CaPo: Cooperative Plan Optimization for Efficient Embodied Multi-Agent Cooperation

Anonymous authors
Paper under double-blind review

Abstract

In this work, we address the cooperation problem among large language model (LLM) based embodied agents, where agents must cooperate to achieve a common goal. Previous methods often execute actions extemporaneously and incoherently, without long-term strategic and cooperative planning, leading to redundant steps, failures, and even serious repercussions in complex tasks like search-and-rescue missions where discussion and cooperative plan are crucial. To solve this issue, we propose Cooperative Plan Optimization (CaPo) to enhance the cooperation efficiency of LLM-based embodied agents. Inspired by human cooperation schemes, CaPo improves cooperation efficiency with two phases: 1) meta-plan generation, and 2) progress-adaptive meta-plan and execution. In the first phase, all agents analyze the task, discuss, and cooperatively create a meta-plan that decomposes the task into subtasks with detailed steps, ensuring a long-term strategic and coherent plan for efficient coordination. In the second phase, agents execute tasks according to the meta-plan and dynamically adjust it based on their latest progress (e.g., discovering a target object) through multi-turn discussions. This progress-based adaptation eliminates redundant actions, improving the overall cooperation efficiency of agents. Experimental results on the ThreeDworld Multi-Agent Transport and Communicative Watch-And-Help tasks demonstrate CaPo's much higher task completion rate and efficiency compared with state-of-the-arts.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding and generating human language, complex reasoning, and planning, achieving impressive performance (OpenAI, 2024; Touvron et al., 2023). These advancements empower LLM-based embodied agents to autonomously make plans (Li et al., 2023a; Padmakumar et al., 2022; Zhu et al., 2023; Wang et al., 2023; Wu et al., 2023b; Huang et al., 2022b) and perform reasoning (Du et al., 2023; Hao et al., 2023; Zhou et al., 2024; Huang et al., 2022a) by using human language to assist people in daily activities, such as housework and daily chores. The next milestone for agents is to cooperate with others to achieve joint tasks. This is crucial not only for efficiently performing simple tasks but also for tackling complex ones that cannot be completed in isolation due to their inherent complexity or the dynamic nature of the environment (Zhang et al., 2023b; Guo et al., 2024; Mandi et al., 2023; Zhang et al., 2023a).

Notably, the cooperation among LLM-based embodied agents is rarely investigated despite being highly desired. Conventional works often focus on adopting reinforcement learning (RL) (Jiang & Lu, 2018; Liu et al., 2021; Wang et al., 2021) to explore the dynamics of cooperative behavior among non-LLM-based agents. In spite of their promising performance in certain scenarios, RL-based cooperation methods exhibit limited adaptability across different tasks (Dittadi et al., 2021; Cobbe et al., 2019), since they are often not trained on large-scale data and lack sufficient generalization ability. To solve this issue, in this work, we are particularly interested in the problem of "how to develop an effective collaboration framework for LLM-based agents", since LLMs have revealed strong reasoning, planning, and communication ability across different tasks and thus are regarded as good agents' brains.

Among the limited related works, CoELA (Zhang et al., 2023b) proposes an LLM-based multi-agent cooperation framework in which after each action execution, agents communicate to devise

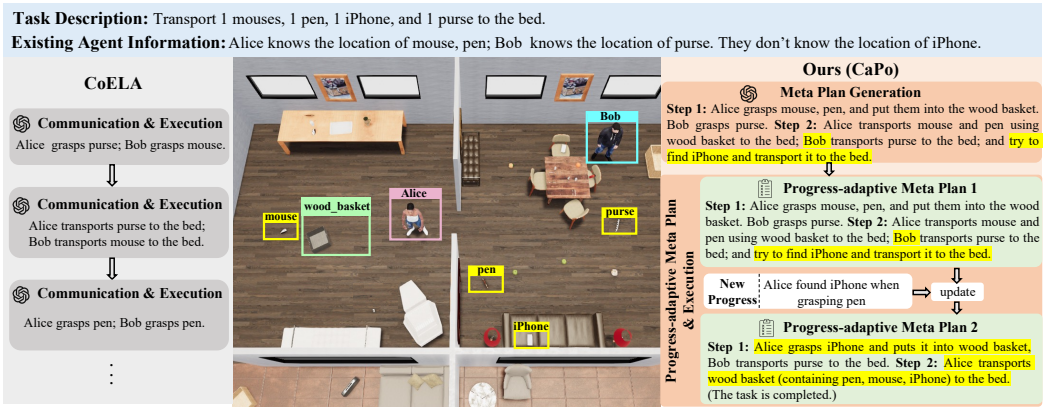

Figure 1: **Procedure example of task accomplishment of CoELA (Zhang et al., 2023b) and our CaPo**. In CoELA, after each action execution, Alice and Bob communicate to decide next action which is a greedy single-step plan and suboptimal. For example, they do not use wood basket which can contain several objects, and both extemporaneously move a single item to the target bed without a long-term strategic and collaborative plan. Differently, in CaPo, Alice and Bob first discuss to make a long-term meta-plan for strategical cooperation in which Alice is arranged to move several target items into a wood basket, and Bob moves the remaining target items and also searches the unknown objects. Then during execution phase, both follow the meta-plan to accomplish task, and dynamically adapt the meta-plan the latest task progress, ensuring its effectiveness and efficiency in coordinating agents.

a single-step plan for the next action. Despite its significant advancements, CoELA's short-term, single-step planning, which lacks consideration for long-term strategic collaboration, often results in extemporaneous and incoherent actions among agents, leading to several potential issues. Firstly, without a long-term coherent collaboration plan, it leads to numerous redundant action steps and increased costs, since agents' movement is not easy and is indeed expensive in the physical world. For instance, as shown in Fig. 1, for the object transport task, agent Alice and Bob do not use the wood basket which can contain several objects, and extemporaneously move their nearest target objects one by one, leading to inferior efficiency. Moreover, complex tasks are difficult to accomplish without thorough discussion and long-term collaboration, especially in (embodied) environments where each agent has only partial observations. Finally, without a long-term cooperative plan, agents' extemporaneous actions can result in mistakes with severe consequences. For instance, in search-and-rescue missions, poor coordination can have dire outcomes, such as endangering human lives due to the complex nature of these operations.

