# LLM$^3$: Large Language Model-based Task and Motion Planning with Motion Failure Reasoning

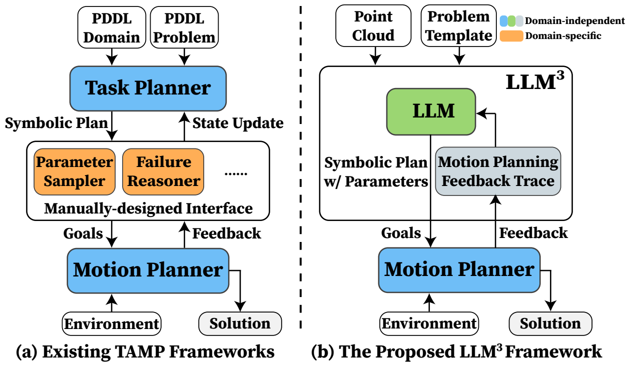
Shu Wang [*1] Muzhi Han [*1] Ziyuan Jiao [*2] Zeyu Zhang [2] Ying Nian Wu [1] Song-Chun Zhu [1] Hangxin Liu [2]

## Abstract

Conventional Task and Motion Planning (TAMP) approaches rely on manually crafted interfaces connecting symbolic task planning with continuous motion generation. These domain-specific and labor-intensive modules are limited in addressing emerging tasks in real-world settings. Here, we present LLM$^3$, a novel multi-modal foundation model TAMP framework featuring a domain-independent interface. Specifically, we leverage the powerful reasoning and planning capabilities of foundation models to propose symbolic action sequences and select continuous action parameters for motion planning. Through a series of simulations in a box-packing domain, we quantitatively demonstrate the effectiveness of our method. Ablation studies underscore the significant contribution of motion failure reasoning to the success of LLM$^3$. Furthermore, we conduct qualitative experiments on a physical manipulator, demonstrating the practical applicability of our approach in real-world settings. Code is available:https://github.com/AssassinWS/LLM-TAMP.

## 1. Introduction

Task and Motion Planning (TAMP) formulates a promising methodology that hierarchically decomposes planning into two stages: the high-level symbolic task planning stage reasons over long-horizon abstract action sequences, and the low-level continuous motion planning stage computes feasible trajectories subject to geometric constraints. In recent years, TAMP has enabled significant advances in diverse applications (Dantam et al., 2016; Toussaint et al., 2018; Garrett et al., 2021; Jiao et al., 2021; 2022; Su et al., 2023). However, a persistent challenge remains to properly interface between the task planner and the motion planner to efficiently solve TAMP, , generating action sequences that satisfy both symbolic task goals and continuous motion constraints.



