Socratic Planner: Inquiry-Based Zero-Shot Planning for Embodied Instruction Following

Suyeon Shin[1], Sujin Jeon[1*], Junghyun Kim[1*], Gi-Cheon Kang[1*], and Byoung-Tak Zhang[1,2]

[1] Interdisciplinary Program in AI (IPAI), Seoul National University
[2] AI Institute, Seoul National University
{syshin,sjjeon,junghyunkim,gckang,btzhang}@bi.snu.ac.kr

Abstract. Embodied Instruction Following (EIF) is the task of executing natural language instructions by navigating and interacting with objects in 3D environments. One of the primary challenges in EIF is compositional task planning, which is often addressed with supervised or in-context learning with labeled data. To this end, we introduce the Socratic Planner, the first zero-shot planning method that infers without the need for any training data. Socratic Planner first decomposes the instructions into substructural information of the task through self-questioning and answering, translating it into a high-level plan, *i.e.,* a sequence of subgoals. Subgoals are executed sequentially, with our visually grounded re-planning mechanism adjusting plans dynamically through a dense visual feedback. We also introduce an evaluation metric of high-level plans, *Relaxed*HLP, for a more comprehensive evaluation. Experiments demonstrate the effectiveness of the Socratic Planner, achieving competitive performance on both zero-shot and few-shot task planning in the ALFRED benchmark, particularly excelling in tasks requiring higher-dimensional inference. Additionally, a precise adjustments in the plan were achieved by incorporating environmental visual information.

Keywords: Embodied Instruction Following · Zero-shot Learning · Socratic Reasoning · Visually Grounded Feedback

1 Introduction

With advancements in AI and robotics, there is a growing emphasis on the development of embodied agents capable of following natural language instructions. The Embodied Instruction Following (EIF) task [26] has emerged as a testbed for such agents, challenging them to effectively generate and execute a high-level plan (*i.e.*, sequence of subgoal), given a human instruction (*e.g., "Rinse off a mug and place it in the coffee maker"*) within interactive 3D environments. Existing approaches in EIF primarily rely on supervised learning [2, 8–10, 14, 17–19, 21, 26, 28, 29, 31, 32, 38], where embodied agents are trained on human-annotated data consisting of instructions and corresponding expert

[*] These authors contributed equally to this work

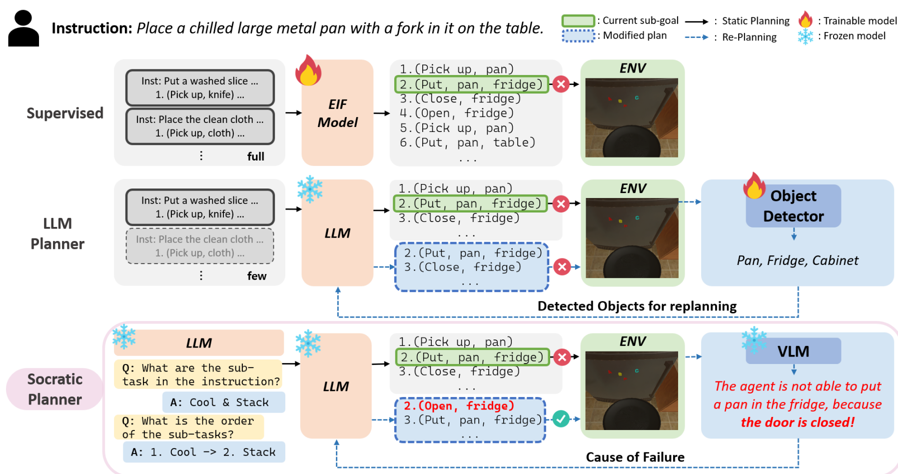

Fig. 1: Comparison between existing EIF methods and the Socratic Planner. The Socratic Planner enriches zero-shot task planning through self-questioning and answering about substructural information. Based on question-answering conversations, a Large Language Model (LLM) generates subgoals. While solid black arrows denote static planning paths, dashed blue arrows represent re-planning paths upon encountering execution failures, with visually grounded feedback from the Vision-Language Model (VLM) guiding LLM to modify the plan.

trajectories. However, they require large amounts of labeled data for training and often struggle to generalize to unseen instructions or environments. More recent research has explored leveraging the reasoning capabilities of Large Language Model (LLM) in task planning [1,7,15,16,27,30,36,37,39]. Most of these approaches plan out a sequence of subgoals by injecting in-context examples into LLM as prompts. While this paradigm significantly reduces the burden of collecting labeled data, the LLM-based planner for EIF [30] still necessitates hundreds of human-annotated data to take advantage of in-context learning.

In this paper, we ask: *how can embodied agents perform the task of EIF without using any labeled data?* Eliciting a Socratic reasoning [24] from LLM [20] and further grounding in the environments through dense visual feedback, we propose a method called, Socratic Planner. The Socratic Planner consists of three components: Socratic Task Decomposer (STD), Task Planner, and Vision-based Socratic Re-planner (VSR). Initially, the **Socratic Task Decomposer (STD)** uses an inquiry-based approach, leveraging LLM to decompose the instructions into substructural information (*e.g.*, desired sub-tasks, their order, target objects for each sub-task, how to execute each sub-task specifically) through self-questioning and answering. For example, in Fig. 1, users provide the instruction *"Place a chilled large metal pan with a fork in it on the table."* The LLM within STD engages in self-questioning and answering to decompose the task. Using question-and-answer pairs, the **Task Planner** synthesizes a sequence of subgoals through LLM, where each subgoal is either an action-object pair (e.g., (Pick up, pan)) or a triplet comprising an action, object, and receptacle (e.g., (Put, pan, fridge)). We integrate Socratic Planner with a low-level controller of the Hierarchical Language-conditioned Spatial Model (HLSM) [2] that executes these subgoals within the environment. If the agent encounters a failure while executing a subgoal sequentially, Vision-Language Model (VLM) [13] within **Vision-based Socratic Re-planning (VSR)** provides dense visual feedback to the Task Planner based on visual information from the environment. As illustrated in Fig. 1, if the agent encounters failure such as *"put a pan in the fridge"* in step 2, the VLM infers the cause of failure based on the current visual state of the environment. Upon inferring *"The agent is not able to put a pan in the fridge, because the door is closed"*, LLM successfully predicts a new subgoal *"open the fridge"* appropriate for the current state.