Contributions. To address the above issues, we propose a novel and effective *Cooperative Plan Optimization (CaPo)* framework that uses LLMs' strong reasoning and planning ability to enhance the cooperation efficiency of LLM-based embodied agents. Inspired by human cooperation schemes (Tuomela, 1998; Thürmer et al., 2017), CaPo engages agents in multi-turn discussions to create and update a long-term strategic and coherent meta-plan, providing step-by-step guidance to coordinate agents and efficiently complete tasks.

Specifically, to accomplish a task, CaPo consists of two phases: 1) meta-plan generation, providing long-term strategical and coherent guidance for coordinating agents, and 2) progress-adaptive meta-plan and execution, dynamically adapting the meta-plan to agents' latest progress. In the first phase, agents analyze the task and discuss with other agents for collecting relevant information. Next, one agent is responsible for making a meta-plan which decomposes the task into subtasks with detailed accomplishment steps like agent allocations, and then collects the feedback from other agents for further meta-plan refinement. The steps of meta-plan generation and refinement will continue until all agents reach a consensus or the communication cost is exhausted. This approach ensures the thorough discussion and analysis of all agents, helping to make a long-term strategical and coherent meta-plan for efficiently coordinating all agents. For example, as illustrated in Fig. 1, in the object transport task, agents Alice and Bob are strategically assigned to different subtasks.

In the second phase, as shown in Fig. 1, agents follow the meta-plan from the first phase, and focus on their assigned subtasks. As progress is made, agents may complete subtasks or make

important observations, such as Alice in Fig. 1 discovering the object "iPhone" which is Bob's target. Accordingly, agents dynamically adapt meta-plan to the latest task progress through multi-turn discussions, allowing Alice to handle the object "iPhone" and complete the task efficiently. This progress-adaptive approach ensures that the meta-plan remains effective in coordinating all agents, thereby enhancing cooperation efficiency.

Finally, experimental results demonstrate that CaPo significantly improves task completion rates and efficiency compared to state-of-the-art (SoTA) methods on the widely used ThreeDworld Multi-Agent Transport task (Zhang et al., 2023b) (object transport task) and the Communicative Watch-And-Help task (Zhang et al., 2023b) (household chore task). For instance, on the ThreeDworld Multi-Agent Transport task, CaPo surpasses the SoTA CoELA by 16.7% and 4.7% in completion rate with GPT-3.5 and GPT-4 based agents, respectively

2 Related Work

LLM-based Agents. LLM-based agents (Hong et al., 2023; Wang et al., 2024; Shen et al., 2024; Liu et al., 2023a) are designed to autonomously perceive environments, execute actions, accumulate knowledge, and evolve themselves, with rich real world knowledge and complex reasoning capability inherited from LLMs. Notable agents like AutoGPT (Richards & et al, 2021), BabyAGI (Nakajima, 2023), and AgentGPT (Reworkd, 2023) showcase remarkable proficiency in decision-making and complex reasoning. In the embodied environment, LLM-based agents have shown superior capacity in strategic planning (Li et al., 2023a; Padmakumar et al., 2022; Wu et al., 2023b; Huang et al., 2022b). Specifically, LLM-planner (Song et al., 2023) harness LLMs to do few-shot planning for embodied agents. PET (Wu et al., 2023a) translates a task description with LLMs into a list of high-level sub-tasks. TaPA wu2023embodied enables the agent to generate executable plans by aligning LLMs with visual perception models. Another line of research focuses on harnessing LLMs's reasoning capabilities in embodied tasks (Zhou et al., 2024; Huang et al., 2022a). ELLM (Du et al., 2023) utilizes LLMs to set pretraining goals in RL, guiding agents towards the goal without human involvement.

Multi-Agent Cooperation. Multi-agent cooperation and communication have been studied for decades to improve communication efficiency (Jiang & Lu, 2018; Li et al., 2023b) and planning (Torreno et al., 2017; Zhang et al., 2023a). Within the domain of embodied intelligence, ProAgent (Zhang et al., 2023a) harnesses LLMs to develop proactive agents that dynamically adjust their behavior to foster better cooperation with teammates. RoCo (Mandi et al., 2023) introduce a multi-robot collaboration framework that employs LLMs for both high-level communication and low-level path planning. (Guo et al., 2024) proposed a prompt-based organizational framework for LLM agents to reduce communication costs and boost team efficiency. CoELA (Zhang et al., 2023b) enables agents to plan, communicate, and collaborate effectively, but its plan is one-step plan and is short-term. Despite these advancements, these methods focus on short-term planning and do not involve sufficient agent discussion, while ours seeks to a long-term strategical and coherent plan via agent's thoughtful discussions for efficient multi-agent cooperation.

Optimization with LLMs. With the advancement of prompting techniques, LLMs have shown remarkable performance across various domains (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022; Zhou et al., 2022; Madaan et al., 2024). Their ability to understand natural language lays out a new possibility for optimization. (Yang et al., 2023) first proposed to leverage LLMs as optimizer, where the optimization task is described in natural language. OPT2I (Mañas et al., 2024) aims to enhance prompt-image consistency in text-to-image models by iteratively generating revised prompts with LLMs to maximize the consistency score. VislingInstruct (Zhu et al., 2024) proposes optimizing multi-modal instruction for multi-modal language models in a zero-shot manner. DyLAN (Liu et al., 2023b) is particularly relevant to our work. DyLAN (Liu et al., 2023b) enables agents to interact for multiple rounds in a dynamic architecture to optimize the selection of agent. In contrast, our work investigates cooperative plan optimization via multi-turn discussion between agents.

