Any House Any Task: Scalable Long-Horizon Planning for Abstract Human Tasks

Zhihong Liu[1,3,*], Yang Li[1,4,*], Renming Huang[1,2], Cewu Lu[1,2] and Panpan Cai[1,2,†]
[1]Shanghai Innovation Institute [2]Shanghai Jiao Tong University
[3]Xi'an Jiao Tong University [4]Nankai University
[*]Equal contribution [†]Corresponding author

***Abstract*—Open-world language-conditioned task planning is crucial for robots operating in large-scale household environments. While many recent works attempt to address this problem using Large Language Models (LLMs) via prompting or training, a key challenge remains: handling large environments, long planning horizons, ambiguous instructions, and complex constraints. In this work, we propose Any House Any Task (AHAT), a household task planner optimized for long-horizon planning in large environments given abstract human instructions. At its core, AHAT utilizes an LLM trained to map task instructions and textual scene graphs into grounded subgoals defined in the Planning Domain Definition Language (PDDL). These subgoals are subsequently solved to generate feasible and long-horizon plans through explicit symbolic reasoning. To enhance the model's ability to decompose complex and ambiguous intentions, we introduce TGPO, a novel reinforcement learning algorithm that integrates external correction of intermediate reasoning trace into Group Relative Policy Optimization (GRPO). Experiments demonstrate that AHAT achieves significant performance gains over strong general-purpose LLMs and prompting-based planning baselines, particularly in human-style household tasks characterized by brief instructions but requiring complex execution plans.**

I. Introduction

Long-horizon task planning in large household environments remains a key challenge for embodied agents. In realistic homes, robots must reason over many objects, rooms, and constraints, while interpreting brief and abstract human instructions that often omit explicit subtasks or target objects. Prior works such as SayPlan [15] and DELTA [14] study related settings, but mainly focus on more concrete instructions and relatively small action spaces. Their performance can degrade when instructions become ambiguous, environments become larger, and tasks require longer execution horizons.

To address these challenges, we propose Any House Any Task (AHAT), a scalable household task planner for abstract human instructions in large environments. AHAT maps a user instruction and a textual scene graph into a sequence of grounded PDDL subgoals, which are then solved by an off-the-shelf symbolic planner to produce executable long-horizon plans. This design separates language-level reasoning from symbolic constraint satisfaction: the LLM focuses on intent interpretation, task decomposition, and subgoal grounding, while the planner ensures feasibility and synthesizes action sequences.

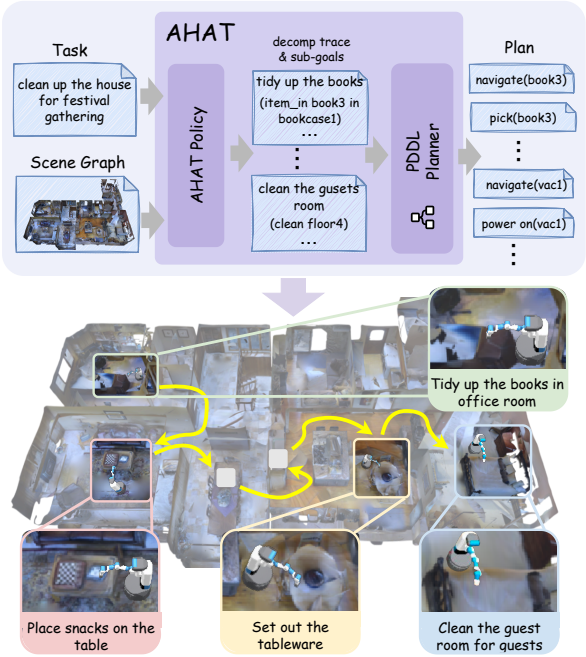

Fig. 1: In large-scale environments, AHAT receives abstract instructions and a scene graph, generates a decomposition trace and corresponding subgoals. These subgoals are then solved using a PDDL planner, resulting in an executable long-horizon plan that satisfies the user's requirements.

