Multi-Modal Grounded Planning and Efficient Replanning For Learning Embodied Agents with a Few Examples

Taewoong Kim, Byeonghwi Kim, Jonghyun Choi*

Seoul National University
{twoongg.kim, byeonghwikim, jonghyunchoi}@snu.ac.kr

Abstract

Learning a perception and reasoning module for robotic assistants to plan steps to perform complex tasks based on natural language instructions often requires large free-form language annotations, especially for short high-level instructions. To reduce the cost of annotation, large language models (LLMs) are used as a planner with few data. However, when elaborating the steps, even the state-of-the-art planner that uses LLMs mostly relies on linguistic common sense, often neglecting the status of the environment at command reception, resulting in inappropriate plans. To generate plans grounded in the environment, we propose FLARE (FEW-SHOT LANGUAGE WITH ENVIRONMENTAL ADAPTIVE REPLANNING EMBODIED AGENT), which improves task planning using both language command and environmental perception. As language instructions often contain ambiguities or incorrect expressions, we additionally propose to correct the mistakes using visual cues from the agent. The proposed scheme allows us to use a few language pairs thanks to the visual cues and outperforms state-of-the-art approaches. Our code and the dataset are publicly available to facilitate further research.

1 Introduction

By the rapid advancement in the fields of computer vision, natural language processing, and embodied AI, we are witnessing a significant improvement in key functionalities of robotic assistants that can perform daily tasks. These functions include navigation (Anderson et al. 2018; Chaplot et al. 2017; Uppal et al. 2024), object manipulation (Zhu et al. 2017; Ryu et al. 2024), and responsive reasoning (Das et al. 2018; Gordon et al. 2018; Majumdar et al. 2024) in simulated 3D spaces (Ge et al. 2024; Chang et al. 2017; Xia et al. 2018; Kim et al. 2024). Practical robotic assistants require all of the aforementioned capabilities to understand language instructions and actively perceive the environment.

To learn an agent that performs such complex tasks, a straightforward approach is to train an agent in a supervised manner by a large amount of natural language instruction and action pairs (Shridhar et al. 2020; Ehsani et al. 2024; Pashevich et al. 2021; Blukis et al. 2021; Kim et al. 2023). However, annotating instructions and providing expert action sequences (*i.e.*, navigating trajectories) is costly and time-consuming, and thus collecting a sufficient amount of language annotations is often prohibitive. When data are insufficient, the above-mentioned data-driven approaches would not be effective (Min et al. 2022; Inoue and Ohashi 2022; Bhambri, Kim, and Choi 2023; Song et al. 2022).

To learn an agent performing a long-horizon task by a small amount of annotated data, recent approaches (Min et al. 2022; Inoue and Ohashi 2022) exploit manually designed action sequences for specific task types. However, such defined actions do not scale to various types of tasks. Another line of research uses large language models (LLMs) to address insufficient data, capitalizing on the remarkable advances achieved by prior knowledge encoded within

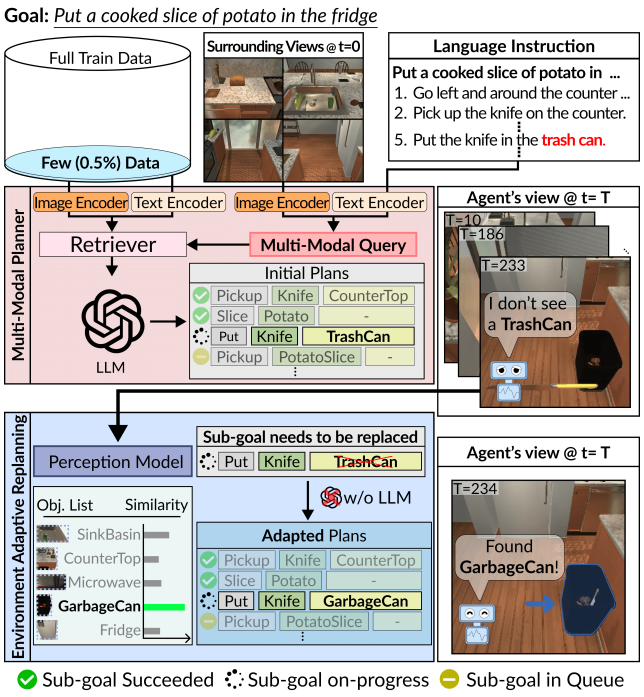

Figure 1: **Overview of the proposed FLARE.** Our agent comprises (1) Multi-Modal Planner (MMP)' for generating subgoals using initial views and instructions via an LLM (*e.g.*, GPT-4), and (2) Environment Adaptive Replanning (EAR)' for refining ungrounded plans using visual cues when agent gets stuck while executing a plan.

*JC is with ECE, ASRI & IPAI in SNU and a corresponding author.