(a) Existing TAMP Frameworks (b) The Proposed LLM³ Framework

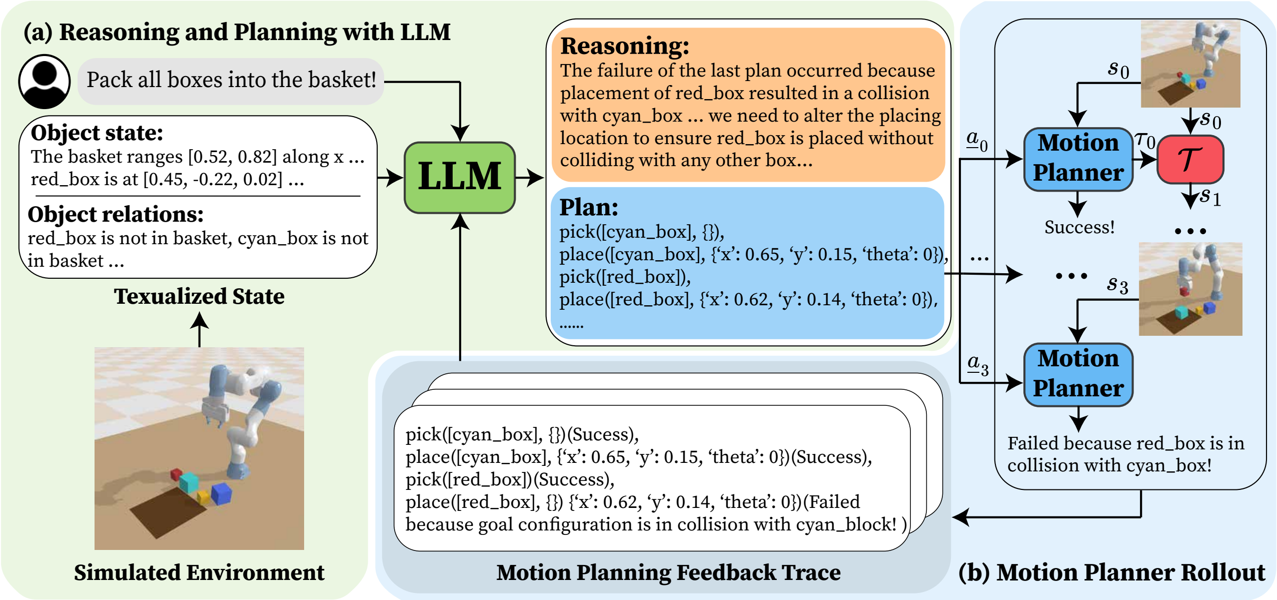
*Figure 1.* **The proposed LLM$^3$framework.** (a) Traditional TAMP frameworks rely on manually designed, domain-specific modules for interfacing between task and motion planners. (b) In contrast, we leverage a pre-trained LLM to iteratively propose refined plans and action parameters, by reasoning on motion planning failures.

Traditional TAMP approaches often rely on manually designed modules to interface between symbolic and continuous domains, as depicted in Figure 1(a). These modules serve two key roles. First, they act as action parameter samplers that generate real-valued parameters for symbolic actions. Previous works propose to learn heuristic parameter samplers from data (Chitnis et al., 2016; Wang et al., 2018), they are tailored to specific domains and lack generalizability. Second, these modules implement mechanisms to incorporate motion failure into the task planner to generate improved action plans, by updating the symbolic state (Srivastava et al., 2014). However, they usually require domain-specific design by human experts. In summary, these modules are domain-specific and require substantial manual effort to design, which hinders generalizability to novel environments.

Recent Multi-Modal Foundation Models (Fei et al., 2022)

[*]Equal contribution [1]University of California, Los Angeles [2]State Key Laboratory of General Artifical Intelligence. Correspondence to: Ziyuan Jiao <jiaoziyuan@bigai.ai>.

*Multi-modal Foundation Model meets Embodied AI Workshop in the $41^{st}$ International Conference on Machine Learning*, Vienna, Austria, 2024. Copyright 2024 by the author(s).


(a) Reasoning and Planning with LLM
Pack all boxes into the basket!
Object state:
The basket ranges [0.52, 0.82] along x ...
red_box is at [0.45, -0.22, 0.02] ...
Object relations:
red_box is not in basket, cyan_box is not in basket ...
Texualized State
Simulated Environment
LLM
Reasoning:
Plan:
Motion Planning Feedback Trace
$\underline{a}_0$
$\underline{a}_3$
$s_0$
$s_0$
$\tau_0$
$\mathcal{T}$
$s_1$
$s_3$
Motion Planner
Motion Planner
Success!
Failed because red_box is in collision with cyan_box!
(b) Motion Planner Rollout

The failure of the last plan occurred because placement of red_box resulted in a collision with cyan_box ... we need to alter the placing location to ensure red_box is placed without colliding with any other box...

```
pick([cyan_box], {}),
place([cyan_box], {'x': 0.65, 'y': 0.15, 'theta': 0}),
pick([red_box]),
place([red_box], {'x': 0.62, 'y': 0.14, 'theta': 0}),
......
```

```
pick([cyan_box], {})(Sucess),
place([cyan_box], {'x': 0.65, 'y': 0.15, 'theta': 0})(Success),
pick([red_box])(Success),
place([red_box], {}) {'x': 0.62, 'y': 0.14, 'theta': 0})(Failed
because goal configuration is in collision with cyan_block! )
```

*Figure 2.* **The system diagram of the proposed LLM$^3$framework.** (a) We show an example of utilizing a pre-trained LLM for reasoning and generating action sequences. (b) The feasibility of the proposed action sequence is verified by rollout with a motion planner and transition function $\mathcal{T}$. The motion planning feedback is saved into a trace that is provided to the LLM in the next iteration.

have demonstrated emergent capabilities in reasoning (Kojima et al., 2022) and planning (Huang et al., 2022). Our intuition is that pre-trained LLMs could provide a general and domain-independent approach to interfacing between symbolic and continuous domains for TAMP, eliminating the need to design domain-specific modules manually.