A central difficulty is learning to decompose abstract instructions into correct intermediate goals. Standard supervised fine-tuning provides useful initialization, but does not directly optimize planning success. Moreover, sparse task-level rewards make reinforcement learning difficult, since failures often arise from missing subtasks, incorrect ordering, or misunderstood user intent. We therefore introduce Trace-Guided Policy Optimization (TGPO), a reinforcement learning method that improves failed intermediate reasoning traces using external feedback. The revised traces guide constrained rollouts and are also included in the training loss, allowing the policy to learn improved task decompositions together with grounded PDDL subgoal generation.

To train and evaluate AHAT, we construct a large-scale household planning dataset with 50k tasks generated from 308 household scene graphs and 1.6k diverse user personas.

The tasks cover different abstraction levels, ranging from explicit object-centric commands to high-level human intents. We evaluate AHAT on held-out synthetic tasks, crowd-sourced human tasks, and two public household planning benchmarks, Behavior-1K [12] and PARTNR [3].

Experiments show that AHAT consistently outperforms strong general-purpose LLMs and prompting-based planning baselines, including GPT-5, Gemini-3, SayPlan, and DELTA. The improvement is especially significant on abstract and long-horizon tasks, where baselines struggle with intent inference and symbolic constraints.

II. Background and Related Works

Symbolic planning. Symbolic planners formulate task planning as searching for a sequence of discrete actions under explicit logical constraints. A common formalism is PDDL [1], where a problem is defined by a domain, an initial state, and a goal condition. Given grounded predicates and action schemas, a PDDL planner such as Fast Downward [9] can synthesize feasible action sequences with strong constraint satisfaction. However, symbolic planning requires well-formed goals and its search complexity grows rapidly in large object-rich environments.

LLM-based household planning. Recent works use LLMs for language-conditioned household task planning. Prompting-based methods, such as SayPlan [15], encode environments as structured representations and query general-purpose LLMs to generate or refine plans. Other works improve planning through memory, interaction history, retrieval, or few-shot demonstrations [4, 7, 17, 18]. Training-based methods further adapt models using supervision or reinforcement learning to improve task feasibility and preference alignment [8, 11, 22]. Nevertheless, end-to-end LLM planners often struggle with long horizons, large scenes, and complex symbolic constraints.

Hybrid LLM-symbolic planning. A growing line of work combines LLMs with symbolic planners by prompting LLMs to translate natural-language instructions into PDDL goals or problems, which are then solved by a planner [2, 5, 13, 16]. Some methods improve correctness through iterative refinement, verification, or feedback [6, 20, 21, 23]. DELTA [14] decomposes long-horizon household tasks into subtasks before symbolic planning. In contrast, AHAT trains a model to generate PDDL subgoals directly and optimizes it with reinforcement learning, improving robustness for abstract instructions and large household scenes.

III. Overview

AHAT is a long-horizon household task planner that takes an abstract natural-language instruction and a household scene representation as input, and outputs an executable plan through LLM-based subgoal generation and symbolic planning. At a high level, AHAT first decomposes the instruction into natural-language subtasks, grounds each subtask into PDDL subgoals, and then invokes a symbolic planner to synthesize executable actions.

A. AHAT Planning

We represent the household environment as a textual scene graph $\mathrm{SG} = (V, E)$, where nodes correspond to rooms, furniture, and objects, and edges encode spatial relations such as `in` and `on`. Each node also stores planning-relevant states, such as `open/closed`, `clean/dirty`, or `on/off`.

As shown in Fig. 1, given a task instruction q and scene graph SG, AHAT first generates a natural-language decomposition trace $T = \{t_k\}_{k=1}^{K}$, where each t_k describes a high-level subtask. It then grounds each subtask into a PDDL subgoal g_k and a set of relevant objects obj_k. For each subtask, we construct a PDDL problem from the current scene state, the predicted subgoal, and the selected objects, and solve it using an off-the-shelf PDDL planner. The resulting subplans are executed sequentially and concatenated into the final long-horizon plan.

This design decouples language-level reasoning from symbolic constraint satisfaction: the LLM focuses on interpreting abstract user intent, decomposing the task, and grounding subtasks into formal goals, while the PDDL planner handles feasibility checking and low-level plan synthesis.

B. AHAT Training

As shown in Fig. 2, AHAT is initialized from Qwen2.5-7B and trained in two stages. We first construct a synthetic household planning dataset using scene graphs and user personas (Fig. 2a). Each sample contains a task instruction, a scene graph, a decomposition trace, grounded PDDL subgoals, and an executable plan.