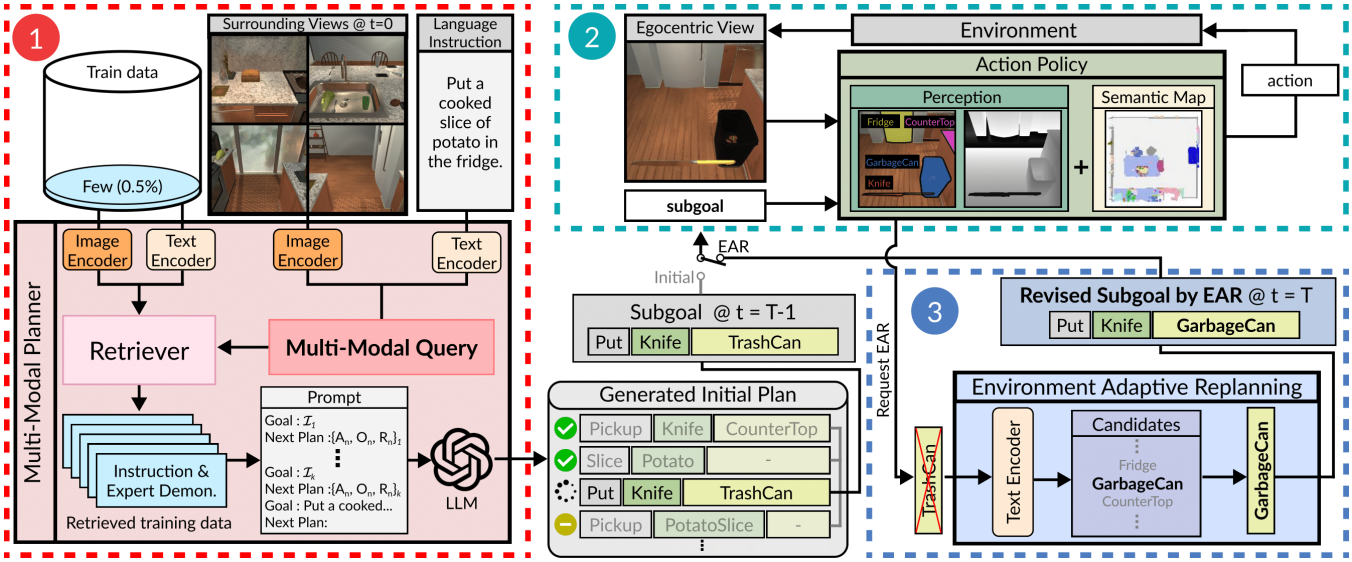

Figure 2: **Detailed architecture of FLARE.** 1 'Multi-Modal Planner (MMP)' retrieves the top k relevant training data pairs with instruction and expert demonstration (indicated with Expert Demon.), based on the agent's initial panoramic surrounding views and language instructions, then plans a sequence of actions through LLMs with these examples. 2 When agent fails to locate the target object, it requests replanning via EAR. 3 Using visual observations and semantic similarity, 'Environment Adaptive Replanning (EAR)' identifies the most similar object available within the scene and replaces the missing one.

LLMs in various domains (Zeng et al. 2023; Singh et al. 2023; Driess et al. 2023; Song et al. 2023; Sarch et al. 2023; Wu et al. 2023). In particular, (Song et al. 2023) proposes using LLMs as a high-level planner with a dynamic grounded replanning on top of an existing agent (Blukis et al. 2021), where human-annotated language is very scarce. But they neglect an environment state that could lead to the generation of implausible plans, since planning often needs to consider the state of the environment when it receives an instruction (*e.g.*, where the agent was located, what is visible from its view, *etc.*). To modify the inappropriate plan, (Song et al. 2023) invokes an LLM multiple times with the prompt that includes a list of observed objects to generate a grounded plan. However, the heavy reliance on large models makes this approach more costly than necessary, as it revises entire sequences when only partial subgoals are incorrect.

To address these issues, we propose FLARE (FEW-SHOT LANGUAGE WITH ENVIRONMENTAL ADAPTIVE REPLANNING EMBODIED AGENT) which considers the multimodal environmental context (*i.e.*, visual input, and language directive) when it begins to plan the subgoal sequences to complete the household task and efficiently (*i.e.*, partially) correct the plan without using the LLM by using visual input. To empirically validate the effectiveness of our approach, we adopt a widely used benchmark for embodied instruction following (Shridhar et al. 2020). We observe that FLARE can generate plausible plans with only a few, *e.g.*, 100 language and demonstration pairs, outperforming state-of-the-art methods in our empirical validation by noticeable margins up to $+24.46\%$ absolute gain in the test unseen split. We summarize our contributions as follows:

- Proposing a multi-modal planner that considers both the environmental status and the language instruction for a long-horizon tasks with a few data.
- Proposing a computationally efficient environment adaptive replanner that revises misleading subgoals by visual cues, enabling the generation of plans grounded to the states in the environment without LLM.
- Achieving state-of-the-art performance in few-shot settings in the ALFRED benchmark (Shridhar et al. 2020) in all metrics.

2 Related Work

We first review the attempts to use an LLM in robotics, especially for task planning. Then, we discuss recent approaches to tackle complex instruction-following tasks.

Foundation models for task planning. With a recent development in large foundation models (*i.e.*, LLMs and VLMs) (Brown et al. 2020; Chen et al. 2021; Zhang et al. 2022; Liu et al. 2023), they are used as a tool for reasoning (Zeng et al. 2023; Singh et al. 2023; Driess et al. 2023), planning (Song et al. 2023; Sarch et al. 2023; Yang et al. 2024; Szot et al. 2024), and manipulation (Wu et al. 2023; Fang et al. 2024) in robot systems. Early approaches in robotic planning (Huang et al. 2022) using LLMs plan the subtasks by iterative enhancement of input prompts. For example, when the agent does not execute a planned action, (Huang et al. 2023) uses multiple environmental feedbacks to adjust the initial plan to recover its failure.

Similarly, (Ahn et al. 2022) enabled robot planning with skill affordance value functions for planning. To directly produce actionable robot policies, (Singh et al. 2023; Liang et al. 2023) structured a programmatic LLM prompt. Meanwhile, VIMA (Jiang et al. 2023) and PaLM-E (Driess et al. 2023) use multimodal prompts to control robots.

For the purpose of expanding to a variety of tasks, (Wang et al. 2024b) reveals that with well-crafted instructions, LLMs can effectively instruct a quadruped robot in locomotion tasks. Scaling to open-ended environments (Fan et al. 2022), agents (Wang et al. 2024a; Zheng et al. 2024) uses LLMs to build a continual learning agent.

While these methods make significant progress in robot planning by using LLMs, they depend on multiple interactions with the LLM to refine or adapt the robot's behavior, leading to heavy inference cost or network overhead if they are used as API calls. In contrast, our FLARE does not use LLMs for replanning to improve computational efficiency in adapting agents to the current environment.

Instruction following embodied agents. Embodied instruction following task requires an agent to generate a sequence of actions that align with natural language instructions within a given environment. Many prior arts (Singh et al. 2021; Pashevich et al. 2021; Kim et al. 2021) train an agent in an end-to-end manner, directly generating low-level actions from natural language instructions. Simultaneously, a templated approach has been proposed for planning a long-horizon tasks (Min et al. 2022; Inoue and Ohashi 2022). While it is data efficient, it is limited to solving the *predefined* tasks and does not generalize to novel tasks.

As the hierarchical or modular planning approach has been shown to be effective in instruction following tasks (Min et al. 2022; Blukis et al. 2021; Kim et al. 2023; Xu et al. 2024), an attempt to take advantage of LLMs as planners occurred. (Song et al. 2023; Sarch et al. 2023) use LLM as a high-level planner in such tasks by prompting LLM with few-shot in-context examples, which have been shown to be highly effective (Brown et al. 2020). Both methods retrieve several prompting examples based on the distances of the embedded language instructions.