3 Preliminaries

Task Formulation. We formulate the embodied multi-agent cooperation task as an decentralized partially observable Markov decision process (DEC-POMDP) (Bernstein et al., 2002; Spaan et al.,

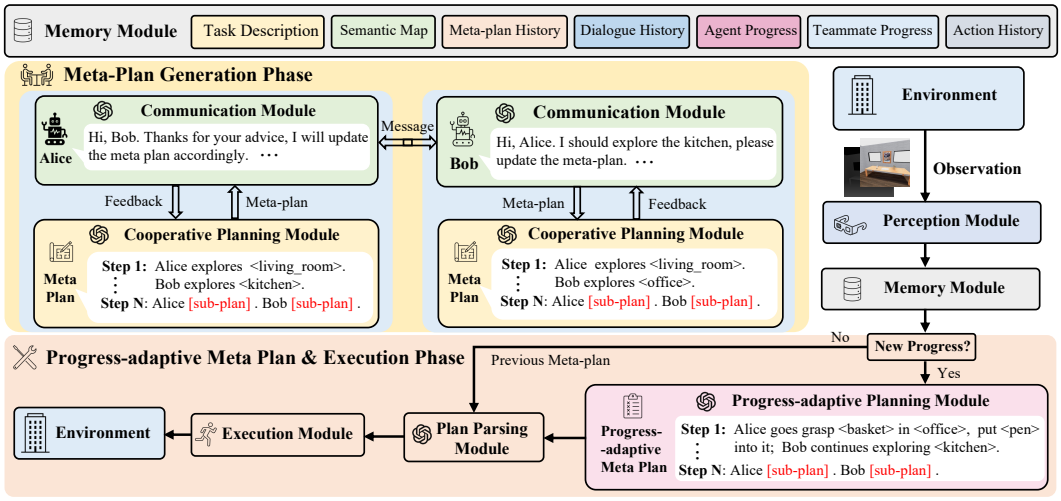

Figure 2: **Overview of the *CooperAtive Plan Optimization (CaPo)* framework for embodied multi-agent cooperation.** CaPo consists of two key phases: 1) **meta-plan Generation**: All agents collaboratively formulate a meta-plan before taking any actions through multi-turn discussions. One agent serves as meta-plan designer, responsible for creating the meta-plan, while all other agents serve as meta-plan evaluators, providing critical feedback about meta-plan. 2) **Progressive-adaptive meta-plan and Execution**: As new progress is made, agents adopt a progress-adaptive planning module to adapt the meta-plan to the latest task progress, ensuring the effectiveness of meta-plan.

2006), which is defined as $< n, \mathcal{S}, \mathcal{O}, \mathcal{A}, P, r, \gamma >$. Here, n represents the number of agents; $\mathcal{S}$ is the finite state space; $\mathcal{O}$ denotes the observation space; $\mathcal{A}$ is a finite joint action space of all agents; $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes the transition probability function; $r = \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function; $\gamma \in [0, 1]$ denotes the discount factor. In this framework, at time step $t \in \mathbb{N}$, each agent i observes the environment's state $s_t \in \mathcal{S}$, and receives an observation set $\mathcal{O}_i$. $\mathcal{O}_i$ consists of a world observation $\mathcal{O}_i^w$, which the agent gathers through its sensors, or a communication message observation $\mathcal{O}_i^c$ from other teammate agents. Agent i takes actions from its action space $\mathcal{A}_i$, which includes a finite set of world action $\mathcal{A}_i^w$, e.g., grasping a target object, or a finite set of messaging action $\mathcal{A}_i^c$. Then agents receive a shared reward $r_t = r(s_t, a_t)$, where $a_t \in \mathcal{A}$ denotes the joint actions of agents, and observe a new state s_{t+1} with probability $P(s_{t+1}|s_t, a_t)$. We formulate the problem with two decentralized intelligent embodied agents working together to complete a long-horizon rearrangement task (Zhang et al., 2023b; Batra et al., 2020) in a multi-room indoor environment. During the task, agents can execute multiple kinds of actions, such as navigation, interaction, and communication by sending messages.

CoELA Framework. CoELA Zhang et al. (2023b) is a pioneering modular framework for embodied multi-agent cooperation, which consists of five key modules: (a) Perception, (b) Memory, (c) Communication, (d) Planning, and (e) Execution. For each agent, the (a) Perception Module gathers observations from environment, including messages from other agents and relevant scene information from the RGB-D image. The (b) Memory Module dynamically stores the shared task, dialogue history between agents, agent progress, team- mate progress, and action history, all formatted as text descriptions. (c) The Communication Module retrieves relevant information from the memory module and uses an LLM to generate messages that are sent to other agents. (d) The Planning Module driven by LLM decides which plan to take given related information retrieved from the memory module and available actions proposed regarding the current state. Finally, the (e) Execution Module convert high-level plan into primitive actions executable in the environment. CoELA exhibits great performances in the embodied multi-agent cooperation tasks, making it a strong baseline for validating the effectiveness of our model in improving multi-agent cooperation efficiency.

4 Cooperative Plan Optimization

We first introduce the overall framework of CooperAtive Plan Optimization (CaPo) for LLM-based embodied agents in Sec. 4.1. We then respectively elaborate on the two key phases of CaPo, i.e., meta-plan generation and progress-adaptive meta-plan and execution, in Sec. 4.2 and Sec. 4.3.

4.1 OVERALL FRAMEWORK OF CAPO

CaPo aims to enhance cooperation efficiency of LLM-based embodied agents. Its key idea is to create a long-term and dynamic meta-plan for strategically and coherently coordinating agents to complete a task step by step. The pipeline of CaPo is illustrated in Fig. 2. We inheritate five key modules from CoELA (Zhang et al., 2023b), including perception, memory, communication, planning, and execution modules. To distinguish between CoELA's planning module and the meta-plan planning process in CaPo, we refer to CoELA's planning module as the plan parsing module. Additionally, we introduce two novel modules: the Cooperative Planning Module and the Progress-Adaptive Planning Module. The Cooperative Planning Module generates or refines the meta-plan, while the Progress-Adaptive Planning Module dynamically updates the meta-plan based on agents' latest task progress.