In this paper, we present **LLM$^3$** (**L**arge **L**anguage **M**odel-based Task and **M**otion Planning with **M**otion Failure Reasoning), an LLM-powered TAMP framework that reasons over motion planning feedback for effective planning Figure 1(b). Specifically, LLM$^3$observes object information in point clouds from a RGB-D camera and then employs a pre-trained LLM to (i) propose symbolic action sequences towards the task goal, (ii) generate continuous action parameters that lead to feasible motion, and (iii) reason over motion planning feedback to iteratively refine the proposed symbolic actions and parameters.

We evaluate LLM$^3$in a simulated tabletop box-packing task, which poses challenges in reasoning about potential failure modes, collisions, and unreachable areas, throughout the sequential manipulation planning problem. Quantitative results demonstrate the effectiveness of LLM$^3$, with ablation studies verifying: (i) reasoning over motion feedback significantly improves success rates and planning efficiency, and (ii) the LLM-based parameter sampler is substantially more sample efficient than a random sampler. Furthermore, we conduct real-robot experiments to show that LLM$^3$can be applied to real-world problems.

## 2. Related Work

### 2.1. Task and Motion Planning

Traditional TAMP approaches employ a high-level task planner to generate symbolic action sequences and a low-level motion planner to generate motion trajectories. The task planner requires pre-designed symbolic planning domains represented in formatted representations, such as Planning Domain Definition Language (PDDL). Significant efforts have been made to develop manually engineered modules that interface the task planner and motion planner, such as incorporating motion-level constraints into task planning (Garrett et al., 2020; Jiao et al., 2021), making approximations at the motion level (Hauser & Ng-Thow-Hing, 2011; Toussaint, 2015), and designing specialized communication modules (Srivastava et al., 2014). However, manually defining task planning domains and interface modules to fully capture real-world complexity is impractical. Furthermore, as the action space grows, searching for geometrically feasible symbolic action sequences becomes computationally challenging without effective heuristics (Garrett et al., 2020). In this work, we employ a pre-trained LLM as both the task planner and the interface between task and motion. We expect that semantic knowledge in the LLM can provide domain-independent heuristics for TAMP.

### 2.2. Robot Planning with Multi-Modal Foundation Models

Recent Multi-Modal Foundation Models encode vast world knowledge and exibit the emergent capability for planning (Huang et al., 2022; Li et al., 2022) through few-shot or zero-shot in-context learning (Brown et al., 2020; Dong et al., 2022). Pre-trained LLMs have been applied for task planning of robots or embodied agents (Ahn et al., 2022; Huang et al., 2023; Liang et al., 2023; Singh et al., 2023; Wang et al., 2023b;a; Yao et al., 2022; Gong et al., 2023a;b; Cui et al., 2024). Notably, Inner Monologue (Huang et al., 2023) takes in textualized environment feedback and generate actions to execute, while ReAct (Yao et al., 2022) further advanced this closed-loop approach by integrating reasoning and acting. Voyager (Wang et al., 2023a) focus on developing open-ended embodied agents that iteratively replan based on execution failure in PC games. Our usage of mutimodal LLMs in TAMP is inspired by many of the above works; however, the major difference is that we leverage the LLM as the core component of our TAMP framework.