3 Approach

Generating executable grounded plans is one of the key components in developing a successful embodied AI agent (Murray and Cakmak 2022; Inoue and Ohashi 2022; Kim et al. 2023). State-of-the-art methods (Kim et al. 2023; Min et al. 2022; Blukis et al. 2021; Pashevich et al. 2021) rely heavily on extensive data, implying that they would not be effective in data-scarce learning scenarios. However, given the high costs of annotating free-form language instructions, it is desirable to develop a more practical approach to learn an agent using small amounts of data. In addition to efforts to use LLMs as planners (Ahn et al. 2022; Huang et al. 2022, 2023; Sarch et al. 2023), (Song et al. 2023) uses them to learn an agent with a few examples.

However, LLMs do not always generate plausible plans without proper prompt, resulting in the generation of nonsensical or impractical subgoals. For example, for the task of 'Put a cooked potato in the fridge,' the LLM may tell an agent to 'wrap the potato with a foil,' where 'wrapping' is not supported by the agent. Although LLMs create executable plans quite successfully, the inherent ambiguity and lexical diversity of open-vocabulary descriptions often make the connection of language-based instructions to the physical world less clear. To be specific, an agent that has learned

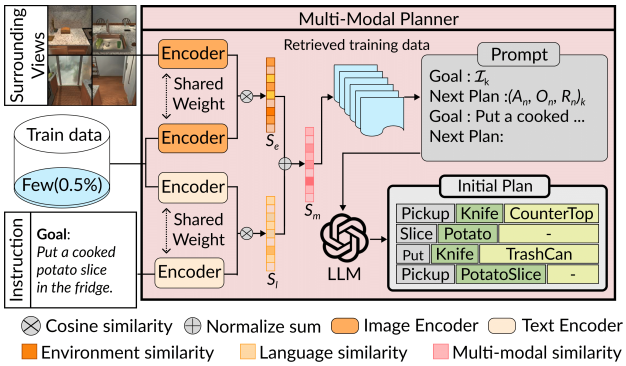

Figure 3: **Multi-Modal Planner.** MMP selects top k expert demonstrations using 'multi-modal similarity' (Eq. (1)) and converts them into subgoal triplets (A_n, O_n, R_n), which guide an LLM to generate task-specific subgoal sequences from natural language instructions.

a *sofa* would fail to recognize a *couch*. This may result in plans that are not well grounded in environments, leaving the agent unable to effectively cope with ungrounded plans when faced with real-world scenarios (*e.g.*, endlessly wandering for an object that is not presented in the scene). To address this issue, we propose FLARE that improves task planning for AI agents embodied by using visual and text inputs. Moreover, our approach uses visual observations from the agent, enabling visually adaptive grounded replanning.

Finally, our agent integrates two proposed components, 'Multi-Modal Planner' and 'Environment Adaptive Replanning.' Figure 2 illustrates the architecture of our FLARE.

3.1 Multi-Modal Planner

To generate interpretable subgoal sequences for an agent by natural language instructions, LLMs are widely used (Zeng et al. 2023; Singh et al. 2023; Driess et al. 2023; Wu et al. 2023; Sarch et al. 2023). For example, (Song et al. 2023; Sarch et al. 2023) retrieves in-context examples from the similarity of language instructions to prompt an LLM. Inspired by them, we propose 'Multi-Modal Planner (MMP)' that considers both the natural language instruction and the agent's egocentric surrounding views at the moment of receiving the command to reflect the environment status only with a few annotated data. We illustrate an MMP in Figure 3.

Multi-modal Similarity. In-context learning for LLM largely improves model performance for a wide spectrum of language tasks (Brown et al. 2020). It uses explicit context within the prompt to refine the model's comprehension and responsiveness to detailed language instructions. To capitalize on LLM as a few-shot learner, we need to carefully select examples that are relevant to the task at hand. When available, such relevant examples assist the LLM's ability to generate appropriate subgoals. For example, when the task is to 'clean a cloth,' it is strategically sound to prompt the model with examples themed in analogy such as 'cleaning a fork' or 'washing dishes' over unrelated tasks such as 'heating apples.' While (Song et al. 2023; Sarch et al. 2023) achieve this by measuring the distances of the embedded language instructions, they do not consider the environment state (*i.e.*,

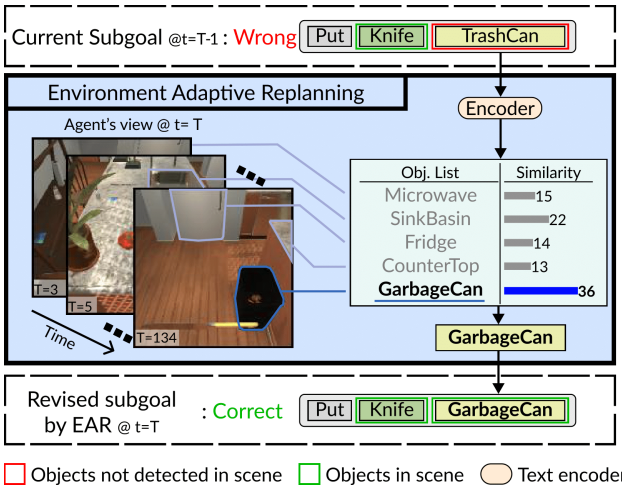

Figure 4: **Environment Adaptive Replanning.** EAR corrects a plan by listing detected objects and calculating semantic similarities to replace inaccurately referenced items. This ensures that the plan is grounded in the environment.

visual information) when generating a plan. This can result in inappropriate example selections for the current task (*e.g.*, cutting an apple with a knife when the knife is absent).

To take into account the state of the environment, we use the surrounding views of the agent when receiving a command. We then embed text instructions with a frozen BERT model (Devlin et al. 2018) and an image with a frozen CLIP-VIT encoder (Radford et al. 2021) to gauge how closely each training example aligns with the current task.

Formally, let $S_l = \{s_{l,1}, s_{l,2}, \ldots, s_{l,N}\}$ and $S_e = \{s_{e,1}, s_{e,2}, \ldots, s_{e,N}\}$ be language similarities and environment similarities, where $s_{l,i}$ and $s_{e,i}$ represent the language and environment similarity between the current task and the i^{th} example of the training set with the cosine similarities for each embedding vector, respectively. Then, we calculate the multi-modal similarities between the current task and the task of the training set (S_m) by taking the normalized sum of these individual similarity scores as:

$$S_m = w_l \cdot \frac{S_l}{\sum_{i=1}^{N} s_{l,i}} + w_e \cdot \frac{S_e}{\sum_{i=1}^{N} s_{e,i}}, \quad (1)$$

where w_l and w_e denote the weight of each instruction and environment similarity. Using the multi-modal similarity score, we retrieve the top k most relevant examples from the training data. These examples serve as in-context learning examples during the LLM's generation process, thus guiding the LLM to generate a more accurate subgoal.

