork, Mug, Mug Lid
7	Dinner Plate, Dinner Fork, Dinner Knife, Strawberry
8	Dinner Plate, Dinner Fork, Dinner Knife, Mug, Mug Lid

TABLE II: Hypermeters of OpenAI's GPT-3 engines in Our Experiment

Parameter	**Value**	**Parameter**	**Value**
Model	text-davinci-003	Temperature	0.0
Top p	1.0	Maximum length	512
Frequency penalty	0.0	Presence penalty	0.0

VIII. EXPERIMENTAL SETUP

A mobile manipulator is assigned the task of setting a dinner table using a specific set of objects. In a simulated environment[1], the robot needs to retrieve multiple objects from various locations and place them on the central table. Additionally, an obstacle (i.e., a chair) will be randomly placed around the table. There are eight tasks that involve handling different objects, as detailed in TABLE I. We execute each task 20 times using the LLM-GROP system, and after each task was completed, we capture an image of the table, the chair, and the objects on the tabletop for later human evaluation. To carry out our experiments, we used OpenAI's GPT-3 engines. Please refer to TABLE II for the specific hyperparameters we adopted.

We have chosen not to use ChatGPT, a well-known language LLM, for large-scale experiments due to the unavailability of its APIs.

IX. RATING CRITERIA

We recruited five graduate students with engineering backgrounds, three females and two males between the ages of 22 and 30. We designed a five-point rating rule, which is outlined in Table III, and tasked the volunteers with scoring

[1]Implemented in the Gazebo simulator

TABLE III: Rating guidelines for human raters in the experiments. **1** point indicates the poorest tableware object arrangement as it suggests that some objects are missing. Conversely, **5** points represent the best arrangement.

Points	Rating Guidelines
1	Missing critical items compared with the objects listed at the top of the interface (e.g., dinner plate, dinner fork, dinner knife), making it hardly possible to complete a meal.
2	All items are present, but the arrangement is poor and major adjustments are needed to improve the quality to a satisfactory level.
3	All items are present and arranged fairly well, but still there is significant room to improve its quality.
4	All items are present and arranged neatly, though an experienced human waiter might want to make minor adjustments to improve.
5	All items are present and arranged very neatly, meeting the aesthetic standards of an experienced human waiter.

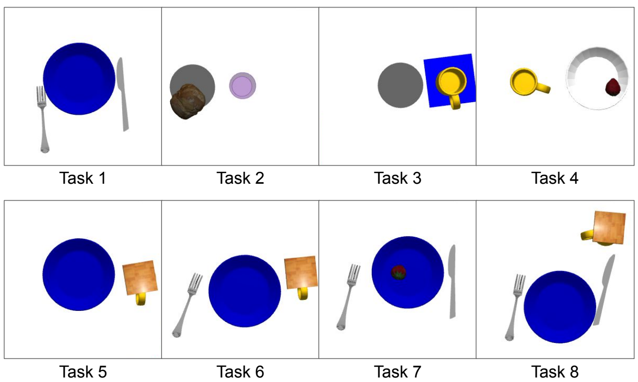

Fig. 5: Examples of tableware objects rearranged by our LLM-GROP agent in eight tasks, where the objects used in these tasks can be found in Table I. Our LLM-GROP enables the arrangement of tableware objects to be both semantically valid.

tableware object rearrangements in images they were shown. We generated 640 images from the four methods (three baselines and LMM-GROP) for eight tasks and each image required evaluation from all volunteers, resulting in a total sample size of 3200 images. The volunteers were shown one image at a time on a website[2] that we provided, and they scored each image from 1 to 5 based on the rating rules. We ensured that the rating was rigorous by using a website to collect rating results, thereby minimizing any potential biases that could arise from further interaction with the volunteers once they entered the website.

X. Individual Tasks

Fig. 5 provides several examples of various tasks that are rearranged by our agent. Fig. 6 presents the individual comparison results of each method for individual tasks. The x-axis corresponds to Task #ID in Table I, while the y-axis represents the average user rating for each method. Our LLM-GROP demonstrates superior performance over the baselines for each task. Specifically, tasks 1 to 5 receive slightly higher scores than tasks 6 and 8. This is reasonable because the latter two tasks require the robot to manipulate more objects, posing additional challenges for the robot.

XI. Real Robot Demonstration

We tested our LLM-GROP approach on a real mobile robot platform to demonstrate its effectiveness in rearranging a set of tableware objects, as shown in Fig. 7. The set included a dinner plate, a dinner fork, a dinner knife, a water cup, and a strawberry. The robot started on the left table and is tasked with rearranging the objects on the right table in the left image. After successfully completing the task, the robot successfully rearranged the objects as shown in the right image. The final object placements were semantically valid, such as the fork being on the left of the dinner plate and the strawberry being on the plate. These outcomes effectively demonstrate the effectiveness of our approach in performing real-world tasks using a robotic platform. We have generated a demo video that has been uploaded as part of the supplementary materials.