We propose a new evaluation metric to measure how well the high-level plan was predicted. Specifically, the ALFRED dataset [26] has only one human annotated ground-truth high-level plan per sample, so the studies in ALFRED evaluate the accuracy of the generated plan based on the exact match to the ground-truth plan. However, there can be semantically equivalent plans producing the same execution result as the ground-truth plan. Accordingly, we introduce a relaxed evaluation metric for high-level planning, *Relaxed*HLP, that extends the existing strict high-level planning metric, *Strict*HLP, by considering multiple possible correct plans. Through human evaluation, we validated the effectiveness of the *Relaxed*HLP metric in measuring the accuracy of high-level plans.

To demonstrate the effectiveness of the Socratic Planner, we conducted extensive experiments in the ALFRED dataset [26], which offers a diverse range of complex environments and tasks for EIF. Our method achieves competitive results with the existing state-of-the-art few-shot method [30] in all metrics even in zero-shot setting. Especially in tasks with long and complex subgoal sequences, it demonstrated substantial performance improvements, thereby validating the high-dimensional zero-shot reasoning capabilities of the Socratic Planner. Moreover, given an equivalent number of samples as a few-shot example, our proposed method consistently outperformed the existing method [30] across all metrics and scenarios. Through ablation studies and qualitative evaluations, we further validated the effectiveness of our proposed inquiry-based Socratic Task Decomposition method and Vision-based Socratic Re-planning.

In summary, our contributions are three-fold:

1. We propose the Socratic Planner, which maximizes the LLM's zero-shot task planning capability through inquiry-based Socratic Task Decomposition (STD) and Vision-based Socratic Re-planning (VSR).

We introduce a novel high-level planning metric, *Relaxed*HLP, designed to evaluate subgoal planning abilities more inclusively, enabling a more accurate assessment. Additionally, we demonstrate that *Relaxed*HLP closely aligns with human evaluation compared to existing HLP metrics.

3. Experimental results show that our approach effectively guides high dimensional reasoning abilities for EIF without providing any samples, leading to

planning performance that closely matches or exceeds that of the state-of-the-art few-shot method.

2 Related work

2.1 Embodied Instruction Following (EIF)

EIF is the task of executing natural language instructions, where embodied agents should navigate and interact with household objects in 3D environments [4, 26]. Prevailing approaches in EIF [9, 19, 21, 26, 29, 31–33, 38] rely on supervised learning, where embodied agents learn to map human instructions to sequences of actions using human-annotated data. However, these approaches necessitate large amounts of labeled data for training and often fail to generalize to novel environments. Meanwhile, recent studies leverage the strong reasoning capabilities of Large Language Model (LLM) to act as high-level planners for embodied AI agents [6, 7, 15, 16, 30]. Some studies have utilized LLM as auxiliary helpers [15], inserting insufficient plans into existing task guidance to achieve the final goal. Most research [6,7,16,30] generates high-level plans by injecting in-context examples as text prompts to LLM. Our approach belongs to this paradigm. However, instead of injecting in-context samples into LLM, Socratic Planner addresses EIF via zero-shot prompting without using any labeled data.

2.2 Multi-Step Reasoning with Large Language Model (LLM)

Our work is closely related to the large body of research on multi-step reasoning that uses LLM to decompose complex problems into intermediate steps. Chain-of-thought (CoT) prompting [35] is one of the most representative methods that prompts LLM to explicitly generate intermediate reasoning steps, thereby enhancing their reasoning capabilities. Some studies [5, 34] employ multiple path decoding and majority voting to select the most relevant output. Notably, a line of research [22, 25, 40] has focused on decomposing a complex problem into more manageable sub-problems to enhance the problem-solving capabilities of LLM. Especially, these methods employ a socratic approach, explicitly inferring answers to intermediate questions, given few-shot samples containing subquestion-solution pairs. Our approach shares the same spirit. However, Embodied Instruction Following (EIF) is inherently complex, covering diverse and long-horizon tasks. Therefore, manually annotating question-answer pairs for in-context learning is laborious and time-consuming. To effectively tackle EIF tasks, we instead define a list of things that need to be discovered, which could be generally applicable in EIF tasks. This enables LLM to autonomously generate and answer questions without in-context examples.

2.3 Re-planning

While executing generated plans in the environments, the agent may encounter failures due to various factors like navigation uncertainty or unexpected environmental states. Hence, iteratively adjusting the plan based on the evolving environmental conditions becomes crucial. However, most EIF studies [1, 2, 6, 8–10, 14, 15, 17, 18, 21, 26, 29, 31, 32] did not consider re-planning mechanisms. A work [38] proposed a limited re-planning method based on simple heuristics. If the embodied agent fails to achieve a subgoal, it backtracks to the previous step and retires the action. In other words, this method could not recover plans from failures caused by incorrect predictions of actions or target objects. Meanwhile, some research [3,7,23,27,30] explores methods where agents directly modify plans through online feedback upon task failure. Some studies [3, 7, 27] provide feedback to LLM in the form of human or simulation states utilizing error messages from simulator. However, applying such approaches in real-world settings where error messages may not be defined or human feedback is unavailable is challenging. Another approach, the re-prompting method [23] defines precondition errors and re-prompts to LLM to generate corrective actions using a template-based prompt strategy. This method lacks visual information, limiting its ability to identify unpredictable failure causes using only LLM's commonsense. In response, the LLM-Planner [30] was developed to update plans based on observed object lists. Yet, as illustrated in Fig. 1, LLM-Planner often fails in re-planning due to its reliance on only a set of visible objects, which does not sufficiently explain the current visual state. Therefore, we propose Vision-based Socratic Re-planning (VSR), which includes a visually grounded feedback advisor, leveraging a pre-trained Vision-Language model [13]. Based on a more detailed visual information (*e.g.*, visual depiction of the scene, subgoal feasibility, and failure causes within the current scene), VSR gives a dense visual feedback to the Task Planner, without directly relying on human or simulation data. This maximizes the LLM's reasoning capabilities for effective plan adjustment.