## 3. Method

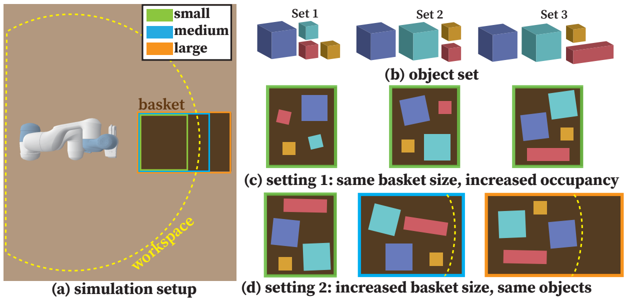
The system diagram of LLM$^3$is shown in Figure 2. Below, we elaborate on the overall LLM$^3$framework, reasoning and planning with the pre-trained LLM, and the designed motion planning feedback.

### 3.1. The LLM$^3$ Framework

As shown in Figure 2, the LLM$^3$framework iterates between: (i) reasoning on previous motion failure and generating an action sequence (, symbolic actions and continuous parameters) with a pre-trained LLM, and (ii) verifying the feasibility of the action sequence with a motion planner. Overall, the LLM$^3$framework can be regarded as a search-then-sample TAMP planner that generates action sequences with incrementally improved quality, guided by the intrinsic heuristics of the foundation model and the previous motion failure. We expect LLM$^3$to exhibit superior efficiency compared to unguided planners that sample action parameters randomly.

### 3.2. Planning and Reasoning with Foundation Models

We prompt a LLM to generate motion failure reasoning and action sequences in text format. Since we want to limit the domain-specific prior provided to the LLM, we use zero-shot prompting (Kojima et al., 2022). We implement two strategies for the LLM to generate a new action sequence that improves on the previous one: (i) *backtrack*, where we expect the LLM to backtrack to a previous action that has feasible motion, and continual to generate actions that complete the plan, and (ii) *from scratch*, where we expect the LLM to directly generate a new action sequence that attempts to avoid the motion failure happened in the history.



(a) simulation setup

(b) object set

(c) setting 1: same basket size, increased occupancy

(d) setting 2: increased basket size, same objects

*Figure 3.* **The box-packing task setup in a simulated environment.** (a) The task requires the robot to place one of (b) three sets of objects fully into the basket. (c) In setting 1, the total object size increases but the basket sizes remain the same. All baskets are fully reachable by the robot. (d) In setting 2, the basket size increases, but some portions of baskets are longer within the robot's reach.

### 3.3. Synthesizing Motion Planning Feedback

We implement a ground action $\underline{a}$ by calculating a collision-free trajectory $\tau$ with a sampling-based motion planner, BiRRT (LaValle, 2006). By default, the motion planner reports a binary signal that indicates whether there is a feasible trajectory. It does not give more abstract-level feedback. We additionally synthesize semantically meaningful motion-level feedback so that LLM$^3$can improve on previous failures more effectively. We observe that typical motion planning failures can be categorized into two types, collisions and unreachability. In practice, we integrate the motion planner with an additional IK solver and collision checker for obtaining these feedbacks.

## 4. Simulation and Experiment

In simulations, we initially perform an ablation study on our LLM$^3$framework in two settings of the tabletop box-packing task, quantitatively evaluating its effectiveness. Additionally, we demonstrate the role of LLM as an informed action parameter sampler by comparing it to a baseline utilizing random sampling strategies. Finally, we validate the proposed LLM$^3$framework through experimentation on a perception-integrated physical robotic manipulator, confirming its validity in real-world scenarios.

### 4.1. Simulation Setup

We developed a PyBullet-based simulation environment for our box-packing tasks, as illustrated in Figure 3. In **Setting 1**, three different sets of objects are given with increasing total sizes, while the size of the basket remains constant.