[2]The link for the questionnaire-based experiment results evaluation is http://150.158.148.22/

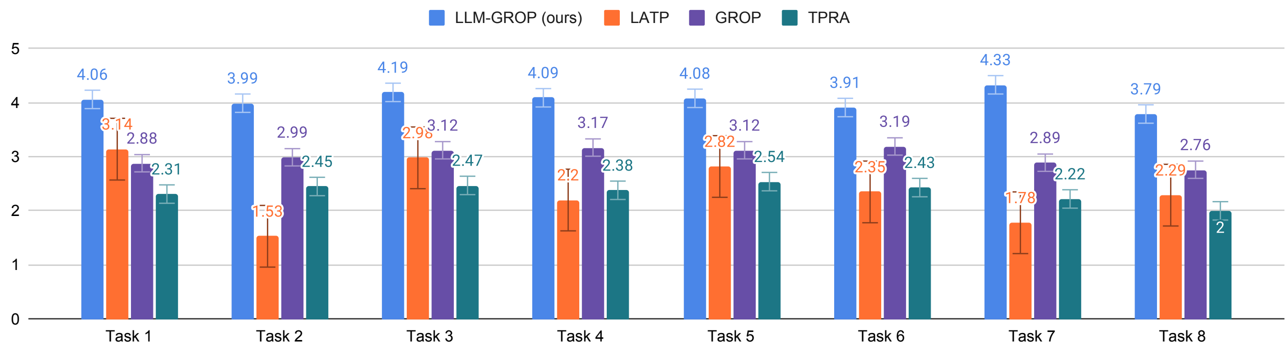

Fig. 6: User ratings of individual object rearrangement tasks, with the x-axis representing the task and the y-axis representing the user rating score. It can be observed that LLM-GROP consistently performs the best compared to baselines. Tasks 1-5 involve three objects, tasks 6 and 7 involve four objects, and task 8 involves five objects. The numerical value displayed on each bar indicates the mean rating for the corresponding task.

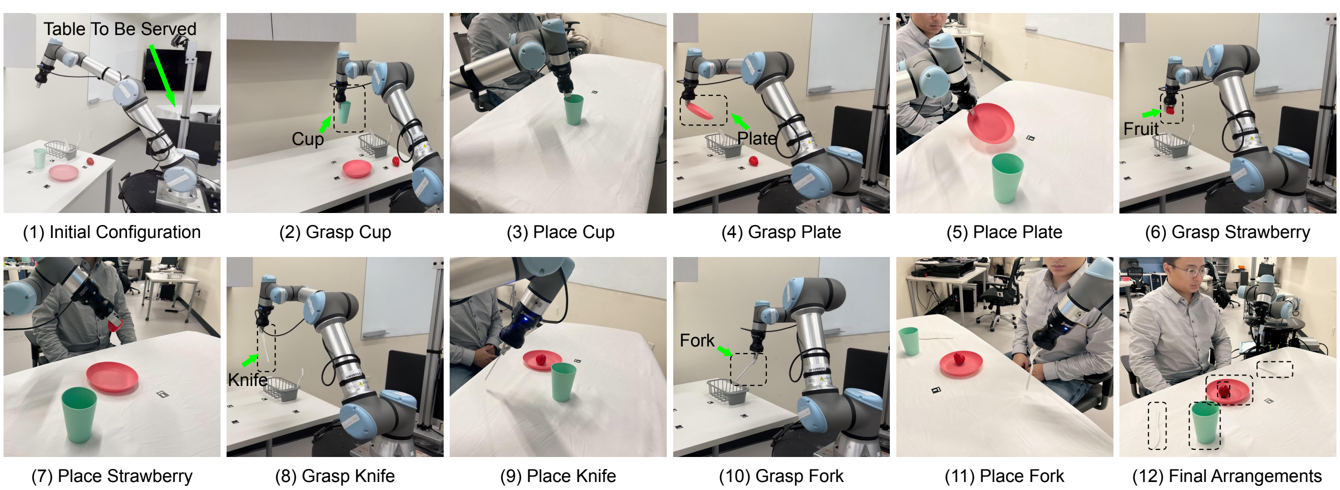

Fig. 7: We demonstrate LLM-GROP on real robot hardware. The task is to serve a human with a knife, a fork, a cup, a plate, and a strawberry. The robot computes a plan that successfully avoids chairs and the human around the table, while being able to place the target objects in plausible physical positions.