3 Method

3.1 Task Definition

Embodied Instruction Following (EIF) is to execute a sequence of actions implied in a natural language instruction I in interactive 3D environments. This task is divided into two stages: high-level planning and low-level control. First, high-level planning is to predict a sequence of subgoals. For instance, a high-level plan for *"put a heated slice of bread in the fridge"* can be translated into a series of subgoals; [(pickup, knife), (slice, bread), ..., (close, fridge)]. Each subgoal consists of (1) an admissible action (*e.g.*, Pickup, Put, ToggleOn, ToggleOff, Open, Close, Slice), (2) a target object, and optionally, (3) a receptacle (*i.e.*, target place). Next, low-level controller translates each subgoal into primitive navigation actions interacting with environments. Our focus lies in high-level planning, which necessitates complex, multi-step reasoning. To conduct the task of EIF, we integrate the Socratic Planner with the off-the-shelf low-level controller [2].

Algorithm 1 Socratic Planner

Require: $I \leftarrow$ Human Instruction
Require: Low-Level-Controller [2], LLM [20], VLM [13], Object-Detector [2]
Require: Prompt generators: φ_* for LLM and ψ_* for VLM
Require: Set of Observed Objects, $\mathcal{O} \leftarrow \emptyset$

Define subgoal $\triangleq (\text{action}_{subgoal}, \text{object}_{subgoal}, \text{receptacle}_{subgoal})$
Define $\mathcal{P}$ as list of subgoals
Define *scene* $\leftarrow$ a RGB image of current camera output
$\mathcal{QA} \leftarrow \text{LLM}(\varphi_{STD}(I))$ // Sec. 3.3
$\mathcal{P} \leftarrow \text{LLM}(\varphi_{HLP}(I, \mathcal{QA}))$ // Sec. 3.4
for subgoal **in** $\mathcal{P}$ **do**
  execute Low-Level-Controller(subgoal), updating *scene* // Sec. 3.5
  $\mathcal{O} \leftarrow \mathcal{O} \cup$ Object-Detector(*scene*)
  if execution *fails* **then**
    $c \leftarrow \text{VLM}(\psi_{caption}(\cdot), scene)$
    $v \leftarrow \text{VLM}(\psi_{validity}(\text{subgoal}, c), scene)$
    if $\text{object}_{subgoal}$ not in $\mathcal{O}$ **or** v is *invalid* **then**
      $f \leftarrow \text{VLM}(\psi_{feedback}(\text{subgoal}, c, v), scene)$
      $\mathcal{P} \leftarrow \text{LLM}(\varphi_{VSR}(f, \mathcal{P}, \mathcal{O}, c))$
    else
      redo // Redo the current subgoal
    end if
  end if
end for

3.2 Overview

The research community has observed the robust zero-shot reasoning abilities of Large Language Models (LLMs) [20] across various tasks. However, leveraging LLMs for zero-shot inference of complex tasks like EIF often falls short, necessitating high-dimensional reasoning. To address this, we introduce the Socratic Planner, a novel zero-shot high-level planning method comprised of three components. The Socratic Task Decomposer (STD, Sec. 3.3) first decomposes tasks into substructural information through self-questioning and answering. Based on the information, LLM synthesizes a high-level plan (*i.e.,* subgoal sequence) in Sec. 3.4. Next, the agent executes these subgoals sequentially by employing the low-level controller [2]. In cases of execution failure, the agent iteratively modifies the existing plan to better suit the environment through dense visual feedback, a process referred to as Vision-based Socratic Re-planning (Sec. 3.5). The LLM used across all sections is GPT-Turbo-3.5 [20].

3.3 Socratic Task Decomposer

Since natural language instructions in EIF are inherently compositional, it is natural to break down the instruction into more fine-grained elements. Inspired by Socrates's Socratic method [24], a method used to stimulate critical thinking through a series of questions and answers – we empower LLMs to decompose

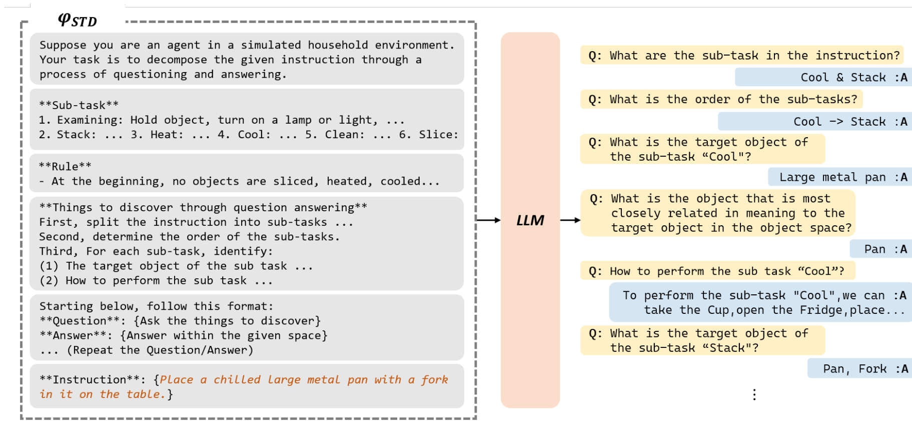

Fig. 2: Input prompt and output of the Socratic Task Decomposer (STD).

complex task instructions into substructural information by engaging in self-questioning and answering. We refer to this process as the Socratic Task Decomposer (STD), which enables the high-dimensional inference required in EIF. Before describing the details of STD, the "sub-task" refers to the intermediate-level task. For example, when faced with the instruction *"put a heated slice of bread in the fridge"* in EIF, the sub-tasks comprising this entire task could involve *"heating the bread"*, *"slicing bread"*, and *"placing bread in the fridge"*, while each sub-tasks consists of multiple subgoals.