We then perform supervised fine-tuning (SFT) on this dataset (Fig. 2b), which equips the model with basic capabilities for task decomposition and PDDL grounding. After SFT, we further optimize the model using Trace-Guided Policy Optimization (TGPO), as illustrated in Fig. 2c.

In each RL iteration, AHAT samples multiple candidate decompositions and subgoals. The resulting plans are evaluated by two rewards: feasibility, checked by the PDDL planner, and task completion, judged by an auxiliary LLM reviewer. For failed candidates, an external trace improver revises the decomposition trace using planner and reviewer feedback. The policy then performs a trace-guided rollout in which the revised trace tokens are constrained during sampling, while the PDDL subgoals are generated autoregressively. Although constrained during rollout, the revised trace tokens are included in the optimization loss, allowing the policy to imitate improved decompositions. The original and trace-guided candidates are jointly used to compute group-relative advantages and update the policy.

IV. Data Generation & Annotation

We construct a large-scale synthetic corpus to train AHAT for AHAT-style planning outputs, including natural-language decomposition traces and grounded PDDL subgoals. The dataset is built from 308 household scenes curated from HSSD [10] and Gibson [19], each converted into a textual scene graph. To increase task diversity and realism, we create 1.6k user personas with different ages, occupations, cultural

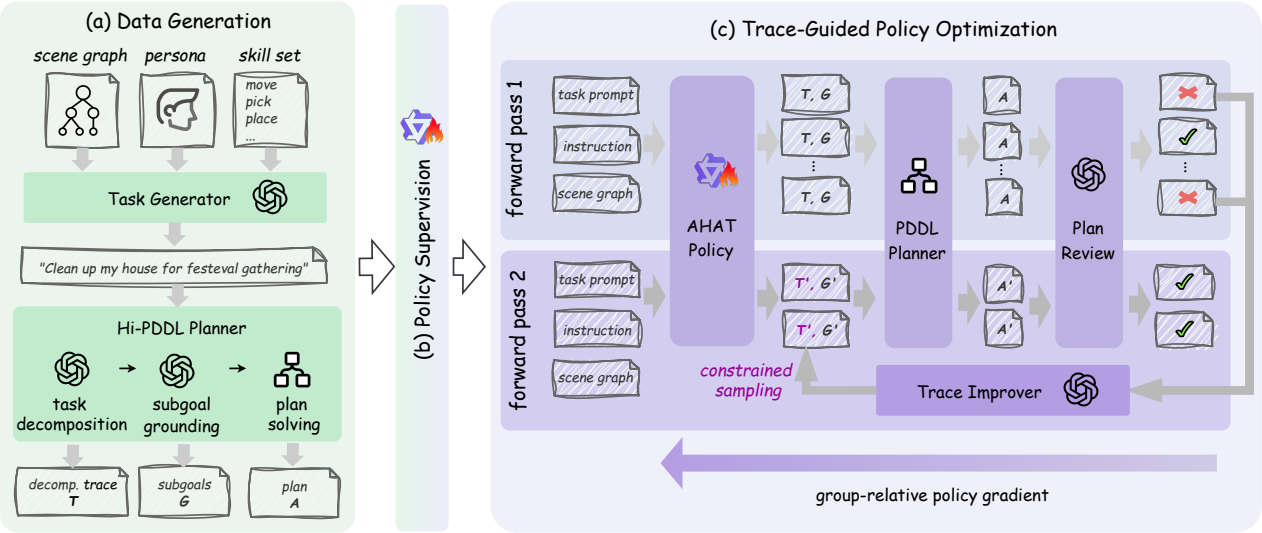

Fig. 2: Overview of AHAT. *(a) Data Generation*: Task synthesis and annotation. *(b) Policy Supervision*: Supervised fine-tuning (SFT) on the constructed long-horizon household planning dataset. *(c) Trace-Guided Policy Optimization*: The reinforcement learning loop that integrates external correction of intermediate reasoning traces, improving subgoal generation and task decomposition through constrained sampling, and optimizing the AHAT model for robust planning performance.

backgrounds, and daily habits. Given a scene graph, a persona, and the robot's predefined skill set, we prompt an LLM to synthesize long-horizon household instructions. Multiple prompt templates are used to cover different abstraction levels, from explicit object-centric commands to high-level user intents. This process produces a large pool of candidate household planning tasks.