Subgoal Representation. Translating language instructions into subgoals is one of the key components of robotic reasoning. *E.g.*, the task 'Move an apple from countertop to a dining table' can be decomposed into subgoals that include both navigation and object interaction, such as [Navigate, CounterTop], [Pickup, Apple], [Navigate, DiningTable], and [Put, DiningTable]. We propose an intermediate subgoal representation using a triplet of [Pickup, Apple, CounterTop] and [Put, Apple, DiningTable] for this representation

Algorithm 1: FLARE algorithm

Input: Time step t, Subgoal index k, Uncertainty u, Uncertainty threshold τ, Language instruction $\mathcal{I}$, Camera input $\mathcal{C}_t$, Subgoal sequences $\mathcal{P}$, Semantic map $\mathcal{S}$, Detected object set $\mathcal{V}$, Current object in interest O_k

$t, k, u \leftarrow 0$ ▷ Initialize
$\mathcal{P} \leftarrow \text{MMP}(\mathcal{I}, \mathcal{C}_t)$ ▷ Generate initial plan (Sec. 3.1)
$\mathcal{S} \leftarrow \text{SemanticMapping}(\mathcal{C}_t)$ ▷ Semantic map
$a_t \leftarrow \text{ActionPolicy}(\mathcal{P}_k, \mathcal{S})$ ▷ First action (Sec. 3.3)
while $k <$ length($\mathcal{P}$) **do**
  $\mathcal{C}_t \leftarrow \text{Execute}(a_t)$
  $\mathcal{S} \leftarrow \text{SemanticMapping}(\mathcal{C}_t)$ ▷ Update semantic map
  $\mathcal{V}$.add(ObjectDetector($\mathcal{C}_t$)) ▷ Update detected object set
  if O_k not in $\mathcal{O}$ **then**
    $u \leftarrow u + 1$
    if $u > \tau$ **then**
      $O_k \leftarrow \text{EAR}(O_k, \mathcal{V})$ ▷ Replanning (Eq. (3))
    end if
  else if Complete(P_k) **then**
    $k \leftarrow k + 1$ ▷ Update subgoal index
  end if
  $t \leftarrow t + 1$
  $a_t \leftarrow \text{ActionPolicy}(\mathcal{P}_k, \mathcal{S})$ ▷ Next action (Sec. 3.3)
end while

to reduce the total length of the instruction sets. Formally, for a given task instruction $\mathcal{I}$, we represent a subgoal as:

$$S_n = (A_n, O_n, R_n), \quad (2)$$

where $n \in \{1, ..., K\}$, K is the total number of subgoals in the sequence, A_n denotes high-level actions (*e.g.*, 'pick' or 'clean'), O_n denotes the target object of the action, and R_n denotes the receptacle where O_n is located. This approach reduces token usage by 25% compared to (Song et al. 2023).

3.2 Environment Adaptive Replanning

Despite the emergent ability of planning by large language models (LLMs), they may generate plans that are not well grounded in environments where agents are deployed. This issue can be attributed to the lexical variation inherent in natural language instructions. For example, consider the task, "Place a tray with a butter knife and slice of the fruit on the table." To complete the task, an agent needs to find the *fruit* to be sliced. However, if the agent has not learned the *fruit* object class for navigation during training, this may lead to navigation failure and possibly, task failure.

To address this issue, the proposed 'Environment Adaptive Replanning (EAR)' revises subgoals by replacing an undetected object with the most semantically similar object among those observed so far. To revise the subgoal, EAR first maintains a list of all detected objects that have been observed so far while completing the task. For each subgoal, if the agent cannot reach a navigation target (*i.e.*, O_n or R_n), EAR infers that the specified object is absent from the environment and replaces it with a semantically analogous one.

To replace a current unavailable object with another one, EAR finds the most *semantically similar* object among the candidates (*i.e.*, objects observed so far). To measure semantic similarity, we compute the cosine similarity of language representations of two object class names. Specifically, EAR

Setting	Model	Goal instructions + Sequential instructions				Goal instruction only			
		Test Seen		Test Unseen		Test Seen		Test Unseen	
		SR	GC	SR	GC	SR	GC	SR	GC
Few-shot (0.5%)	HLSM (Blukis et al. 2021)†	0.82 (N/A)	6.88 (N/A)	0.61 (N/A)	3.72 (N/A)	N/A	N/A	N/A	N/A
	FILM (Min et al. 2022)†	0.00 (N/A)	4.23 (N/A)	0.20 (N/A)	6.71 (N/A)	N/A	N/A	N/A	N/A
	CAPEAM (Kim et al. 2023)	0.00 (0.00)	3.90 (2.29)	0.20 (0.00)	6.63 (2.36)	N/A	N/A	N/A	N/A
	LLM-Planner (Song et al. 2023)	18.20 (N/A)	26.77 (N/A)	16.42 (N/A)	23.37 (N/A)	15.33 (N/A)	24.57 (N/A)	13.41 (N/A)	22.89 (N/A)
	FLARE-LLaMA2	16.96 (4.60)	24.84 (8.09)	17.79 (5.62)	27.40 (9.46)	12.00 (3.01)	20.05 (7.22)	13.73 (4.27)	21.98 (8.46)
	FLARE-Vicuna	20.61 (6.28)	29.57 (10.17)	22.04 (7.61)	33.57 (12.06)	16.37 (4.57)	23.68 (8.84)	18.05 (5.98)	26.75 (10.75)
	FLARE-GPT-3.5	32.55 (12.17)	42.02 (16.94)	31.79 (12.21)	43.94 (17.44)	23.48 (8.71)	33.40 (14.40)	25.38 (9.37)	36.02 (15.28)
	FLARE-GPT-4 (Ours)	**40.05** (**16.68**)	**48.84** (**21.31**)	**40.88** (**18.14**)	**51.72** (**22.78**)	**31.96** (**12.93**)	**41.36** (**18.55**)	**32.57** (**12.72**)	**43.23** (**18.40**)
Full	HLSM (Blukis et al. 2021)	29.94 (8.74)	41.21 (14.58)	20.27 (5.55)	30.31 (9.99)	25.11 (6.69)	35.79 (11.53)	16.29 (4.34)	27.24 (8.45)
	FILM (Min et al. 2022)	28.83 (11.27)	39.55 (15.59)	27.80 (11.32)	38.52 (15.13)	25.77 (10.39)	36.15 (14.17)	24.46 (9.67)	34.75 (13.13)
	CAPEAM (Kim et al. 2023)	51.79 (21.60)	60.50 (25.88)	46.11 (19.45)	57.33(24.06)	47.36 (19.03)	54.38 (23.78)	43.69 (17.64)	55.66 (22.76)