The depth and content of substructural information in a task vary depending on its complexity. To aid Large Language Models (LLMs) in accurately reasoning about this information for general Embodied Instruction Following (EIF) tasks, we provide them with guidance in the form of "things to discover". These directives prompt LLMs to consider essential aspects such as identifying constituent sub-tasks, their sequencing, target objects, and specific methods for task execution. As depicted in Fig. 2, the prompt for STD comprises the aforementioned "things to discover", in addition to the mission of the LLM, concise sub-task explanations, and 3D simulator rules. The LLM finally generates question-and-answer sequences ($\mathcal{QA}$) that discover the substructural information based on the prompt.

3.4 High-Level Plan Generation

Utilizing the $\mathcal{QA}$ generated in Sec. 3.3, the Task Planner generates a high-level plan based on the LLM. The prompt generator for High-Level Plan Generation, denoted as φ_{HLP}, takes I and $\mathcal{QA}$ as input to produce a prompt. φ_{HLP} is augmented with sub-task information and simulator rules, along with the command to create a high-level plan (*"Based on this conversation, create a detailed plan for executing instructions that consist of various sub-tasks"*). Subsequently, this prompt is fed into the LLM to generate the high-level plan,

$\mathcal{P} \leftarrow \text{LLM}(\varphi_{HLP}(I, \mathcal{QA}))$. This plan is composed of a sequence of subgoals where each subgoal is a combination of primitive actions, interacting objects, and optionally, receptacles.

3.5 Vision-based Socratic Re-planning

Vision-based Socratic Re-planning (VSR) enables agents to recover from execution failures by adjusting the existing plan generated by the Socratic planner, through dense visual feedback. Starting from the 6^{th} line of Algorithm 1, the high-level plan generated in Sec. 3.4 is executed by the low-level controller [2], with each subgoal being performed sequentially. As execution proceeds, the agent continuously updates its observed scene, and detected objects by the low-level controller's object detector are added to the set of observed objects. When execution encounters failures, the agent seeks guidance from the Vision-Language Model (VLM) [13]. The re-planning process starts with the agent querying the VLM for the caption c of the current environment scene, and whether the subgoal seems valid (*i.e.,* executable). Utilizing the accumulated information, the agent determines whether the re-planning is required. Suppose the current subgoal's target object is visible, and the action seems valid. In that case, the agent assumes that the failure results from a malfunction of the low-level controller and retries the subgoal again. Otherwise, the VLM generates feedback (*i.e.,* a cause of the failure), denoted as f, based on the cues obtained from the VLM itself. Leveraging this dense visual feedback, the Task Planner finally reasons about revising the plan and adjusts it to be better suited to the environment. Once the plan is modified, execution resumes from the revised subgoal, with the low-level controller proceeding with the action sequence. This iterative process, called Vision-based Socratic Re-planning (VSR), makes the LLM's reasoning more grounded to the environments. Detailed information about the prompt generator for VLN and LLM will be discussed in Appendix 1.

4 Experiment

4.1 Dataset

We validate Embodied Instruction Following (EIF) capabilities of Socratic Planner using the ALFRED benchmark [26]. This benchmark evaluates a mapping from natural language instructions and egocentric vision to sequences of actions for long-horizon household tasks. The ALFRED dataset consists of 25k language instructions describing 8k expert demonstrations, with each demonstration corresponding to a high-level plan. The dataset includes seven admissible actions, 108 distinct objects, and seven types of tasks in 120 scenes. The data is divided into training, validation, and test splits, where the validation and test datasets are further classified into "seen" and "unseen" subsets. The former contains samples with seen environments of the training dataset, while the latter contains samples with unseen environments.

Table 1: Comparison with the state-of-the-art methods on SR and GC across four splits. Static models are without the visually grounded re-planning. The results marked with * indicate reproduced results of LLM-Planner [30]. Bold symbols in numbers denote the highest accuracy, while underlined symbols indicate the runner-up for each experiment setting.

Model		Valid Seen		Valid Unseen		Test Seen		Test Unseen	
	n-shot	SR	GC	SR	GC	SR	GC	SR	GC
HiTUT [38]	full	<u>18.41</u>	<u>25.27</u>	<u>10.23</u>	<u>20.71</u>	13.63	21.11	11.21	17.89
HLSM [2]	full	**29.63**	**38.74**	**18.28**	**31.24**	25.11	35.79	20.27	27.24
FILM [17]	full	-	-	-	-	25.77	36.15	24.46	34.75
LGS-RPA [18]	full	-	-	-	-	<u>33.01</u>	<u>41.71</u>	<u>27.80</u>	<u>38.55</u>
CAPEAM [10]	full	-	-	-	-	**47.36**	**54.38**	**43.69**	**54.66**
LLM-Planner (Static) [30]	few	11.82	23.54	11.10	22.44	13.05	20.58	<u>11.58</u>	18.47
LLM-Planner [30]	few	<u>13.53</u>	**28.28**	<u>12.92</u>	**25.35**	**15.33**	**24.57**	**13.41**	**22.89**
Socratic-Planner (Static)	few	13.29	24.09	12.79	24.20	11.74	19.95	10.14	18.48
Socratic-Planner	few	**14.88**	<u>25.47</u>	**13.40**	<u>24.91</u>	<u>13.24</u>	<u>21.51</u>	10.66	<u>19.53</u>
LLM-Planner* (Static) [30]	zero	5.73	11.24	4.39	12.97	5.71	10.42	6.54	12.12
Socratic-Planner (Static)	zero	<u>10.12</u>	<u>20.44</u>	<u>8.40</u>	<u>21.18</u>	<u>9.85</u>	<u>18.66</u>	<u>8.96</u>	<u>17.86</u>
Socratic-Planner	zero	**11.10**	**20.77**	**10.23**	**22.50**	**12.26**	**20.86**	**9.94**	**18.88**

4.2 Evaluation Metric

We use standard evaluation metrics in the ALFRED benchmark, Success Rate (SR) and Goal Conditioned (GC) success rate of tasks. SR represents the percentage of tasks wherein the state changes driven by the agent completely satisfy predefined goal-conditions. *i.e.*, a task is considered a success only when all subgoals are achieved. On the other hand, GC measures the percentage of completed goal-conditions for each task.