To obtain supervision labels, we develop Hierarchical PDDL Planner (Hi-PDDL), a prompting-based annotation pipeline that follows the same hierarchical planning procedure as AHAT, but replaces the trained AHAT policy with a strong general-purpose LLM. For each task, Hi-PDDL generates a decomposition trace and grounded PDDL subgoals, solves the subgoals sequentially using a PDDL planner, and concatenates the resulting subplans into a complete executable plan. We retain only samples that satisfy both feasibility and task-completion checks. When a sample fails, feedback from the PDDL planner and the LLM reviewer is used to revise the trace or subgoals within a limited retry budget. If the reviewer detects missing task-critical objects, the scene graph is repaired by adding the suggested objects and the annotation process is rerun.

The final dataset contains 50k tasks with decomposition traces, grounded PDDL subgoals, and executable plans. We use this corpus for supervised fine-tuning of Qwen2.5-7B, which provides a strong initialization for the subsequent reinforcement learning stage.

V. Trace-Guided Policy Optimization

Supervised fine-tuning provides AHAT with the basic ability to produce task decompositions and PDDL subgoals, but it does not directly optimize for planning success. In long-horizon household tasks, failures often originate from errors in the intermediate decomposition trace, such as missing necessary subtasks, incorrect ordering, or misunderstanding abstract user intent. Since the final reward is sparse, standard reinforcement learning has difficulty identifying which part of the reasoning caused the failure.

To address this issue, we propose *Trace-Guided Policy Optimization* (TGPO), a reinforcement learning procedure built on top of GRPO. The key idea is to improve failed intermediate reasoning traces using external feedback, and then train the policy to regenerate PDDL subgoals conditioned on the improved traces. Each TGPO iteration contains two rollout stages.

In the first stage, the current AHAT policy samples a group of candidate outputs for each task. Each candidate consists of a natural-language decomposition trace and a set of grounded PDDL subgoals. We evaluate each candidate using two criteria. First, *feasibility* checks whether all predicted subgoals can be solved by the PDDL planner. Second, *task completion* checks whether the resulting action sequence satisfies the user instruction, which is judged by an auxiliary LLM reviewer. The final reward combines these two signals, assigning high reward only to plans that are both executable and aligned with the user's intent.

In the second stage, TGPO focuses on failed candidates. For outputs with low reward, we collect feedback from the PDDL planner and the LLM reviewer, and use an auxiliary LLM, called the *trace improver*, to revise only the decomposition trace. The AHAT policy then performs another rollout conditioned on this revised trace. During this rollout, the corrected trace tokens are fixed through token-level constrained sampling, while the policy autoregressively generates the corresponding PDDL subgoals and object arguments. Although the corrected trace tokens are constrained

TABLE I: Comparison of Success Rates (%) on AHAT Benchmark, Human Tasks, and Public Benchmarks

Model	Time(s)	AHAT Benchmark			Human Tasks	Public Benchmarks	
		Easy	Complex	Abstract		Behavior-1K	PARTNR
GPT-5	8.0	35.7 ± 4.9	21.3 ± 1.5	11.5 ± 5.0	25.7 ± 4.3	39.0 ± 9.3	69.2 ± 7.2
Gemini-3.0-pro	23.5	33.6 ± 3.3	17.0 ± 7.4	18.0 ± 6.1	12.6 ± 6.0	28.3 ± 8.2	71.8 ± 8.4
SayPlan	63.7	50.3 ± 4.2	26.5 ± 3.9	20.1 ± 3.3	26.0 ± 5.1	49.0 ± 7.0	72.2 ± 7.8
DELTA	37.8	50.5 ± 4.0	19.7 ± 5.2	9.1 ± 1.3	10.0 ± 4.6	36.1 ± 4.62	69.3 ± 6.1
AHAT-TGPO	**3.1**	**99.5** ± 0.4	**94.6** ± 2.4	**89.0** ± 2.8	**82.2** ± 2.9	**84.4** ± 6.6	**82.6** ± 5.1

during sampling, they are included in the optimization loss by computing their log-probabilities under the current policy. This allows the policy to imitate successful trace revisions instead of merely using them as conditioning context. The regenerated outputs are evaluated with the same reward function and added to the original candidate group.