To complete a task cooperatively and efficiently, inspired by humans collaboration (Tuomela, 1998; Thürmer et al., 2017), CaPo first analyzes the task at hand to create a long-term meta-plan before agents take any actions. All agents participate in this plan-making process, either generating meta-plan or providing feedback. The meta-plan is then dynamically refined based on the latest agent progress to ensure its effectiveness in coordinating agents. To this end, it contains two key phases, including 1) meta-plan generation, and 2) progress-adaptive meta-plan and execution. In the meta-plan generation phase, given a task, multiple embodied agents first gather relevant information such as object locations. Then, they discuss together to create a meta-plan that decomposes the task into subtasks and consider agent situation (e.g., agent and object locations) to assign agents to different subtasks with accomplishment steps. In the progress-adaptive meta-plan and execution phase, agents dynamically align the meta-plan with their latest progress. This is achieved through multi-turn discussion triggered by clear task progress, such as discovering target objects or successfully completing subtasks. In the following, we will elaborate on these two phases in turn.

4.2 META-PLAN GENERATION

To generate the long-term meta-plan which coordinates all agents to accomplish tasks efficiently, CaPo introduces two key steps, including 1) meta-plan initialization where one agent initializes a meta-plan according to the task description and existing information, and 2) meta-plan evaluation and optimization where all agents evaluate the meta-plan and provide feedback to improve the plan.

Meta-plan Initialization. At the beginning of a task, the task description is provided to all agents, e.g, `Transport 2 apples and 3 bananas to the bed`. One agent, e.g., Alice in Fig. 2, is randomly selected as the meta-plan designer, and creates the meta-plan through a *cooperative planning module*. Note that the meta-plan here, as illustrated in Fig. 3, differs from the short-term or unorganized plans used in previous work (Zhang et al., 2023b;a; Mandi et al., 2023). Specifically, the cooperative planning module is equipped with a pre-trained LLM, and leverage the LLM to generate the meta-plan. The prompting for the LLM is organized as follows:

Prompt: `<Task Desc>` + `<Instruct Head>` \n. LLM: `<Meta-plan>`.

Here, `<Task Desc>`, `<Instruct Head>`, and `<Meta-plan>` are three placeholders for the task description, instruction head, and generated meta-plan. The task description provides background descriptions about the task, while the instruction head introduces additional constraints into the generation of meta-plan, such as the format of meta-plan and available actions to generate a clear and executable plan. Detailed prompt design is shown in Fig. 9 of Appendix.

Meta-plan Evaluation and Optimization. The meta-plan generated by a single agent is often biased by that agent's partial observations, resulting in a suboptimal plan that fails to coordinate all agents effectively. To address this issue, CaPo involves all agents in a multi-turn discussion to optimize the meta-plan. Specifically, the meta-plan designer (e.g., Alice in Fig. 3) broadcasts the meta-plan to all teammate agents, while teammate agents (e.g., Bob in Fig. 3) serve as meta-plan evaluators, providing feedback about the meta-plan. Since teammate agents have different partial observations of the environment, they provide the meta-plan designer with better situational awareness to help generate a more efficient and effective meta-plan. This optimization process continues until all agents reach a consensus, i.e., the evaluator agents are satisfied with the meta-plan, or the communication budget (e.g., maximum discussion round) is exhausted. Indeed, Fig. 8

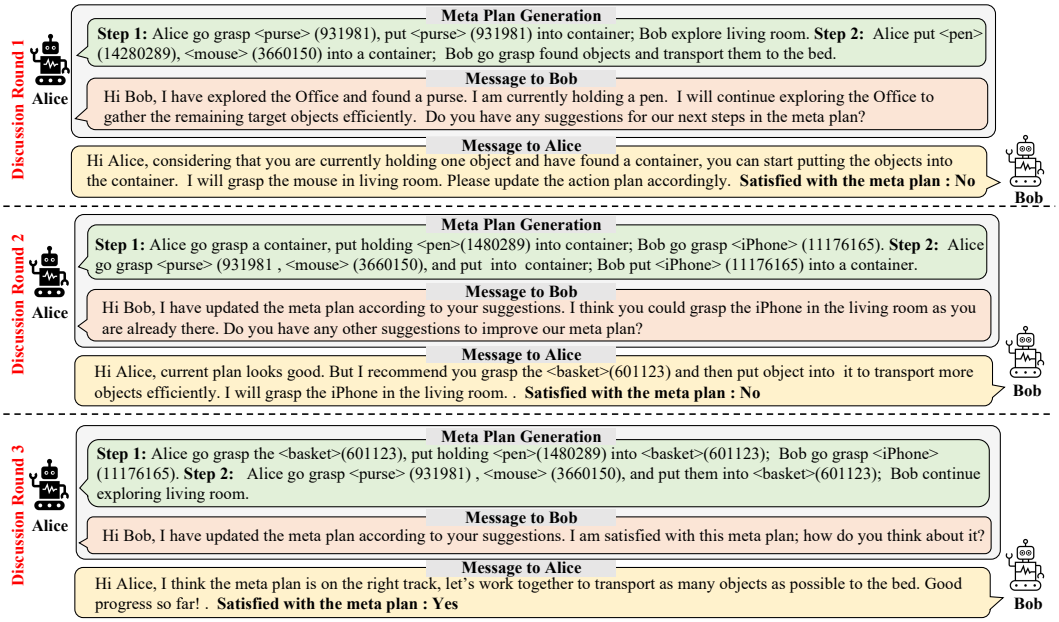

Figure 3: **Examples of the evaluation and optimization process of meta-plan via multi-turn discussion between agents**. The discussion is triggered by new progress, i.e., Alice founds new object 'purse'. Here, Alice acts as the meta-plan designer, while Bob serves as the meta-plan evaluator. The example is derived from the transporting task of TDW-MAT.

in Sec. 5.2 analyzes the convergence analysis of the meta-plan optimization process, and shows that typically agents would reach a consensus within three rounds of discussion.