*Table 1.* Ablation Study

| **Method** | **Setting 1** | | | | | | | | | **Setting 2** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Easy** | | | **Medium** | | | **Hard** | | | **Small** | | | **Medium** | | | **Large** | | |
| | **%SR** | **#LM** | **#MP** | **%SR** | **#LM** | **#MP** | **%SR** | **#LM** | **#MP** | **%SR** | **#LM** | **#MP** | **%SR** | **#LM** | **#MP** | **%SR** | **#LM** | **#MP** |
| LLM$^3$Backtrack | **100** | **1.6** | **11.8** | **100** | **4.4** | **28.4** | 60 | 11.4 | 39.8 | 60 | 11.4 | 39.8 | **80** | **9.5** | **50** | 50 | 13.5 | 44.8 |
| Backtrack | 100 | 1.8 | 12.6 | 90 | 6.3 | 32 | 40 | 15.1 | 55.3 | 40 | 15.1 | 55.3 | 30 | 14.6 | 16 | 30 | 15.8 | 48 |
| LLM$^3$Scratch | 100 | 1.7 | 13.2 | 100 | 7 | 46.1 | **70** | **8.8** | **30.9** | **70** | **8.8** | **30.9** | 70 | 11.5 | 50.2 | **60** | **10.6** | **32** |
| Scratch | 100 | 2.4 | 17.6 | 60 | 12.3 | 45.3 | 50 | 13.7 | 42.8 | 50 | 13.7 | 42.8 | 30 | 16.2 | 45.7 | 40 | 13.2 | 24 |

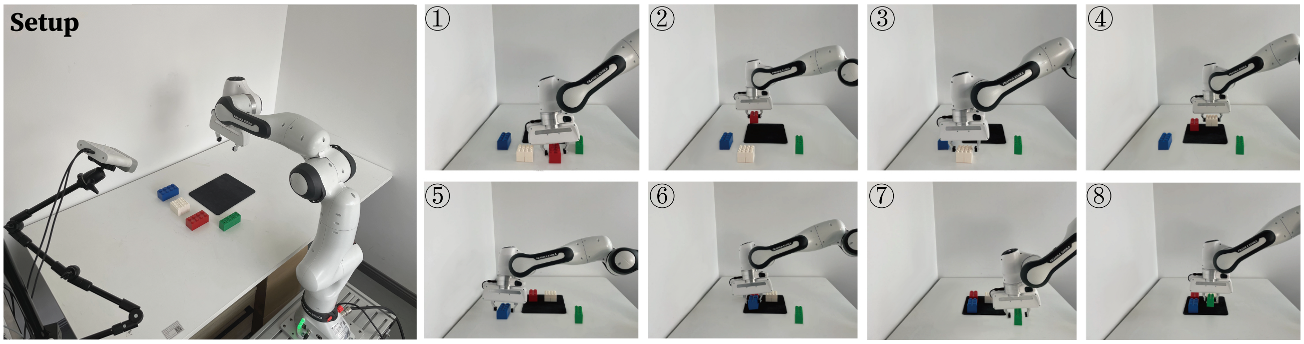


*Figure 4.* **The real-world experiment on a physical robot.** The figure to the left shows the box-packing task setup. Actions 1 to 8 are proposed by LLM$^3$and successfully carried out by the physical manipulator.

This task requires the LLM$^3$to oversee potential collisions between objects or the robot throughout the action sequence. The LLM must reason why collisions occur and adjust previous actions to ensure feasible task and motion plans. **Setting 2** involves placing the Set 3 objects into baskets of increasing sizes. Here, the robot cannot access the entire basket region but encounters a collision likelihood similar to the most crowded condition in Setting 1. Throughout the simulations, we utilize GPT-4 Turbo (OpenAI, 2023) as the LLM planner and BiRRT (LaValle, 2006) as the motion planner, with 100 attempts for each setting.

### 4.2. Ablation Study

The conducted ablation study compares the proposed LLM$^3$with baseline methods: 1) **LLM$^3$Backtrack**: The proposed LLM$^3$framework backtrack variant. 2) **Backtrack**: It proposes plans with backtracking without motion planning feedback. 3) **LLM$^3$Scratch**: The proposed LLM$^3$framework from scratch variant. It replans the entire action sequence if any action fails, incorporating motion planning feedback. 4) **Scratch**: It plans the action sequence only once and executes the plan without any feedback.

Three evaluation criteria are considered: The number of LLM calls (#LM), the total success rate %SR, and the number of motion planner calls #MP. The study results are summarized in Table 1. It indicates that the LLM can reason about failures from motion planning feedback, and propose adjusted task plans and action parameters that are more likely to produce feasible motions.