Finally, AHAT is updated using the augmented group of original and trace-guided candidates. Following GRPO, we compute group-relative advantages within the augmented group and optimize the policy with a clipped policy-gradient objective and KL regularization to a reference policy. For trace-guided candidates, the loss is computed over both the constrained trace tokens and the policy-generated PDDL subgoal tokens, so successful corrected traces directly increase the likelihood of better decompositions. By comparing failed rollouts with trace-guided improved rollouts under the same task context, TGPO provides more informative learning signals and improves both task decomposition and symbolic subgoal grounding.

VI. Experiment

We evaluate AHAT on four task sources: our AHAT benchmark, human-authored household tasks, and two public benchmarks, Behavior-1K [12] and PARTNR [3]. These tasks cover synthesized and human-written instructions, explicit and abstract goals, and different scene scales. We report success rate (SR) and average planning time.

A. Experiment Setup

1) Tasks and Environments: **AHAT benchmark.** We construct the AHAT benchmark as a held-out split of our synthesized corpus, containing 346 tasks across 179 household scenes. The benchmark uses whole-house scene graphs with distractor objects, making the planning state space larger than benchmarks that only include task-relevant objects. We divide the tasks into three subsets: **Easy** tasks with explicit objects and short plans, **Complex** tasks with explicit but long-horizon and constraint-heavy instructions, and **Abstract** tasks that express high-level user needs and require intent inference before planning.

Human-authored tasks. To test realistic user instructions, we collect 50 human-authored tasks through a web interface where users write requests based on visualized scene graphs. These instructions are concise, averaging 5.7 words, and therefore require the planner to infer missing details from the scene.

Public benchmarks. We also evaluate on PARTNR [3] and Behavior-1K [12]. For PARTNR, we use the validation mini-split with 368 tasks in HSSD houses. For Behavior-1K, we remove non-household tasks from the public release and obtain 30 tasks. Since Behavior-1K scenes mainly contain task-relevant objects, we add plausible distractor objects to better reflect household clutter.

2) Metrics and Baselines: A plan is considered successful only if it is both executable and task-complete. Executability is checked by the PDDL planner, while task completion is judged by the reviewer. We report the average success rate over all tasks and the average planning time.

We compare AHAT with two types of baselines: general-purpose LLMs and prompting-based planners. For general-purpose LLMs, we directly prompt `GPT-5` and `Gemini-3-Pro` to generate household plans from the instruction and scene graph. For prompting-based planners, we compare against SayPlan [15] and DELTA [14], both of which rely on general-purpose LLMs for planning or task decomposition.

B. Main Results

Table I reports the success rate (SR) and average planning time across the AHAT benchmark, Human Tasks, and public benchmarks. Across all tests, AHAT improves the success rates over the strongest baseline by 49% on easy tasks, 68.1% on complex tasks, and 68.9% on abstract tasks, reducing 61% planning time from the fastest baseline. Notably, it remains robust on the Abstract tasks with high instruction ambiguity. On human tasks, AHAT achieves a success rate of 82.2%, outperforming the strongest baseline by 56.2%. Furthermore, on public benchmarks with concrete instructions, AHAT also significantly outperforms all baselines. The performance gain is attributed to both subgoal-based planning, which decouples reasoning over user intents with optimal planning under constraints, and TGPO reinforcement learning, where externally improved reasoning traces explicitly guide the policy to improve its task decomposition capability.

VII. Conclusion & Limitations

We present AHAT, a household task planner that addresses long-horizon planning in large environments from abstract human instructions. AHAT uses a model trained with TGPO, a reinforcement learning method, to predict a sequence of grounded subgoals, which are then solved sequentially by off-the-shelf PDDL planners. We further propose a data

generation and annotation pipeline that produces abstract and diverse instructions in large-scale environments, resulting in a synthetic dataset of 50K long-horizon household planning tasks. Experiments on the AHAT benchmark, Human Tasks, PARTNR and Behavior-1K demonstrate that AHAT substantially outperforms strong general-purpose LLMs and prompting-based planners, achieving higher success rates and stronger robustness across varying levels of task complexity.