We also evaluate the accuracy of high-level planning (HLP). In the study [30], HLP accuracy was measured solely based on the exact match with the annotation of a High-Level Plan from ALFRED [26], a metric we refer to as *Strict*HLP. However, tasks often have multiple viable subgoal sequences to achieve completion. For instance, in the dish-washing task, both high-level plans are correct: turning on the faucet after placing the dish in the sink, and vice versa. Thus, we introduce *Relaxed*HLP, a novel HLP metric aimed at offering a more comprehensive assessment by considering a broader range of valid plans. We validate the effectiveness of *Relaxed*HLP in Tab. 4, and the detailed explanation for *Relaxed*HLP is provided in Appendix 2.1.

4.3 Comparison with SOTA

Compared Models. The compared models are categorized into three groups, each with its distinct training setup: fully-supervised setting [2,10,17,18,38], few-shot setting [30], and zero-shot setting. In the fully-supervised setting, models are trained using the entire training dataset. For the few-shot setting, the model

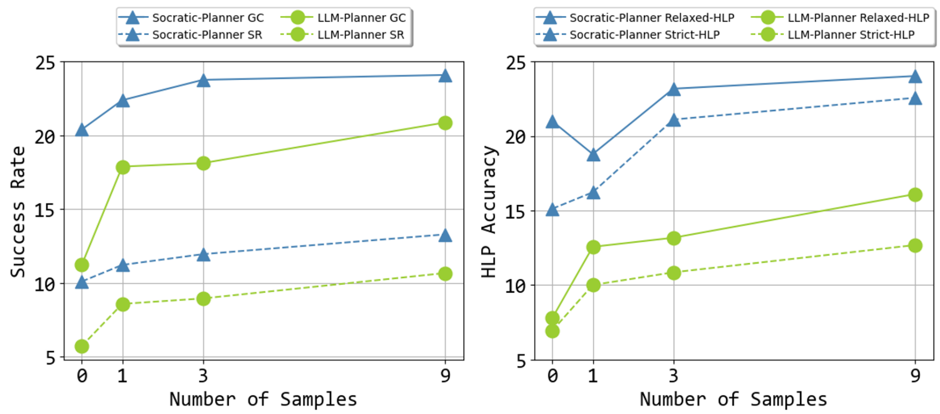

Fig. 3: Comparison with LLM-Planner on zero-shot and few-shot settings in EIF metrics (left) and HLP metrics (right), where valid seen splits are used for evaluation.

engages in in-context learning by prompting a few samples extracted from the training data. Finally, the zero-shot setting entails models inferring high-level plans without utilizing any training data. Since Socratic Planner is the first zero-shot approach, we regard the state-of-the-art few-shot method (*i.e.,* LLM-Planner [30]) as the most comparable model. To ensure maximal comparability between the Socratic Planner and LLM-Planner, we further identify the results from two models: (1) LLM-Planner under zero-shot setting and (2) Socratic Planner under few-shot setting.

Results on Embodied Instruction Following (EIF). As shown in Tab. 1, Socratic Planner significantly outperforms LLM-Planner [30] in the zero-shot setting. We also observe that the use of vision-based Socratic re-planning (VSR) consistently boosts SR and GC compared with the Socratic Planner without VSR (*i.e.,* Static). In the few-shot setting, the Socratic Planner, taking the same in-context examples as LLM-Planner (*i.e.,* nine-shot), shows improved performance on SR. It indicates that our proposed method is effective in both zero-shot and few-shot settings. Moreover, we conduct a more detailed analysis by adjusting the number of in-context examples (0, 1, 3, and 9). As depicted on the left side of Fig. 3, Socratic Planner outperforms LLM Planner on both SR and GC.

Results on High-Level Planning (HLP). We also identify the accuracy of high-level planning. As illustrated on the right side of Fig. 3, we visualize the performance of *Strict*HLP and *Relaxed*HLP in n-shot settings, $n \in \{0, 1, 3, 9\}$. Similar to the results on Embodied Instruction Following, the Socratic Planner significantly improves the HLP performance across all settings compared with LLM-Planner. Furthermore, we conduct a task-type analysis to identify the accuracy of high-level planning (*i.e.,* *Relaxed*HLP) for each task in the ALFRED benchmark. In this analysis, we compare the Socratic Planner with zero-shot reasoning with the original LLM-Planner with few-shot reasoning. In Tab. 2,

Task Type	Task Length	LLM Planner*	Socratic Planner
Heat	12.8	3.74	**16.82** (+3.50)
Cool	10.3	3.17	**21.43** (+5.76)
Clean	7.2	12.50	**16.96** (+0.37)
Pick Two	5.8	**25.00**	5.65 (−0.77)
Stack	5.7	**6.96**	4.35 (−0.38)
Pick	2.5	23.24	**36.62** (+0.58)
Examine	2.1	40.43	**46.81** (+0.16)

Table 2: *Relaxed*HLP accuracy for LLM-Planner and Socratic Planner by task type in the static planning setting. The results marked with * indicate reproduced results from LLM-Planner [30]. The numbers in brackets represent the relative difference of the Socratic Planner compared to the LLM Planner.