To validate the effectiveness of our proposed method in a real-world setting, we conducted an experiment using a Franka Research 3 manipulator. The robot observed a single point cloud from a third-person-view RGB-D camera (Kinect Azure), capturing the workspace containing various objects such as blocks and a plate. To identify and locate individual objects, we employed (Ren et al., 2024) for object segmentation. This approach yielded per-object point clouds, essential for planning and executing manipulation tasks.

Figure 4 presents a qualitative evaluation of our method, where the robot was tasked with placing all blocks on the plate. The results demonstrate that our method enabled the robot to successfully identify and manipulate objects despite the uncertainties in the environment.

## 5. Conclusions

In this paper, we introduced LLM$^3$, which leverages the rich knowledge encoded in and the powerful reasoning capability processed by LLMs. Our study also revealed that although the LLM can generate action parameters more efficiently than random samplers, it still necessitated multiple feedback iterations and motion planner calls. Looking ahead, fine-tuning multi-modal foundation models holds promise for empowering robots to tackle emerging tasks in real-world scenarios.

## References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chitnis, R., Hadfield-Menell, D., Gupta, A., Srivastava, S., Groshev, E., Lin, C., and Abbeel, P. Guided search for task and motion plans using learned heuristics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

Cui, J., Liu, T., Liu, N., Yang, Y., Zhu, Y., and Huang, S. Anyskill: Learning open-vocabulary physical skill for interactive agents. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Dantam, N. T., Kingston, Z. K., Chaudhuri, S., and Kavraki, L. E. Incremental task and motion planning: A constraint-based approach. In *Robotics: Science and Systems (RSS)*, volume 12, pp. 00052. Ann Arbor, MI, USA, 2016.

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

Fei, N., Lu, Z., Gao, Y., Yang, G., Huo, Y., Wen, J., Lu, H., Song, R., Gao, X., Xiang, T., et al. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 13(1):3094, 2022.

Garrett, C. R., Lozano-Pérez, T., and Kaelbling, L. P. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2020.

Garrett, C. R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L. P., and Lozano-Pérez, T. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 2021.

Gong, R., Gao, X., Gao, Q., Shakiah, S., Thattai, G., and Sukhatme, G. S. Lemma: Learning language-conditioned multi-robot manipulation. *IEEE Robotics and Automation Letters*, 2023a.

Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., Zheng, Z., Zhu, S.-C., Terzopoulos, D., Fei-Fei, L., et al. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023b.

Hauser, K. and Ng-Thow-Hing, V. Randomized multi-modal motion planning for a humanoid robot manipulation task. *International Journal of Robotics Research (IJRR)*, 30(6):678–698, 2011.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning (ICML)*, 2022.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2023.

Jiao, Z., Zhang, Z., Wang, W., Han, D., Zhu, S.-C., Zhu, Y., and Liu, H. Efficient task planning for mobile manipulation: a virtual kinematic chain perspective. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

Jiao, Z., Niu, Y., Zhang, Z., Zhu, S.-C., Zhu, Y., and Liu, H. Sequential manipulation planning on scene graph. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 22199–22213, 2022.

LaValle, S. M. *Planning algorithms*. Cambridge university press, 2006.

Li, S., Puig, X., Paxton, C., Du, Y., Wang, C., Fan, L., Chen, T., Huang, D.-A., Akyürek, E., Anandkumar, A., et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:31199–31212, 2022.

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. Progprompt: Generating situated robot task plans using large language models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Srivastava, S., Fang, E., Riano, L., Chitnis, R., Russell, S., and Abbeel, P. Combined task and motion planning through an extensible planner-independent interface layer. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

Su, Y., Li, J., Jiao, Z., Wang, M., Chu, C., Li, H., Zhu, Y., and Liu, H. Sequential manipulation planning for overactuated unmanned aerial manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

Toussaint, M. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Toussaint, M. A., Allen, K. R., Smith, K. A., and Tenenbaum, J. B. Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems (RSS)*, 2018.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Wang, Z., Garrett, C. R., Kaelbling, L. P., and Lozano-Pérez, T. Active model learning and diverse action sampling for task and motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

Wang, Z., Cai, S., Chen, G., Liu, A., Ma, X., and Liang, Y. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2022.