Despite its strong performance, AHAT has a few limitations. It relies on structured textual scene representations, which may cause information loss, and future iterations will support direct visual input. Additionally, it operates under a closed domain assumption with predefined PDDL domains, and extending this to multiple domains is a key direction. Lastly, our experiments assume full observability, overlooking challenges like perceptual noise and environmental dynamics; integrating active exploration and online planning is a promising path forward.

References

[1] Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins Sri, Anthony Barrett, Dave Christianson, et al. Pddl—the planning domain definition language. *Technical Report, Tech. Rep.*, 1998.

[2] Timo Birr, Christoph Pohl, Abdelrahman Younes, and Tamim Asfour. Autogpt+ p: Affordance-based task planning with large language models. *arXiv preprint arXiv:2402.10778*, 2024.

[3] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv preprint arXiv:2411.00081*, 2024.

[4] Siwei Chen, Anxing Xiao, and David Hsu. Llm-state: Open world state representation for long-horizon task planning with large language model. *arXiv preprint arXiv:2311.17406*, 2023.

[5] Gautier Dagan, Frank Keller, and Alex Lascarides. Dynamic planning with a llm. *arXiv preprint arXiv:2308.06391*, 2023.

[6] Elliot Gestrin, Marco Kuhlmann, and Jendrik Seipp. Nl2plan: Robust llm-driven planning from minimal text descriptions. *arXiv preprint arXiv:2405.04215*, 2024.

[7] Marc Glocker, Peter Hönig, Matthias Hirschmanner, and Markus Vincze. Llm-empowered embodied agent for memory-augmented task planning in household robotics. *arXiv preprint arXiv:2504.21716*, 2025.

[8] Dongge Han, Trevor McInroe, Adam Jelley, Stefano V Albrecht, Peter Bell, and Amos Storkey. Llm-personalize: Aligning llm planners with human preferences via reinforced self-training for housekeeping robots. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1465–1474, 2025.

[9] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[10] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024.

[11] Zixing Lei, Sheng Yin, Yichen Xiong, Yuanzhuo Ding, Wenhao Huang, Yuxi Wei, Qingyao Xu, Yiming Li, Weixin Li, Yunhong Wang, et al. Embomatrix: A scalable training-ground for embodied decision-making. *arXiv preprint arXiv:2510.12072*, 2025.

[12] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.

[13] 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.

[14] Yuchen Liu, Luigi Palmieri, Sebastian Koch, Ilche Georgievski, and Marco Aiello. Delta: Decomposed efficient long-term robot task planning using large language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10995–11001. IEEE, 2025.

[15] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *7th Annual Conference on Robot Learning*.

[16] Keisuke Shirai, Cristian C Beltran-Hernandez, Masashi Hamaya, Atsushi Hashimoto, Shohei Tanaka, Kento Kawaharazuka, Kazutoshi Tanaka, Yoshitaka Ushiku, and Shinsuke Mori. Vision-language interpreter for robot task planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058. IEEE, 2024.

[17] 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.

[18] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009, 2023.

[19] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax,

Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[20] Haoming Ye, Yunxiao Xiao, Cewu Lu, and Panpan Cai. Unidomain: Pretraining a unified pddl domain from real-world demonstrations for generalizable robot task planning.

[21] Li Zhang, Peter Jansen, Tianyi Zhang, Peter Clark, Chris Callison-Burch, and Niket Tandon. Pddlego: Iterative planning in textual environments. *arXiv preprint arXiv:2405.19793*, 2024.

[22] Wenqi Zhang, Mengna Wang, Gangao Liu, Xu Huixin, Yiwei Jiang, Yongliang Shen, Guiyang Hou, Zhe Zheng, Hang Zhang, Xin Li, et al. Embodied-reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks. *arXiv preprint arXiv:2503.21696*, 2025.

[23] Zhehua Zhou, Jiayang Song, Kunpeng Yao, Zhan Shu, and Lei Ma. Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2081–2088. IEEE, 2024.