Collaborative Tree Search for Enhancing Embodied Multi-Agent Collaboration

Supplementary Material

The supplementary material is organized as follows:

- Appendix A: Additional Experiments
- Appendix B: Additional Details about CoTS
- Appendix C: Embodied Environments and Baselines
- Appendix D: Planning and Evaluating Examples
- Appendix E: Prompt Templates

A. Additional Experiments

A.1. Collaborating with Humans

Following CoELA, we evaluate human-agent cooperation in C-WAH (symbolic observations). Since CoTS requires one collaborator to act as a planner, this role can be challenging and time-consuming for humans in complex embodied environments. However, LLMs excel in such tasks. Therefore, in this experiment, Alice (the Planner) is controlled by GPT-4, while Bob (the Reflector and Evaluator) is a human. This setup leverages the reasoning capabilities of the language model alongside human experience and precise evaluation.

To assess efficiency, we measure the average number of steps taken. Additionally, eight participants evaluate their teammate based on three key criteria—communication effectiveness, helpfulness, and trust—using a 7-point Likert scale. The results are summarized in the Table 4.

Table 4. Human experiments results.

Method	Ave steps	Comm Effectiveness	Helpfulness	Trust
CoELA	48	5.7/7	5.4/7	6.0/7
CoTS	43	6.3 / 7	6.1 / 7	6.4 / 7

Communication Effectiveness is evaluated based on the following aspects: 1) Clarity of Information Transmission – Whether the provided prompts or explanations are easy to understand, and whether the language is accurate, concise, and well-organized. 2) Understanding User Needs – Whether the Agent can correctly interpret user instructions, questions, or intentions and provide appropriate responses. 3) Overall Interaction Experience – Whether the conversation flows naturally and whether the communication frequency is reasonable. From the experimental results, CoTS demonstrates better "communication effectiveness" mainly because it establishes a detailed long-term collaboration plan, making communication more logical and persuasive while reducing unnecessary exchanges and improving overall communication efficiency.

Helpfulness measures whether the Agent can provide targeted, highly feasible, and productive dialogue content, effectively reducing the user's cognitive and reasoning burden, thereby making task completion easier. This metric focuses on the Agent's actual contribution to user goals and whether users perceive value from the interaction. According to the experimental results, CoTS is primarily responsible for integrating large amounts of information and formulating complex plans, requiring humans only to provide feedback and evaluations, which significantly reduces the user's workload. Additionally, CoTS exhibits higher task execution efficiency, making it easier for users to obtain assistance and improve their productivity.

Trust assesses the degree to which participants trust the Agent's response, including whether the answers provided are accurate and reliable, whether the decision-making process is coherent and well-founded, and whether the information is professional and specific. During the collaboration process, users demonstrated a higher level of trust in CoTS, mainly due to its proactive incorporation of human input during planning and its collaborative approach in multi-round discussions to develop long-term plans. Moreover, because CoTS has well-defined functions for each, its responses are more professional and targeted, making it easier for humans to accept and trust its decisions.

Overall, CoTS outperforms CoELA in communication effectiveness, helpfulness, and trust. Experimental participants generally found CoTS to be more fluid in communication, more helpful, and more trustworthy. The efficiency of experiments conducted with CoTS was also significantly higher than with CoELA, further demonstrating CoTS's potential in the field of human-AI collaboration.

A.2. Cost Discussion

The primary costs for embodied agents come from communication and movement, with movement costs often being higher, as current robots lack the agility of humans, especially in complex and large-scale environments like urban areas. However, through communication, CoTS enables more effective collaborative planning and reduces errors that could lead to critical outcomes in sensitive applications like healthcare or surgery. While this incurs added communication costs (mainly from LLM inference), these are essential to enhance system intelligence, and as LLM technology advances, these costs are likely to diminish. Besides, enhanced planning in CoTS minimizes redundant or erroneous actions by agents, thereby reducing overall task completion time.

Regarding communication cost, we compare CoTS with the traditional tree search method ToT (DFS) on C-WAH (symbolic obs), and the results are shown in Table 5. The communication cost represents the average number of char-

Table 5. Cost of each plan update.