As shown in Fig. 2, each agent is equipped with a communication module powered by a pretrained LLM to facilitate multi-turn discussions. Specifically, the communication module first retrieves relevant information from the memory, e.g., meta-plan, agent state, and previous dialogue history among agents, then prompts the LLM to generate the message to send via the following prompt:

```
Prompt: <Task Desc> + <Instruct Head> + <Meta-plan> + <Agent State> +
<Dialog History> \n.                    LLM:  <Messages> .
```

The tags `<Meta-plan>`, `<Agent State>`, and `<Dialog History>` act as placeholders for inserting the meta-plan, the agent's state, and the dialogue history between agents. The tag `<Instruct Head>` differs for the meta-plan designer and evaluator: the former instructs the LLM to generate messages asking teammates for their opinions, while the latter focuses on providing feedback on the meta-plan. After receiving feedback from the teammate agents, the meta-plan creator reinitiates the process to generate a new meta-plan. Fig. 3 illustrates the evaluation and optimization process of a meta-plan through multi-turn discussions among agents. It is evident that the optimized meta-plan effectively integrates partially observed information from all agents, resulting in improved coordination and efficiency. Detailed prompt designs for the communication module can be found in Fig. 10 and 11 in the Appendix.

4.3 Progress-Adaptive Meta-plan & Execution

The optimized meta-plan acts as a high-level guide, assigning subtasks to each agent and coordinating them for efficient task completion. However, due to dynamic environmental changes and task progress updates, the meta-plan can become outdated during execution. As illustrated in Fig. 4, agents may encounter significant progress, such as discovering target objects or completing subtasks, necessitating adjustments to the meta-plan. In such cases, the previous plan becomes less effective or invalid for coordinating the agents.

To address this, we design a *progress-adaptive planning module* for CaPo for adapting the meta-plan to the agents' latest progress. This module follows a similar process as described in Sec. 4.2—meta-plan initialization, evaluation, and optimization—but with modified prompting strategies for the LLMs. Whenever an agent makes new progress, the meta-plan designer promptly generates an updated meta-plan, followed by a multi-turn discussion among all agents to further optimize it. The LLM prompting strategies for the progress-adaptive planning module are structured as follows:

Prompt: `<Task Desc>` + `<Instruct Head>` + `<Meta-plan>` + `<Agent Progress>` + `<Teammate Progress>` + `<Dialog History>` \n.
LLM: `<meta-plan>` or `<Messages>`.

Here we introduce two placeholders, `<Agent Progress>` and `<Teammate Progress>`, to capture the task progress of agents and enable the LLM to generate progress-aware responses, such as meta-plans or communication messages. Agents engage in discussions to optimize the meta-plan until a consensus is reached or communication resources are exhausted (e.g., after three discussion rounds). Detailed prompt designs for the LLMs—responsible for generating the meta-plan and facilitating messages for both the meta-plan designer and evaluator—are provided in Fig. 11∼12.

Once the meta-plan or progress-adaptive meta-plan is established, each agent autonomously transforms the plan into executable actions via a plan parsing module and an execution module. The plan parsing module generates the latest sub-plan by retrieving relevant information from the memory module and converting it into text descriptions, and then compiles an Action List of all available high-level sub-plans. We implement the plan parsing module as a pretrained LLM, and prompt it with a concatenation of `Instruct Head`, `Task Description`, `meta-plan`, `Action History`, `Agent Progress`, and `Action List` to choose the most suitable sub-plan. See Fig. 13 in Appendix for more prompt details. Given the sub-plan, we adopt a similar execution module as in (Zhang et al., 2023b) to generate primitive actions for executing the sub-plan.

(a) Discover a target object (b) Complete a subtask

Figure 4: **Two types of new progress during task execution.** (a) Discover a new object poundcake. (b) complete a subtask.

5 Experiments

Benchmarks. We follow CoELA, and adopt the ThreeDworld Multi-Agent Transport (TDW-MAT) task (Zhang et al., 2023b), and the Communicative Watch-And-Help (C-WAH) task (Zhang et al., 2023b) to test our CaPo. TDW-MAT is built on the general purpose virtual world simulation platform TDW platform (Gan et al., 2020), and requires agents to move objects by their hands or containers which can contain several objects for efficient moving to the destination. Moreover, agents can receive ego-centric 512×512 RGB-D images as observation and can communicate with others. The test set of TDW-MAT consists 24 episodes, which evenly divided into food and stuff tasks. In C-WAH, agents are requested to complete five types of household activities, represented as various predicates with specific counts that must be satisfied. The test set contains 10 episodes, including both *symbolic and visual observation settings*. More details about TDW-MAT and C-WAH environments are provided in Appendix B.1 and B.2, respectively.

Metrics. On TDW-MAT, we adopt *Transport Rate*, i.e., the fraction of subtasks completed within 3000 time steps (a.k.a. frames), as performance metric. Note, one action step may last multiple time steps, e.g., resetting arms. On C-WAH, *Average Steps* to complete all tasks is used as the metric to evaluate cooperation efficiency.