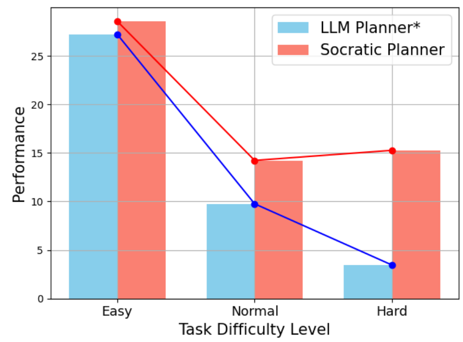

Fig. 4: Subtask-wise results. Dividing tasks into three levels of difficulty based on task length, and comparing the *Relaxed*HLP accuracy of LLM-Planner and Socratic Planner based on the difficulty level. The results marked with * indicate reproduced results from LLM-Planner [30].

Table 3: Ablation of Socratic Planner's STD in the static setting, where valid seen and valid unseen splits are used for evaluation.

Model	Valid Seen				Valid Unseen			
	*Relaxed*HLP	*Strict*HLP	SR	GC	*Relaxed*HLP	*Strict*HLP	SR	GC
Socratic-Planner w.o. STD	13.78	12.32	7.07	15.98	16.93	12.79	5.40	17.20
Socratic-Planner w. CoT [11]	13.66	9.88	7.20	17.07	16.44	11.45	6.33	18.25
Socratic-Planner	**20.98**	**15.12**	**10.10**	**20.40**	**23.02**	**14.01**	**8.40**	**21.37**

the Socratic Planner shows improved high-level planning scores in five out of seven tasks. It is also noteworthy that the Socratic Planner yields strong HLP performance, especially in long-horizon tasks that have relatively long subgoal sequences. Meanwhile, the performance of the Socratic Planner appears to be relatively lower in the "Pick Two" and "Stack" tasks. Both tasks entail picking up two objects and relocating them. However, the AI2-THOR [12] simulator for the ALFRED benchmark does not support picking up two objects simultaneously. We believe that this unique characteristic is challenging, especially for the zero-shot methods that do not observe any prior data.

To identify how HLP performance varies based on the task length, we reorganize the valid-seen data into three categories: the length of five or less (0-5) as easy, 5-10 as normal, and over 10 as hard. As shown in Fig. 4, the Socratic Planner shows better performance in all categories. Surprisingly, the performance gap between the Socratic Planner and LLM-Planner widens as the level of task difficulty increases. This showcases that the reasoning capabilities of the Socratic Planner are relatively robust on challenging EIF tasks.

Table 4: Comparison of alignment between HLP metrics and human evaluation. For evaluation, 70 samples were randomly selected from the valid seen split, with 10 samples per task type.

	Accuracy	Precision	Recall	F1-Score
*Relaxed*HLP	86.96	83.33	71.43	76.92
*Strict*HLP [26]	81.16	78.57	52.38	62.86

4.4 Ablation Study

To demonstrate the effectiveness of components in the Socratic Planner, we conducted an ablation study. In Tab. 3, we can clearly identify the merits of the Socratic Task Decomposer (STD) across all evaluation metrics. It indicates that discovering substructural information – represented as question-and-answer pairs helpful for synthesizing a sequence of subgoals – is crucial for EIF. However, a question still remains as to why the information should be a form of question-and-answer. Therefore, we compare the Socratic Planner with one of the most representative multi-step reasoning methods for LLMs, Chain-of-though (CoT) prompting [11]. Specifically, we reproduce zero-shot CoT [35] by replacing the description about questioning and answering in the prompts $\varphi_{STD}(I)$ with the sentence, "How can you decompose the instruction: {instruction}? Let's think step by step." In Tab. 3, the zero-shot CoT struggles to yield useful information. In our manual inspection, this method extracts information in a haphazard manner, which ultimately hindered the ability to handle the complex reasoning required for EIF tasks.

4.5 Analysis of *Relaxed*HLP

To validate the effectiveness of *Relaxed*HLP, we study how well the results from *Relaxed*HLP are aligned with human evaluation of high-level planning. A set of 70 samples was randomly extracted from the plans generated by the Socratic Planner, and these were evaluated by *Relaxed*HLP, StricHLP, and human evaluators. Note that the result of each sample is either 0 (incorrect) or 1 (correct). We measure the agreement between the results from each metric and the results from human evaluation and finally report four types of scores; Accuracy, Precision, Recall, and F1-score.

As shown in Tab. 4, *Relaxed*HLP shows better scores on all four metrics. Especially, the higher Recall score of *Relaxed*HLP indicates that our proposed metric encompasses more samples considered correct by humans. Additionally, the Precision score of *Relaxed*HLP is also improved compared with *Strict*HLP, resulting in higher F1-Score. In summary, the results indicate that while the ground truth annotated by humans typically represents only one sequence for performing the task, *Relaxed*HLP can cover a broader range of high-level plans that make the task successful, even if they do not perfectly align with the ground truth plan.

4.6 Qualitative Results

To visualize the effects of the STD Sec. 3.3, We present the qualitative examples of the generated action sequence in Fig. 5 from Socratic Planner and Socratic Planner without STD. In this case, the Socratic Planner without STD struggles to complete the instruction *"Slice bread and chill it in the fridge"* due to the lack of structured information regarding the required sub-tasks, their sequence, and the objects needed to complete the instructions. In this scenario, the agent attempts to slice bread without first picking up a knife, which is necessary to slice the bread. On the other hand, Socratic Planner effectively decompose this task into two sub-tasks: *"slice the bread '* and *"chill the bread"* (Q1 to A3). It then clearly structures the sequence of sub-tasks and the necessary objects for each sub-task through dialogue (Q3 to A9). Based on this structured information, it accurately predicts and successfully executes the lengthy sequence of subgoals.