Method	Ave steps	Communication Cost	Time Cost
СоТ	57	-	-
ToT	52	10568 characters	32.4s
CoTS	49	7923 characters	26.8s

acters consumed per round of plan updates. Results show that 1) ToT and CoTS outperform CoT because of their plan search and exploration, but are more costly. 2) CoTS surpasses ToT by 5.8% relative improvement and a big efficiency improvement including a 17% reduction in time cost. 3) Unlike CoTS, ToT lacks selection and backpropagation steps, weakening its value evaluation ability, increasing exploration costs and consuming more memory due to maintaining more nodes. In contrast, CoTS ensures precise long-term planning, minimizing errors and redundant actions, which ultimately reduces overall costs.

B. Additional Details about CoTS

B.1. Plan Reward Based on LLMs

In the Collaboration Tree Search Module and Plan Evaluation Module of CoTS, assessments based on large language models (LLMs) are required to quantify rewards for collaboration plans. (1) In the tree search, the evaluation aims to assign a reward to each newly generated node, guiding the tree toward exploring more promising directions. (2) The evaluation module assesses the viability of current plans based on the progress and actions of embodied agents. If a plan remains viable, it is retained; otherwise, a new action plan is sought.

Utilizing LLMs to assess collaborative plans leverages their formidable reasoning capabilities and comprehensive world knowledge. In the context of Collaborative Tree Search, LLM-based evaluation bypasses the need for intricate simulations by directly providing reward values within the Monte Carlo tree, offering distinct advantages for embodied environments. However, the straightforward application of LLMs for evaluating collaboration plans often yields outcomes that are vague and exhibit a significant lack of consistency with human subjective judgments, marked by considerable randomness.

Specifically, we enhance our prompts with labels to refine the assessment process. For example, here we present three different prompt templates in the planning evaluation module. The label-enhanced version is applied to the plan evaluation module in CoTS (as shown in Fig. 8). If the final reward exceeds the set threshold (*e.g.*, 3), there is no need to update the plan.

Basic Version. Please analyze whether the current collaboration plan requires immediate adjustments. If the current plan is satisfactory, respond with "satisfied"; otherwise, respond with "unsatisfied."

Reward Version. Please analyze whether the current collaboration plan requires immediate adjustments and assign a reward. The higher the reward, the better the current plan and the less it needs updating. The reward should be an integer between 1 and 5.

Our Label Enhanced Version in Plan Evaluation. Please analyze whether the current collaboration plan requires immediate adjustments. Your response should include both your reasoning and a reward. The reward should be assigned based on the following criteria: Plan_Reward: [the reward between 0 and 5: 1, The current plan is completely unreasonable and needs to be revised 2, The current plan requires significant adjustments 3, The current plan needs improvement, its core structure is sound, though details are unclear 4, The current plan requires only minor adjustments 5, The current plan is fully reasonable and needs no changes. Only output the reward.]

The evaluation of plans in collaborative tree search follows a similar pattern. It assesses the rationality of collaborative plans by considering the allocation of agent actions and whether the action distances between agents are accounted for. After evaluation, the rewards from these two aspects are aggregated and normalized to a range between 0 and 1, serving as the final reward for the node.

Dis_Reward. [the reward between 0 and 5: 1, No consideration of distance 2, Minimal consideration of distance, overlooking key factors 3, Distance is considered but not entirely accurate 4, Distance is sufficiently considered, with only minor oversights 5, Comprehensive and accurate consideration of distance. Only output the.]

Task_Reward. [the reward between 0 and 5: 1, No consideration of work distribution between Alice and Bob 2, Minimal consideration of work distribution, leading to unreasonable allocation 3, Work distribution is considered but not entirely accurate 4, Work distribution is reasonable, with only minor oversights 5, Work distribution is highly effective, making full use of Alice and Bob's abilities. Only output the reward.]

B.2. Monte Carlo Tree Search

MCTS is a technique for identifying optimal decisions within a defined decision space by randomly sampling and constructing a search tree based on these outcomes. It has profoundly influenced artificial intelligence (AI), particularly in game strategies and planning challenges. Monte Carlo trees are constructed incrementally and asymmetrically, with each node in the tree representing a specific state. During each iteration of the algorithm, the child state with the highest UCT (Upper Confidence Bound for Trees) value is selected for further expansion, balancing exploration (searching under-sampled areas) and exploitation (focusing on promising areas). Upon the conclusion of



Input: [Goal] [Current plan] [Dialogue history] [Previous action] [Progress desc] Please help me analyze whether the current collaboration plan needs to be adjusted immediately. Your response needs to include your thought process and rewarding.