Table 1: **Comparison with state-of-the-art methods.** The path-length-weighteed (PLW) metrics are presented in the parentheses for each metric. †We excerpt 'SR' and 'GC' from (Song et al. 2023). For models without the PLW metric, we noted 'N/A.'

first obtains the language representations of the names of the current target object and observed objects using a pre-trained language model (Devlin et al. 2018; Raffel et al. 2020; Brown et al. 2020). Once obtained, EAR then computes the similarity scores of the observed objects with respect to the current target object.

Formally, we compute the similarity scores and obtain the most semantically similar object as following:

$$V^* = \arg\max_{V_i} \; S_C(\text{Enc}(O_k), \text{Enc}(V_i)), \tag{3}$$

where V^* is an object that maximizes $S_C(\cdot, \cdot)$, O_k is a current object, and V_i is a i^{th} detected objects so far. $\text{Enc}(\cdot)$ denotes a language encoder, and $S_C(\cdot, \cdot)$ denotes the cosine similarity of the two embeddings. Note that O_k can be either O_n or R_n. We illustrate the EAR in detail in Figure 4.

3.3 Action Policy

For object interaction, the agent first navigates to a target object and reaches it in a close vicinity. For navigation, a viable approach is to use imitation learning (Shridhar et al. 2020; Singh et al. 2021; Pashevich et al. 2021; Nguyen et al. 2021). However, it requires a large number of training episodes for acceptable performance, but collecting these episodes may not always be available, especially in our case where training data collection is often costly and time-consuming.

To avoid this issue, recent approaches (Inoue and Ohashi 2022; Kim et al. 2023) incorporate deterministic algorithms (*e.g.*, A* algorithm, FMM (Sethian 1996), *etc*.) obstacle-free path planning, leading to significant performance improvements compared to those learned by imitation learning. Inspired by recent observations, we adopt the deterministic approach (Sethian 1996) for effective path planning.

4 Experiments

4.1 Experimental Setup

We employ four large language models for our FLARE to validate the compatibility of the proposed methods with different models, incorporating both proprietary and open-source models. Specifically, we use GPT-4 and GPT-3.5 as proprietary models, and LLaMA2-13B (Touvron et al. 2023) and Vicuna-13B (Zheng et al. 2023) as open-source models. We select $k = 9$ in-context examples, following (Song et al. 2023) for a fair comparison with it and set w_l and w_e to the same values in equation (1) to treat each modality equally.

4.2 Dataset and Metrics

We evaluate the effectiveness of our FLARE in the ALFRED (Shridhar et al. 2020) benchmark. It requires agents to complete household tasks based on language instructions and egocentric observations within interactive 3D environments (Kolve et al. 2017). Both validation and test sets include *seen* and *unseen* scenarios, where the *seen* scenario is part of the training data, while the *unseen* scenario represents a new and unfamiliar environment for evaluation.

To evaluate the efficiency of FLARE where human language pairs are scarce, we followed the same few-shot setting(0.5%) as in the previous work (Song et al. 2023). For a fair comparison with the previous methods, we use the same number of examples (Song et al. 2023) (*i.e.*, 100 examples). The selected 100 examples contain all 7 task types for fair representations of 21,023 training examples.

For evaluation, we follow the same evaluation protocol as (Shridhar et al. 2020). The primary metric is a success rate (SR), measuring the percentage of completed tasks. A goal-condition success rate (GC) measures the percentage of satisfied goal conditions. Furthermore, we assess the efficiency of agents penalizing SR and GC (*i.e.*, PLWSR and PLWGC) with the path length of a trajectory taken by the agents.

4.3 Comparison with State of the Arts

We first compare our method with state-of-the-art methods (Blukis et al. 2021; Min et al. 2022; Kim et al. 2023; Song et al. 2023) and summarize the result in Table 1. Following (Min et al. 2022; Kim et al. 2023; Blukis et al. 2021), we report the performance of agents using 1) only a goal statement, denoted by 'Goal instruction only,' and 2) both goal statement and step-by-step instructions, denoted by 'Goal instructions+Sequential instructions.'

First, we observe significant performance drops from the full-shot setting to the few-shot setting from methods that require a large amount of data to train planners (HLSM, FILM, and CAPEAM). This implies that learning task-performing agents with limited training examples poses a significant challenge, as this data scarcity can hinder the learning of

LLM	Method	Seen Acc.	Unseen Acc.
LLaMA2	LLM-Planner (Static) (Song et al. 2023)	0.006	0.002
LLaMA2	**FLARE (w/o EAR)**	**18.54**	**22.29**
Vicuna	LLM-Planner (Static) (Song et al. 2023)	8.17	7.06
Vicuna	**FLARE (w/o EAR)**	**24.51**	**33.62**
GPT-3.5	LLM-Planner (Static) (Song et al. 2023)	29.78	31.67
GPT-3.5	**FLARE (w/o EAR)**	**46.10**	**55.66**
GPT-4	LLM-Planner (Static) (Song et al. 2023)	31.54	30.12
GPT-4	**FLARE (w/o EAR)**	**61.34**	**67.48**

Table 2: **Planner accuracy comparison.** 'Seen Acc.' and 'Unseen Acc.' denote accuracies in valid seen and unseen folds. To solely compare planner accuracy, we omit replanning noeted as 'Static' and 'w/o EAR.'