Implementation. Following CoELA, we test two settings on TDW-MAT task: 1) a real-world setting where the perception module is instantiated as Mask-RCNN (He et al., 2017) that is trained using collected scene images (Zhang et al., 2023b), and 2) an oracle setting with segmentation ground-truth. We use GPT-3.5-turbo and GPT-4 from the OpenAI API (OpenAI, 2024), and LLAMA-2-13B-CHAT (Touvron et al., 2023), as LLMs in embodied agents. We set default parameters for

	Classic Agents		GPT-3.5 Agents				LLAMA-2 Agents			
	RHP*†	RHP†	CoELA†	ProAgent	RoCo	CaPo$_{(ours)}$	CoELA†	ProAgent	RoCo	CaPo$_{(ours)}$
w/o Oracle Perception										
Food (↑)	49	67	67	68 $_{+1.5\%}$	64 $_{-4.5\%}$	70 $_{+4.5\%}$	57	60 $_{+5.3\%}$	59 $_{+3.5\%}$	66 $_{+15.8\%}$
Stuff (↑)	36	54	39	44 $_{+12.8\%}$	47 $_{+20.1\%}$	45 $_{+15.4\%}$	48	55 $_{+14.6\%}$	51 $_{+6.3\%}$	56 $_{+16.7\%}$
Avg. (↑)	43	61	52	56 $_{+7.7\%}$	55 $_{+5.6\%}$	57 $_{+9.6\%}$	53	57 $_{+7.5\%}$	55 $_{+3.8\%}$	61 $_{+15.1\%}$
w/ Oracle Perception										
Food (↑)	52	76	72	80 $_{+11.1\%}$	74 $_{+2.8\%}$	85 $_{+18.1\%}$	60	64 $_{+6.7\%}$	62 $_{+3.3\%}$	66 $_{+10.0\%}$
Stuff (↑)	49	74	73	76 $_{+4.1\%}$	80 $_{+9.6\%}$	84 $_{+15.1\%}$	63	62 $_{-1.6\%}$	69 $_{+9.5\%}$	76 $_{+20.6\%}$
Avg. (↑)	50	75	72	78 $_{+8.3\%}$	77 $_{+6.9\%}$	84 $_{+16.7\%}$	62	63 $_{+1.6\%}$	65 $_{+4.8\%}$	71 $_{+14.5\%}$

Table 1: **Comparison of average Transport Rate (TR, %) of all baselines on the TDW-MAT w/o and w/ Oracle Perception task.** Each task requires agents to move two kinds of items, including Food and Stuff. RHP* uses a single agent while all others adopt two agents. † denotes results quoted from CoELA. The subscript value like $+9.6\%$ in 57 $_{+9.6\%}$ denotes the relative improvement when comparing with baseline CoELA.

	Classic Agents		GPT-4 Agents			
	MHP*†	MHP†	CoELA†	ProAgent	RoCo	CaPo$_{(ours)}$
Symbolic Obs (↓)	111	75	57	62 $_{-8.8\%}$	57 $_{+0.0\%}$	51 $_{+10.5\%}$
Visual Obs (↓)	141	103	92	90 $_{+2.2\%}$	89 $_{+3.2\%}$	83 $_{+9.8\%}$

Table 2: **Comparison of Average Step (AS) of all compared methods on the C-WAH task.** Symbolic obs and visual obs denote symbolic observation and visual observation settings, respectively. MHP* uses a single agent while all others adopt two agents. † denotes results quoted from CoELA.

LLMs: temperature of 0.7, a maximum of 256 output tokens, and top-p sampling with $p = 1$. Our code will be made publicly available.

Baselines. We adopt two types of methods as our baseline: 1) classical agents, including MCTS-based Hierarchical Planner (**MHP**) (Puig et al., 2020) and Rule-based Hierarchical Planner (**RHP**) (Gan et al., 2022). 2) LLM-driven agents, including CoELA Zhang et al. (2023b), ProAgent (Zhang et al., 2023a), and RoCo (Mandi et al., 2023). CoELA Zhang et al. (2023b) features a modular framework for multi-agent planning, communication, and complete long-horizon tasks, but generate independent short-term plan for each agent. ProAgent (Zhang et al., 2023a), and RoCo (Mandi et al., 2023) generate joint plans for cooperative agents, and introduce a reflection loop or environment feedback for plan validation. See more details in Appendix A.1.

5.1 Main Results

Performance comparison. We follow CoELA to test two-agent cooperation setting, and compare with classical methods like MHP and RHP, and LLM-driven methods CoELA, ProAgent, and RoCo.

Table 1 summarizes the performance of all compared methods under the two settings of the TDW-MAT task, and shows several observations. **1)** Compared with the single-agent baseline RHP, CaPo and all two-agent baselines consistently make significant improvements, showing the effectiveness of multi-agent cooperation in embodied tasks. **2)** In multi-agent comparisons, our CaPo outperforms LLM-driven methods by a remarkable margin, e.g., respectively making 16.7% and 8.4% improvement over CoELA and RoCo under the oracle perception setting. **3)** CaPo with different LLMs as agent brain exhibits consistent superior performance across all settings. Indeed, CaPo with LLAMA-2 achieves comparable performance with CoELA with GPT-3.5-turbo under oracle perception setting. The improvement of CaPo is derived from its meta-plan and progress-adaptive meta-plan, which both provide strategical and coherent guidance for agent cooperation, thereby improving cooperation performance.

Table 2 reports the performance of all methods on the C-WAH task, and shows similar and consistent observations to those on the TDW-MAT task. Specifically, with GPT-4 agents, our CaPo respectively makes 9.8%, 7.6% and 6.6% relative improvement on CoELA, ProAgent and RoCo, consistently highlighting the superiority of CaPo in enhancing multi-agent cooperation.

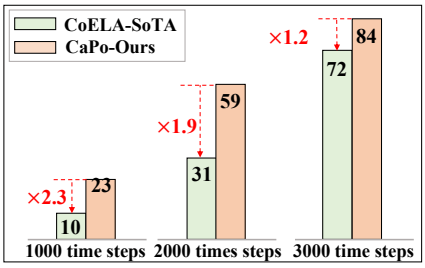

Figure 5: **Comparison of Transport Rate (%) of CoELA and CaPo using GPT-3.5 under different time steps**.