[the reward between 0 and 5: 1, The current plan is completely unreasonable and needs to be revised 2, The current plan requires significant adjustments 3, The current plan needs improvement, its core structure is sound, though details are unclear 4, The current plan requires only minor adjustments 5, The current plan is fully reasonable and needs no changes. Only output the reward.]

LLM Output: 3

Figure 8. An example of using LLMs to assess whether the collaboration plan need to be updated.



Figure 9. Monte Carlo Tree Search [4].

an episode, backpropagation is performed, as illustrated in Fig. 9. MCTS consists of four essential stages:

(1) Selection: Starting from the root node, the algorithm identifies the most promising node according to the tree policy, continuing this process until a leaf node is reached.

(2) Expansion: At the selected leaf node, if it is not a terminal state, new child nodes are added to represent potential future states.

(3) Simulation: From the newly added nodes, the algorithm conducts random simulations until reaching a terminal state, then evaluates the node's potential based on the outcomes.

(4) Backpropagation: Simulation results are propagated back to the root, updating the scores of all visited nodes along the way.

The effectiveness of MCTS depends heavily on its ability to balance exploring unknown areas with exploiting known information, making it particularly valuable in domains requiring precise and adaptive decision-making.

C. Supplementary Introduction to Embodied Environments and Contrastive Methods

C.1. Embodied Environments.

Following CoELA [43], this paper considers two extremely challenging embodied-world multi-agent environments to test the performance of EMCT, including a subset of 24 tasks in TDW-MAT and a subset of 10 tasks in C-WAH.

1) **TDW-MAT** (**ThreeD World Multi-Agent Transport**), an extension of the ThreeD World Transport Challenge, TDW-MAT simulates collaborative transportation tasks in a physics-enabled household environment, built on the TDW platform. In this environment, agents must explore a house to locate dispersed objects, pick them up, and deliver them to designated locations. Each agent can carry only one item per hand, but they can strategically use containers to enhance transportation efficiency.

This study selects 24 representative subtasks, evenly split between food transportation (12 tasks) and item transportation (12 tasks). Each subtask environment contains 10 target objects and 2–5 containers, with a maximum duration of 3,000 action frames. Agents are tasked with transporting as many target objects as possible to the goal position, utilizing containers as tools. While a container can hold up to three objects, an agent without a container is limited to carrying two objects at a time.

Agents retain access to their ego-centric visual observations and action spaces throughout the tasks. Communication among agents, mediated by the LLM, occurs exclusively through natural language, leveraging the LLM's advanced reasoning and comprehension capabilities.

The food-transporting task involves six types of targets (apple, banana, orange, bread, loaf bread, and burger) and three types of containers (bowl, plate, and tea tray). In contrast, the object-transporting task includes six different targets (calculator, mouse, pen, lighter, purse, and iPhone) with three container types (plastic basket, wooden basket, and wicker basket). In each task, there are 10 target objects and 2 to 5 containers in total. Objects are distributed across four types of rooms—living room, office, kitchen, and bedroom—arranged in alignment with common-sense placement.

Observation Space The embodied agent primarily receives an egocentric RGB image and a depth image, along with several auxiliary observations. The observation space details are as follows:

- **RGB Image**: An egocentric image captured by a forward-facing camera, with a resolution of 512×512 and a 90-degree field of view.
- **Depth Image**: A depth map captured using the same camera parameters as the RGB image.
- Oracle Perception (optional): A color-mapped image where each object ID corresponds to a unique color, using the same intrinsic camera parameters as the RGB image.
- Agent Position and Rotation: The agent's position and orientation within the simulated environment.
- Held Objects: Information about objects currently in the agent's possession.
- **Opponent Held Objects**: Enumerates the objects held by another agent, provided that this agent is visible.
- Messages: Communications exchanged among agents.