#	MMP	EAR	Test Seen		Test Unseen	
			SR	GC	SR	GC
(*a*)	✓	✓	32.55 (12.17)	42.02 (16.94)	31.79 (12.21)	43.94 (17.44)
(*b*)	✗	✓	30.20 (12.13)	41.26 (17.27)	30.35 (11.62)	42.40 (16.66)
(*c*)	✓	✗	30.79 (11.98)	40.20 (16.51)	30.28 (12.01)	42.48 (17.03)
(*d*)	✗	✗	28.05 (11.48)	38.64 (16.23)	28.58 (11.82)	39.92 (16.13)

Table 3: **Ablation study.** PLW metrics are presented in the parentheses for each metric. MMP and EAR each denotes 'Multi-Modal Planner' and 'Environment Adaptive Replanning,' respectively.

models with diverse tasks, objects, and environments, implying challenging generalization.

We then compare the results with very recent work using LLMs (Song et al. 2023) that learns tasks only with a few training examples. We explore both proprietary and open source language models, including comparative models of lower performance, as shown in (Zheng et al. 2023).

Despite with a relatively less capable language models (*i.e.*, LLaMA2 (Touvron et al. 2023)), our proposed agent still outperforms in all metrics in unseen environments in both 'Goal instructions + Sequential instructions' and 'Goal instruction only,' implying its effectiveness. Furthermore, using better language models such as GPT-4 can notably improve our agent by large margins up to 24.46% as expected.

Planner Accuracy Comparison. To investigate the performance of the initial planner that generates action sequences, denoted by *static* planning, we compare accuracy of our agent and recent LLM-based planning methods (Song et al. 2023; Ahn et al. 2022) by removing their respective replanning strategies and report the result in Table 2.

To isolate LLM's effect in planning, we validate methods using different LLMs. We observe that our agent equipped with MMP, denoted by 'FLARE (w/o EAR),' consistently outperforms prior work (Song et al. 2023) by noticeable margins in accuracy for both seen and unseen environments across the LLMs, implying that the improvement of our MMP is not attributed to a particular LLM choice.

4.4 Ablation Study

We conduct a quantitative ablation study to analyze components proposed in FLARE and summarize the result in Table 3. We choose GPT-3.5 over GPT-4 as the language model due to the latter's significantly higher token generation cost.

Figure 5: **Benefits of proposed multi-modal planner (MMP).** Without MMP, the agent misinterprets the task, simply placing a *SoapBar* in the *SinkBasin*. With MMP, the agent seems to understand the *cleaning* objective, generates a plausible plan, and completes the task successfully.

Without Multi-Modal Planner. First, we ablate the 'MMP' from our method and the agent considers unimodal similarity to retrieve in-context examples from the dataset, neglecting the environment state for planning. Without the proposed component, we select in-context examples based on instruction similarity. Since a prompt reflects a single modality, the agent may omit environmental cues and misinterpret task requirements, leading to performance drops in both seen and unseen splits, as shown in $(\#a)$ *vs.* $(\#b)$.

Without Environment Adaptive Replanning. We then ablate 'EAR' from our agent.Without EAR, an agent cannot handle language variation and often misinterprets natural language instruction, leading to an erroneous subgoal. We observe noticeable performance drops ($1.76\%p$, $1.51\%p$ in SR) in both seen and unseen splits, as shown in $(\#a)$ *vs.* $(\#c)$. This implies that LLMs often fail to generate grounded plans in the environment where the agent is deployed, causing the agent to wander in search of an object that may not be present, eventually leading to task failure.

Without both. Without any of the proposed components, the agent adheres to the initial plan, which may not correspond to the current task. As expected, our agent without both 'MMP' and 'EAR' achieves the lowest performance among the agents equipped with either or both $(\#d)$ *vs.* $(\#a, b, c)$. Furthermore, we observe that using both multi-modal planning and adaptive replanning of the environment improves performance compared to using only either of them $((\#d) \rightarrow \#(b, c))$ *vs.* $(\#d) \rightarrow \#(a))$, implying that both components are complementary to each other.

4.5 Qualitative Analysis

We analyze our method with several qualitative results and illustrate the result in Figure 5 and 6.

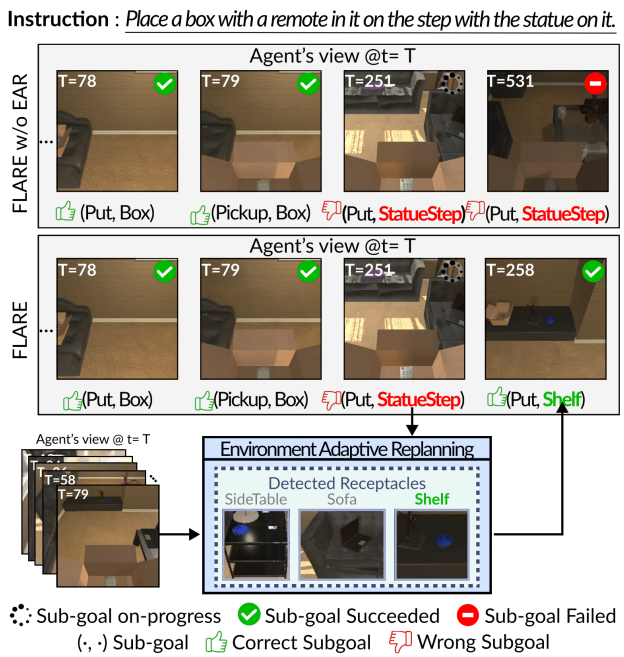

Figure 6: **Benefits of proposed environment adaptive replanning (EAR).** When the agent fails to find a specified object, EAR adapts the plan by selecting the most semantically similar detected replacement.

Multi-Modal Planner. To investigate the advantage of multi-modal planning, we present a qualitative example in Figure 5. As MMP retrieves relevant examples for the current task with multi-modal queries, it would encourage a large language model to generate more plausible subgoal.

We observe that the agent without MMP generates inappropriate subgoals, failing to understand the task context of the language instructions, which leads to placing a *SoapBar* on the *SinkBasin*. Subsequently, the agent fails to proceed with the generated subgoal sequence, as it cannot execute the *Put* action with an empty hand. In contrast, the agent equipped with MMP appears to succeed in extracting prior knowledge from the LLM. The agent generates satisfying subgoal sequences to pick up a *SoapBar*, clean it in a *SinkBasin*, and finally place it on the *CounterTop*.

Environment Adaptive Replanning. We then investigate the benefit of EAR. It adapts an unrounded plan with visual cues when the agent fails to locate the specified object.