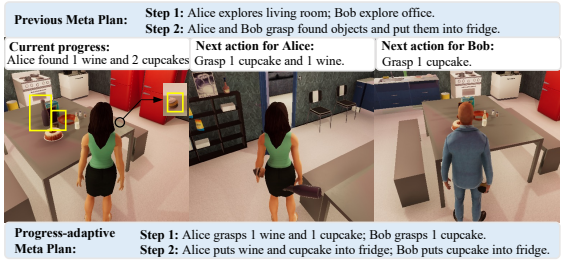

Figure 6: **Example of progress-adaptive meta-plan adaptation.** New progress: Alice found target objects, 1 wine and 2 cupcakes.

Figure 7: **Examples of cooperative behaviors introduced by meta-plan.** Guided by meta-plan, agents show clear work and task allocation, thereby improving cooperative efficiency.

Efficiency comparison. To demonstrate the cooperation efficiency, we compare the transport rates of CaPo and CoELA with different time steps on TDW-MAT. As shown in Fig. 5, with the same time step budget and GPT-3.5 agents, CaPo consistently outperforms CoELA across various time steps, indicating its superiority to coordinate agents effectively. The improvement is particularly clear in scenarios of small time steps. For example, given 1,000 time steps, CaPo doubles the transport rate of CoELA by improving 10% to 23%. This shows that with limited time or resources, a cooperative meta-plan can significantly improve cooperation efficiency.

Qualitative Analysis. Here we investigate the agents' behavior in CaPo with GPT-4 on the C-WAH task. In the meta-plan generation phase, as shown in Fig. 3, agents ask questions, provide feedback, and collaboratively refine the initial meta-plan. Moreover, in this phase, Fig. 7 shows that with meta-plan as guidance, two agents, Alice and Bob, have clear work/labor allocation to complete tasks, thereby avoiding redundant steps and improving cooperation efficiency. For the progress-adaptive meta-plan and execution phase, Fig. 6 also shows that when agent Alice achieves progress, e.g., discovering three target objects, both agents will accordingly discuss to adapt the meta-plan, e.g., grasping 1 wine and 1 cupcake by Alice. This ongoing adaptation of the meta-plan provides strategic, coherent, and timely guidance, facilitating efficient coordination among agents and ultimately enhancing multi-agent cooperation.

5.2 Ablation Study

Effects of each component in CaPo. Here we examine the effects of two key components: 1) meta-plan generation, which includes meta-plan initialization, evaluation, and optimization, and 2) the progress-adaptive meta-plan. To evaluate their impact, we first remove both components from CaPo, resulting in CaPo_1. As shown in Table 3, CaPo_2, which includes meta-plan initialization but freezes the meta-plan during subsequent procedures, improves upon CaPo_1 by approximately 1% across three metrics, demonstrating the value of meta-plan initialization. Similarly, CaPo_3, which incorporates the full meta-plan generation process, outperforms CaPo_2 by a significant margin, highlighting the benefits of meta-plan evaluation and optimization. Finally, CaPo achieves a 7% improvement over CaPo_3, showcasing the effectiveness of the progress-adaptive meta-plan. These results underscore the importance of each component in the CaPo framework.

Effects of agent number. Table 4 investigates the effects of agent number in CaPo using GPT-4 on the C-WAH task, where "CaPo × C" denotes using C GPT-4 agents. We can observe that

Method	Food (↑)	Stuff (↑)	Avg. (↑)
CaPo$_1$ (No MP + No Pro. MP)	72	75	73
CaPo$_2$ (MP Initialization + No Pro. MP)	73	76	74
CaPo$_3$ (MP Generation + No Pro. MP)	74	80	77
CaPo (MP Generation + Pro. MP)	**85**	**84**	**84**

Table 3: **Effects of the components in CaPo using GPT-3.5 on TDW-MAT task**. We report the transport Rate (TR, %). MP" denote 'Meta Plan" and Progress-Adaptive Meta Plan", respectively.

Method	Symbolic Obs (↓)	Visual Obs (↓)
CaPo×1	93	106
CaPo×2	51	83
CaPo×3	46	**72**
CaPo×4	**45**	74

Table 4: **Benefits of increasing agent numbers in our CaPo using GPT-4 on the C-WAH task**. Average steps required to complete task are reported.

increasing agent number to 3 significantly reduces the overall time step number required to complete tasks. This improvement also shows the effectiveness of our CaPo on multiple agent cooperation. However, increasing the number of agents to four results in only minor or degraded improvements. This is because for simple tasks, agents are too much and suffer from highly-frequent agent dispatch, leading to inferior collaboration efficiency. For instance, setting up a dining table does not require four waiters, as a maximum of two agents is sufficient.

Progress in meta-plan adaptation. Fig. 4 in Sec. 4.3 shows two clear task progress examples: 1) discovering a target object and 2) completing a subtask, both of which can trigger agents to adapt the meta-plan to their latest task progress. Such progress is crucial, as agents need to continually refine the meta-plan to complete tasks efficiently and maximize cooperation. Conversely, actions without significant progress, such as entering a new room, do not prompt agents to adjust the current (progress-adaptive) meta-plan. This is because it would be unnecessary, and updating the meta-plan involves communication, which incurs additional time overhead.

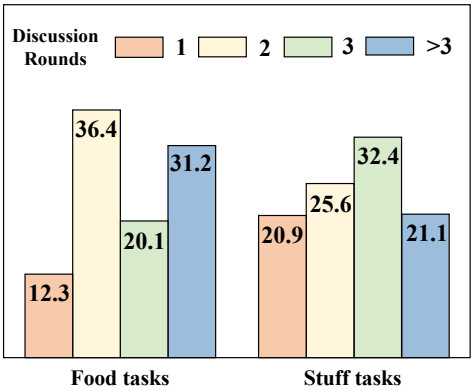

Figure 8: **Percentage (%) of discussion rounds needed for agents to reach consensus on a meta-plan**, based on results from TDW-MAT.