Fig. 6 shows the Vision-based Socratic Re-planning procedure. In this scenario, the agent needs to perform the instruction *"turn on the desk lamp."* The plan initially generated by the static Socratic Planner involves picking up the desk lamp to turn on the light. However, in the simulation, it's not possible to pick up the desk lamp, resulting in a failed action. Using the scene at the point of failure, the VLM infers whether there is subgoal's target object in the scene (Q2 and A2) and whether the action of the failed subgoal (Pickup, Desklamp) is feasible (Q3 and A3). Based on this information, it decides whether to re-plan or the same subgoal needs to be attempted again due to uncertainty from the low-level planner. If a re-plan is necessary, the VLM explicitly infers the cause of failure, such as *"The desk lamp is too heavy"*, based on vision information (Q4 and A4). This inference is provided as feedback, allowing LLM to adjust its plan accordingly. Using this dense visual feedback, LLM infer how to modify the plan (Q6 and A6), and subsequently adjusts its initial plan successfully to directly turn on the light instead of attempting to pick up the desk lamp.

5 Conclusion

This paper explores an effective zero-shot reasoning method for Embodied Instruction Following (EIF), the Socratic Planner. Leveraging Socratic inquiry-based task decomposition and visually grounded re-planning strategy, the Socratic Planner shows robust and competitive performance on high-level planning and EIF. Furthermore, we introduce a novel evaluation metric, *Relaxed*HLP, to perform a more comprehensive evaluation of high-level planning. We hope these contributions mark a significant advancement in enabling more effective task planning for embodied agents, showing the potential for following instructions in complex environments without relying on extensive labeled data.

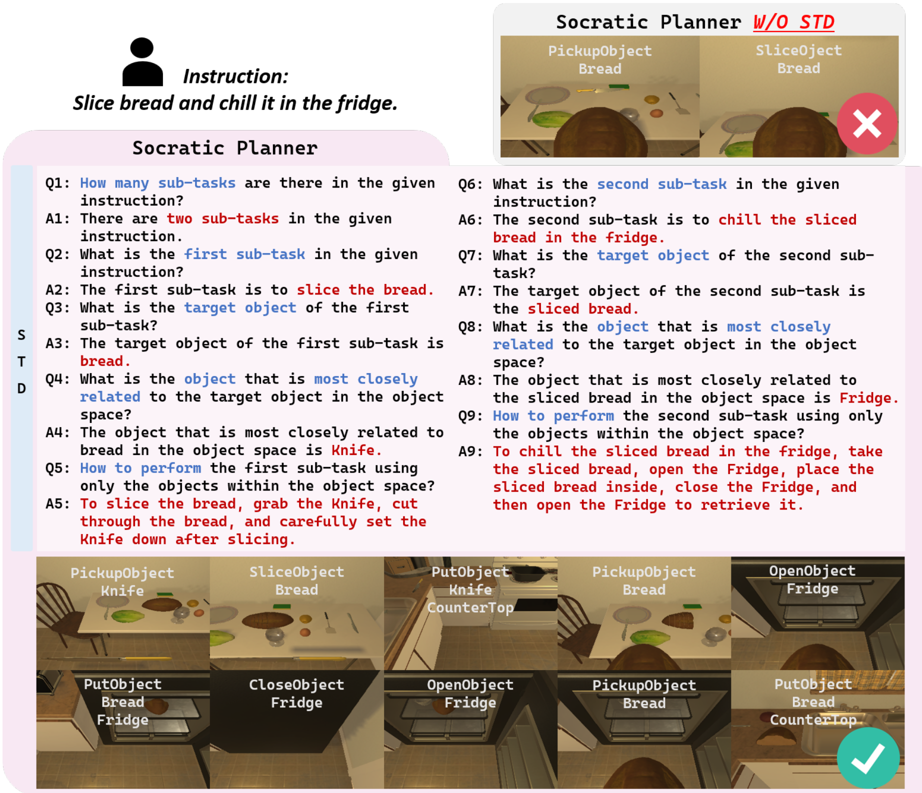

Fig. 5: Visualization of the agent action sequence acquired by Socratic Planner without STD (top right) and our Socratic Planner (bottom). The case is sampled in the static setting of the valid seen split.

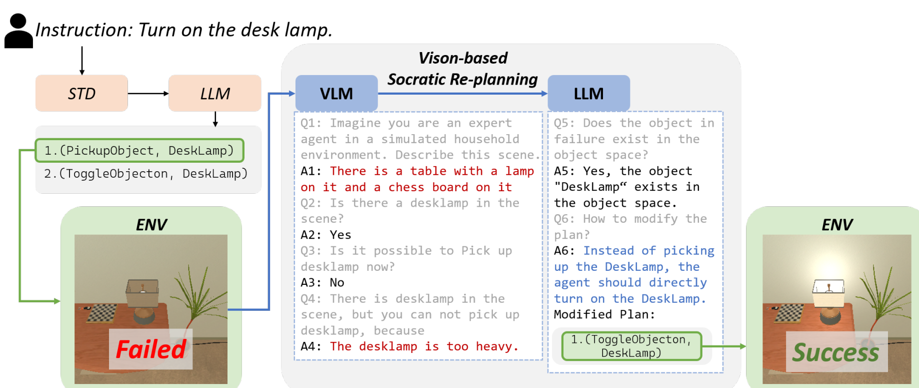

Fig. 6: Illustration of Vision-based Socratic Re-planning. The case is sampled on the test seen split.