Due to the various ways in referring to the object, an LLM confused by such diversity may create ungrounded subgoals. For example, Figure 6 shows a scenario where the agent is asked to place a *Box* on '*the step with the statue on it*.' An LLM that maximizes the given information generates *(Put, Box, StatueStep)* as a subgoal, causing the agent to wander around looking for a *StatueStep* while holding a *Box*.

We observe that the agent without EAR could not distinguish whether the current subgoal is inappropriate (*i.e.*, *StatueStep* does not exist), as it endlessly wanders around the scene and fails to specify receptacle object. On the contrary, an agent with EAR starts to search for a *StatueStep* initially and notices that *StatueStep* may not be present in the scene. After replacing the inappropriate object with the most relevant objects presented in the scene (*i.e.*, *StatueShelf* $\rightarrow$ *Shelf*), agent now executes the revised subgoal.

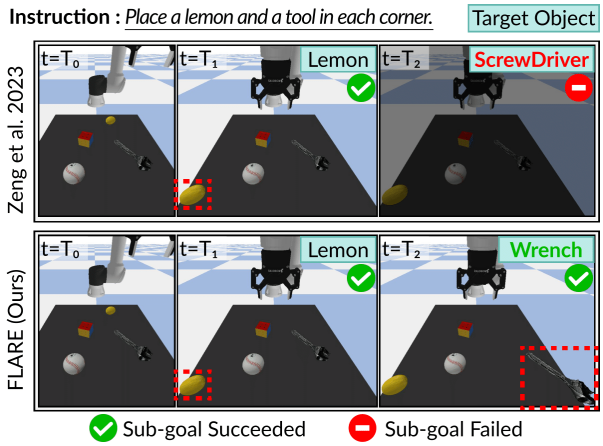

Figure 7: **An example of robotic task applications.** The baseline (Zeng et al. 2023) produces an ungrounded plan due to ambiguous instructions (*e.g.*, *tool*), while FLARE generates a grounded plan and executes actions successfully.

4.6 Application in Robotic Task Planning

We demonstrate the generalizability of the proposed FLARE to other robotic task applications. Specifically, we use a simulated tabletop environment with an UR5 robot arm and illustrate a comparison between FLARE and the baseline model in Figure 7. We choose (Zeng et al. 2023) as the baseline model for its effectiveness in few-shot robot planning. Both models use GPT-3.5 as an LLM to generate subgoals and employ a privileged low-level policy which uses the environment's metadata for end effector pose prediction.

We observe that FLARE successfully rearranges objects as instructed, demonstrating its capability in planning for grounded execution. In contrast, the baseline (Zeng et al. 2023) fails due to an ungrounded plan (*e.g.*, attempting to pick a *ScrewDriver* that is not present in the environment).

5 Conclusion

We propose FLARE with a multi-modal planner that reflects both environmental status by visual input and language instruction to generate detailed plans (*i.e.*, subgoals) to accomplish a long-horizon tasks with a few data. Additionally, it revises only the subset of the subgoals that are incorrect to generate physically grounded plans without using LLMs, leading to computationally efficient replanning. We empirically validate the effectiveness of the proposed components in ALFRED (Shridhar et al. 2020) and observe that our FLARE outperforms the state-of-the-art methods in few-shot settings by significant margins in all metrics.

Limitations and future work. Although our method requires a very few fraction of training data (0.5%), it still requires the training data. We aim to develop an agent that learns about environments through exploration, assisted by large language models, without needing any training data.

Acknowledgments

This work was partly supported by the IITP grants (No.RS-2022-II220077, No.RS-2022-II220113, No.RS-2022-II220959, No.RS-2022-II220871, No.RS-2021-II211343 (SNU AI), No.RS-2021-II212068 (AI Innov. Hub), No.RS-2022-II220951) funded by the Korea government(MSIT).

References

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

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*.

Bhambri, S.; Kim, B.; and Choi, J. 2023. Multi-level Compositional Reasoning for Interactive Instruction Following. In *AAAI*.

Blukis, V.; Paxton, C.; Fox, D.; Garg, A.; and Artzi, Y. 2021. A Persistent Spatial Semantic Representation for High-level Natural Language Instruction Execution. In *CoRL*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*.

Chaplot, D. S.; Sathyendra, K. M.; Pasumarthi, R. K.; Rajagopal, D.; and Salakhutdinov, R. 2017. Gated-attention architectures for task-oriented language grounding. In *AAAI*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. d. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv:2107.03374*.

Das, A.; Datta, S.; Gkioxari, G.; Lee, S.; Parikh, D.; and Batra, D. 2018. Embodied Question Answering. In *CVPR*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. 2023. PaLM-E: An Embodied Multimodal Language Model. In *ICML*.

Ehsani, K.; Gupta, T.; Hendrix, R.; Salvador, J.; Weihs, L.; Zeng, K.-H.; Singh, K. P.; Kim, Y.; Han, W.; Herrasti, A.; et al. 2024. SPOC: Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World. In *CVPR*.

Fan, L.; Wang, G.; Jiang, Y.; Mandlekar, A.; Yang, Y.; Zhu, H.; Tang, A.; Huang, D.-A.; Zhu, Y.; and Anandkumar, A. 2022. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. In *NeurIPS Datasets and Benchmarks Track*.

Fang, K.; Liu, F.; Abbeel, P.; and Levine, S. 2024. MOKA: Open-Vocabulary Robotic Manipulation through Mark-Based Visual Prompting. In *RSS*.

Ge, Y.; Tang, Y.; Xu, J.; Gokmen, C.; Li, C.; Ai, W.; Martinez, B. J.; Aydin, A.; Anvari, M.; Chakravarthy, A. K.; Yu, H.-X.; Wong, J.; Srivastava, S.; Lee, S.; Zha, S.; Itti, L.; Li, Y.; Martin-Martin, R.; Liu, M.; Zhang, P.; Zhang, R.; Fei-Fei, L.; and Wu, J. 2024. BEHAVIOR Vision Suite: Customizable Dataset Generation via Simulation. In *CVPR*.

Gordon, D.; Kembhavi, A.; Rastegari, M.; Redmon, J.; Fox, D.; and Farhadi, A. 2018. Iqa: Visual question answering in interactive environments. In *CVPR*.

Huang, W.; Abbeel, P.; Pathak, D.; and Mordatch, I. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *ICML*.

Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; Sermanet, P.; Jackson, T.; Brown, N.; Luu, L.; Levine, S.; Hausman, K.; and Ichter, B. 2023. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *CoRL*.