Convergence analysis of agent discussion. Here we investigate the convergence of agent discussions, specifically focusing on how many rounds are required for agents to reach a consensus on the meta-plan. In the TDW-MAT environment, we set the maximum number of discussion rounds—referred to as the discussion budget—at three. As shown in Fig. 8, agents reach consensus on the new meta-plan within three rounds in most cases, with 78.9% achieving consensus in the "Stuff" tasks. This meta-plan, which incorporates the states of all agents, enables more efficient cooperation in task completion. Furthermore, by limiting the number of discussion rounds, CaPo strikes a balance between discussion effectiveness and budget, preventing unnecessary or prolonged discussions.

6 Conclusion

In this work, we introduce Cooperative Plan Optimization (CaPo) to enhance cooperation efficiency of LLM-driven embodied agents. CaPo first proposes to create a strategic and coherent meta-plan through multi-turn agents discussion before executing any actions. CaPo first proposes creating a strategic and coherent meta-plan through multi-turn discussions among agents before executing any actions. This meta-plan serves as an action guide to efficiently coordinate multiple agents in completing tasks. During the execution phase, agents dynamically adapt the meta-plan to their latest task progress, maintaining the effectiveness of the meta-plan in coordinating agents to complete tasks efficiently. Experimental results on TDW-MAT and C-WAH tasks show the higher task completion rates and efficiency of CaPo compared to state-of-the-arts.

While CaPo significantly improves multi-agent cooperation efficiency, it has limitations, specifically its heavy reliance on LLMs for reasoning and planning during meta-plan generation and adaptation. As shown in Table 1, agents using stronger LLMs like GPT-3.5 outperform those using weaker ones like LLAMA-2. This dependency is a common challenge for LLM-based frameworks like CoELA.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the two aforementioned embodied environments in Sec.B, covering task settings, as well as the observation and action spaces of the agents. Additionally, we present the detailed prompt designs used in our LLMs in Sec.C of the Appendix. Furthermore, we include a section in Appendix Sec. A.3 to demonstrate the reproducibility of our experimental results on the TDW-MAT environments.

REFERENCES

Dhruv Batra, Angel X. Chang, Sonia Chernova, Andrew J. Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, Manolis Savva, and Hao Su. Rearrangement: A challenge for embodied ai, 2020. URL https://arxiv.org/abs/2011.01975.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pp. 1282–1289. PMLR, 2019.

Andrea Dittadi, Frederik Träuble, Manuel Wüthrich, Felix Widmaier, Peter Gehler, Ole Winther, Francesco Locatello, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. The role of pretrained representations for the ood generalization of reinforcement learning agents. *arXiv preprint arXiv:2107.05686*, 2021.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pp. 8657–8677. PMLR, 2023.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel LK Yamins, James J DiCarlo, Josh McDermott, Antonio Torralba, et al. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. In *2022 International conference on robotics and automation (ICRA)*, pp. 8847–8854. IEEE, 2022.

Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L Griffiths, and Mengdi Wang. Embodied llm agents learn to cooperate in organized teams. *arXiv preprint arXiv:2403.12482*, 2024.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pp. 9118–9147. PMLR, 2022a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022b.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. *Advances in neural information processing systems*, 31, 2018.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

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

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. *arXiv preprint arXiv:2310.10701*, 2023b.

Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 6826–6836. PMLR, 2021.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023a.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. Improving text-to-image consistency via automatic prompt optimization. *arXiv preprint arXiv:2403.17804*, 2024.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. *arXiv preprint arXiv:2307.04738*, 2023.

Yohei Nakajima. Babyaig. https://github.com/yoheinakajima/babyagi, 2023.

OpenAI. Gpt-4 technical report, 2024.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2017–2025, 2022.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*, 2020.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. In *International Conference on Learning Representations*, 2021.

Reworkd. Agentgpt. https://github.com/reworkd/AgentGPT, 2023.

Toran Bruce Richards and et al. Auto-gpt: An autonomous gpt-4 experiment. https://github.com/Significant-Gravitas/AutoGPT, 2021.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2998–3009, 2023.

Matthijs TJ Spaan, Geoffrey J Gordon, and Nikos Vlassis. Decentralized planning under uncertainty for teams of communicating agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 249–256, 2006.

J Lukas Thürmer, Frank Wieber, and Peter M Gollwitzer. Planning and performance in small groups: Collective implementation intentions enhance group goal striving. *Frontiers in Psychology*, 8:603, 2017.

Alejandro Torreno, Eva Onaindia, Antonín Komenda, and Michal Štolba. Cooperative multi-agent planning: A survey. *ACM Computing Surveys (CSUR)*, 50(6):1–32, 2017.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

R Tuomela. Collective goals and cooperation. In *Discourse, Interaction and Communication: Proceedings of the Fourth International Colloquium on Cognitive Science (ICCS-95)*, pp. 121–139. Springer, 1998.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):1–26, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. *arXiv preprint arXiv:2111.09189*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yue Wu, So Yeon Min, Yonatan Bisk, Ruslan Salakhutdinov, Amos Azaria, Yuanzhi Li, Tom Mitchell, and Shrimai Prabhumoye. Plan, eliminate, and track–language models are good teachers for embodied agents. *arXiv preprint arXiv:2305.02412*, 2023a.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023b.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*, 2023.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: Building proactive cooperative ai with large language models. *arXiv preprint arXiv:2308.11339*, 2023a.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023b.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.

Dongsheng Zhu, Xunzhu Tang, Weidong Han, Jinghui Lu, Yukun Zhao, Guoliang Xing, Junfeng Wang, and Dawei Yin. Vislinginstruct: Elevating zero-shot learning in multi-modal language models with autonomous instruction optimization. *arXiv preprint arXiv:2402.07398*, 2024.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.