References

1. 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) 2, 5
2. Blukis, V., Paxton, C., Fox, D., Garg, A., Artzi, Y.: A persistent spatial semantic representation for high-level natural language instruction execution. In: Conference on Robot Learning. pp. 706–717. PMLR (2022) 1, 3, 5, 6, 8, 9
3. Choi, J.W., Yoon, Y., Ong, H., Kim, J., Jang, M.: Lota-bench: Benchmarking language-oriented task planners for embodied agents. In: The Twelfth International Conference on Learning Representations (2024) 5
4. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4089–4098 (2018) 4
5. Huang, J., Gu, S., Hou, L., Wu, Y., Wang, X., Yu, H., Han, J.: Large language models can self-improve. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 1051–1068 (2023) 4
6. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: International Conference on Machine Learning. pp. 9118–9147. PMLR (2022) 4, 5
7. 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., brian ichter: Inner monologue: Embodied reasoning through planning with language models. In: 6th Annual Conference on Robot Learning (2022) 2, 4, 5
8. Inoue, Y., Ohashi, H.: Prompter: Utilizing large language model prompting for a data efficient embodied instruction following. arXiv preprint arXiv:2211.03267 (2022) 1, 5
9. Kim, B., Bhambri, S., Singh, K.P., Mottaghi, R., Choi, J.: Agent with the big picture: Perceiving surroundings for interactive instruction following. In: Embodied AI Workshop CVPR (2021) 1, 4, 5
10. Kim, B., Kim, J., Kim, Y., Min, C., Choi, J.: Context-aware planning and environment-aware memory for instruction following embodied agents. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10936–10946 (2023) 1, 5, 9
11. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22199–22213 (2022) 11, 12
12. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., et al.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017) 11
13. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023) 3, 5, 6, 8
14. Liu, H., Liu, Y., He, H., Yang, H.: Lebp–language expectation & binding policy: A two-stream framework for embodied vision-and-language interaction task learning agents. arXiv preprint arXiv:2203.04637 (2022) 1, 5
15. Lu, G., Wang, Z., Liu, C., Lu, J., Tang, Y.: Thinkbot: Embodied instruction following with thought chain reasoning. arXiv preprint arXiv:2312.07062 (2023) 2, 4, 5

16. Lu, Y., Feng, W., Zhu, W., Xu, W., Wang, X.E., Eckstein, M., Wang, W.Y.: Neuro-symbolic procedural planning with commonsense prompting. In: The Eleventh International Conference on Learning Representations (2023) 2, 4
17. Min, S.Y., Chaplot, D.S., Ravikumar, P.K., Bisk, Y., Salakhutdinov, R.: FILM: Following instructions in language with modular methods. In: International Conference on Learning Representations (2022) 1, 5, 9
18. Murray, M., Cakmak, M.: Following natural language instructions for household tasks with landmark guided search and reinforced pose adjustment. IEEE Robotics and Automation Letters **7**(3), 6870–6877 (2022) 1, 5, 9
19. Nguyen, V.Q., Suganuma, M., Okatani, T.: Look wide and interpret twice: Improving performance on interactive instruction-following tasks. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 923–930 (2021) 1, 4
20. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022) 2, 6
21. Pashevich, A., Schmid, C., Sun, C.: Episodic transformer for vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15942–15952 (October 2021) 1, 4, 5
22. Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N.A., Lewis, M.: Measuring and narrowing the compositionality gap in language models. arXiv preprint arXiv:2210.03350 (2022) 4
23. Raman, S.S., Cohen, V., Rosen, E., Idrees, I., Paulius, D., Tellex, S.: Planning with large language models via corrective re-prompting. In: NeurIPS 2022 Foundation Models for Decision Making Workshop (2022), `https://openreview.net/forum?id=cMDMRBe1TKs` 5
24. Seeskin, K.: Dialogue and discovery: A study in Socratic method. Suny Press (1987) 2, 6
25. Shridhar, K., Stolfo, A., Sachan, M.: Distilling reasoning capabilities into smaller language models. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 7059–7073 (2023) 4
26. Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., Fox, D.: Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 1, 3, 4, 5, 8, 9, 12
27. Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., Garg, A.: Progprompt: Generating situated robot task plans using large language models. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 11523–11530. IEEE (2023) 2, 5
28. Singh, K.P., Bhambri, S., Kim, B., Mottaghi, R., Choi, J.: Factorizing perception and policy for interactive instruction following. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1888–1897 (2021) 1
29. Song, C.H., Kil, J., Pan, T.Y., Sadler, B.M., Chao, W.L., Su, Y.: One step at a time: Long-horizon vision-and-language navigation with milestones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15482–15491 (2022) 1, 4, 5
30. Song, C.H., Wu, J., Washington, C., Sadler, B.M., Chao, W.L., Su, Y.: Llm-planner: Few-shot grounded planning for embodied agents with large language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 2, 3, 4, 5, 9, 10, 11

31. Storks, S., Gao, Q., Thattai, G., Tur, G.: Are we there yet? learning to localize in embodied instruction following. arXiv preprint arXiv:2101.03431 (2021) 1, 4, 5
32. Suglia, A., Gao, Q., Thomason, J., Thattai, G., Sukhatme, G.: Embodied bert: A transformer model for embodied, language-guided visual task completion. arXiv preprint arXiv:2108.04927 (2021) 1, 4, 5
33. Van-Quang Nguyen, T.: A hierarchical attention model for action learning from realistic environments and directives. In: European Conference on Computer Vision (ECCV) EVAL Workshop (2020) 4
34. Wang, X., Wei, J., Schuurmans, D., Le, Q.V., Chi, E.H., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. In: The Eleventh International Conference on Learning Representations (2023) 4
35. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems **35**, 24824–24837 (2022) 4, 12
36. Wu, Z., Wang, Z., Xu, X., Lu, J., Yan, H.: Embodied task planning with large language models. arXiv preprint arXiv:2307.01848 (2023) 2
37. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: The Eleventh International Conference on Learning Representations (2023) 2
38. Zhang, Y., Chai, J.: Hierarchical task learning from language instructions with unified transformers and self-monitoring. arXiv preprint arXiv:2106.03427 (2021) 1, 4, 5, 9
39. Zheng, K., Zhou, K., Gu, J., Fan, Y., Wang, J., Di, Z., He, X., Wang, X.E.: Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents. arXiv preprint arXiv:2208.13266 (2022) 2
40. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q.V., Chi, E.H.: Least-to-most prompting enables complex reasoning in large language models. In: The Eleventh International Conference on Learning Representations (2023) 4