Inoue, Y.; and Ohashi, H. 2022. Prompter: Utilizing Large Language Model Prompting for a Data Efficient Embodied Instruction Following. *arXiv:2211.03267*.

Jiang, Y.; Gupta, A.; Zhang, Z.; Wang, G.; Dou, Y.; Chen, Y.; Fei-Fei, L.; Anandkumar, A.; Zhu, Y.; and Fan, L. 2023. VIMA: General Robot Manipulation with Multimodal Prompts. In *ICML*.

Kim, B.; Bhambri, S.; Singh, K. P.; Mottaghi, R.; and Choi, J. 2021. Agent with the Big Picture: Perceiving Surroundings for Interactive Instruction Following. In *Embodied AI Workshop @ CVPR*.

Kim, B.; Kim, J.; Kim, Y.; Min, C.; and Choi, J. 2023. Context-Aware Planning and Environment-Aware Memory for Instruction Following Embodied Agents. In *ICCV*.

Kim, T.; Min, C.; Kim, B.; Kim, J.; Jeung, W.; and Choi, J. 2024. ReALFRED: An Embodied Instruction Following Benchmark in Photo-Realistic Environment. In *ECCV*.

Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Gordon, D.; Zhu, Y.; Gupta, A.; and Farhadi, A. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv:1712.05474*.

Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; and Zeng, A. 2023. Code as policies: Language model programs for embodied control. In *ICRA*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In *NeurIPS*.

Majumdar, A.; Ajay, A.; Zhang, X.; Putta, P.; Yenamandra, S.; Henaff, M.; Silwal, S.; Mcvay, P.; Maksymets, O.; Arnaud, S.; et al. 2024. Openeqa: Embodied question answering in the era of foundation models. In *CVPR*.

Min, S. Y.; Chaplot, D. S.; Ravikumar, P.; Bisk, Y.; and Salakhutdinov, R. 2022. FILM: Following Instructions in Language with Modular Methods. In *ICLR*.

Murray, M.; and Cakmak, M. 2022. Following natural language instructions for household tasks with landmark guided search and reinforced pose adjustment. *RA-L*.

Nguyen, V.-Q.; Suganuma, M.; Okatani; and Takayuki. 2021. Look Wide and Interpret Twice: Improving Performance on Interactive Instruction-following Tasks. In *IJCAI*.

Pashevich, A.; Schmid, C.; Sun; and Chen. 2021. Episodic Transformer for Vision-and-Language Navigation. In *ICCV*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Ryu, H.; Kim, J.; Chang, J.; Ahn, H. S.; Seo, J.; Kim, T.; Choi, J.; and Horowitz, R. 2024. Diffusion-EDFs: Bi-equivariant Denoising Generative Modeling on SE(3) for Visual Robotic Manipulation. In *CVPR*.

Sarch, G.; Wu, Y.; Tarr, M.; and Fragkiadaki, K. 2023. Open-Ended Instructable Embodied Agents with Memory-Augmented Large Language Models. In *EMNLP*.

Sethian, J. A. 1996. A fast marching level set method for monotonically advancing fronts. In *PNAS*.

Shridhar, M.; Thomason, J.; Gordon, D.; Bisk, Y.; Han, W.; Mottaghi, R.; Zettlemoyer, L.; and Fox, D. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*.

Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; and Garg, A. 2023. Progprompt: Generating situated robot task plans using large language models. In *ICRA*.

Singh, K. P.; Bhambri, S.; Kim, B.; Mottaghi, R.; and Choi, J. 2021. Factorizing Perception and Policy for Interactive Instruction Following. In *ICCV*.

Song, C. H.; Kil, J.; Pan, T.-Y.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2022. One Step at a Time: Long-Horizon Vision-and-Language Navigation with Milestones. In *CVPR*.

Song, C. H.; Wu, J.; Washington, C.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2023. LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models. In *ICCV*.

Szot, A.; Schwarzer, M.; Agrawal, H.; Mazoure, B.; Metcalf, R.; Talbott, W.; Mackraz, N.; Hjelm, R. D.; and Toshev, A. T. 2024. Large Language Models as Generalizable Policies for Embodied Tasks. In *ICLR*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Uppal, S.; Agarwal, A.; Xiong, H.; Shaw, K.; and Pathak, D. 2024. SPIN: Simultaneous Perception, Interaction and Navigation. In *CVPR*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024a. Voyager: An Open-Ended Embodied Agent with Large Language Models. *TMLR*.

Wang, Y.-J.; Zhang, B.; Chen, J.; and Sreenath, K. 2024b. Prompt a robot to walk with large language models. In *CDC*.

Wu, J.; Antonova, R.; Kan, A.; Lepert, M.; Zeng, A.; Song, S.; Bohg, J.; Rusinkiewicz, S.; and Funkhouser, T. 2023. Tidybot: Personalized robot assistance with large language models. In *IROS*.

Xia, F.; Zamir, A. R.; He, Z.; Sax, A.; Malik, J.; and Savarese, S. 2018. Gibson env: Real-world perception for embodied agents. In *CVPR*.

Xu, X.; Luo, S.; Yang, Y.; Li, Y.-L.; and Lu, C. 2024. DISCO: Embodied Navigation and Interaction via Differentiable Scene Semantics and Dual-level Control. *arXiv:2407.14758*.

Yang, Y.; Zhou, T.; Li, K.; Tao, D.; Li, L.; Shen, L.; He, X.; Jiang, J.; and Shi, Y. 2024. Embodied multi-modal agent trained by an llm from a parallel textworld. In *CVPR*.

Zeng, A.; Attarian, M.; Ichter, B.; Choromanski, K.; Wong, A.; Welker, S.; Tombari, F.; Purohit, A.; Ryoo, M.; Sindhwani, V.; et al. 2023. Socratic models: Composing zero-shot multimodal reasoning with language. In *ICLR*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv:2205.01068*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS Datasets and Benchmarks Track*.

Zheng, S.; jiazheng liu; Feng, Y.; and Lu, Z. 2024. Steve-Eye: Equipping LLM-based Embodied Agents with Visual Perception in Open Worlds. In *ICLR*.

Zhu, Y.; Gordon, D.; Kolve, E.; Fox, D.; Fei-Fei, L.; Gupta, A.; Mottaghi, R.; and Farhadi, A. 2017. Visual semantic planning using deep successor representations. In *ICCV*.