Action Space In TDW-MAT, agents can perform seven types of actions for environmental interaction or inter-agent communication. Each action spans multiple frames, with specifics listed below:

- Move Forward: Advances the agent by 0.5 meters.
- Turn Left: Rotates the agent 15 degrees to the left.
- Turn Right: Rotates the agent 15 degrees to the right.
- **Grasp**: Allows the agent to grasp an object if it is within close proximity. The object can be a target or container.
- **Put In**: Enables the agent to place a target object into a container, provided the agent is holding the target and container simultaneously.
- Drop: Releases any objects held by the agent.
- Send Message: Transmits a message to other agents, with a limit of 500 characters per frame.

We compile a comprehensive Action List encompassing all available actions relevant to the current state, providing LLMs with a structured framework for action selection. These selectable actions can be conveniently executed by embodied agents. For example, the TDW-MAT (Three-Dimensional World Multi-Agent Task) includes the following sub-plans:

- · explore current room
- go to the xx room
- go grasp the xxx (target object or container)
- put holding objects into the holding container
- transport holding objects to the bed

2) C-WAH (Communicative Watch-And-Help) builds

upon the Watch-And-Help challenge and is implemented on the VirtualHome-Social multi-agent simulation platform, introducing inter-agent communication. The environment encompasses five common household scenarios: afternoon tea preparation, dish washing, meal preparation, grocery organization, and table setting. Each task comprises multiple subgoals defined by predicates, such as "ON/IN(x,y)" denoting the placement of object x on/in location y. Detailed task descriptions and subgoal listings are provided in Table 6. Our test set consists of 10 tasks, with two tasks selected from each of the five scenario types. The task goal is to meet all given sub-goals within 250 time steps, the faster the better.

C-WAH encompasses a diverse range of household tasks, each with specific objectives: preparing afternoon tea, dishwashing, meal preparation, grocery storing, and dinner table arrangement.

Objective. Agents are required to complete all specified subgoals within a 250-timestep limit. Tasks consist of three to five subgoals, challenging agents to demonstrate effective strategic planning and inter-agent communication.

Observation Space. CWAH encompasses two modalities: *Symbolic Observation* and *Visual Observation*. In the *Symbolic Observation* mode, aligned with the setup in the foundational Watch-And-Help challenge, an agent accesses extensive object-related data within its vicinity, encompassing attributes such as location, status, name, and relational context. In contrast, *Visual Observation* permits agents to capture egocentric RGB and depth visuals, alongside supplementary observational data. Specifics of the observation space are as follows:

- **RGB Image**: Captured through a forward-oriented camera, presenting a resolution of 256×512 and a 60-degree viewing angle.
- **Depth Image**: Utilizes the same intrinsic parameters as the RGB camera to provide depth cues.
- **Oracle Perception**: This visualization assigns unique colors to each object identifier, consistent with the RGB camera's intrinsic settings.
- Agent Position: Reflects the agent's location within the simulated environment.
- Messages: Includes all communications disseminated by the agents.

Action Space. Expanding on the established Watch-And-Help Challenge, the action repertoire includes a novel messaging capability. Enumerated actions are:

- Walk Towards: Navigate towards an object or a different room.
- Turn Left: Execute a 30-degree left turn.
- Turn Right: Execute a 30-degree right turn.
- **Grasp**: Ability to grasp an object, contingent on proximity.
- Open: Capability to open a nearby closed container.

Task Name	Predicate Set	
Prepare afternoon tea	ON(cupcake,coffeetable), ON(pudding,coffeetable),	
	ON(apple,coffeetable), ON(juice,coffeetable), ON(wine,coffeetable)	
Wash dishes	IN(plate,dishwasher), IN(fork,dishwasher)	
Prepare a meal	ON(coffeepot,dinnertable), ON(cupcake,dinnertable),	
	ON(pancake,dinnertable), ON(pudding,dinnertable),	
	ON(apple,dinnertable), ON(juice,dinnertable), ON(wine,dinnertable)	
Put groceries	IN(cupcake,fridge), IN(pancake,fridge),	
	IN(pudding,fridge), IN(apple,fridge), IN(juice,fridge),	
	IN(wine,fridge)	
Set up a dinner table	ON(plate,dinnertable), ON(fork,dinnertable)	

Table 6. Task description in C-WAH. The tasks are divided into five types, each containing several predicates.

- Close: Ability to close an open container when in proximity.
- **Put**: Place objects into an open container or onto a surface if nearby.
- Send Message: Transmit a message to peers, limited to 500 characters per instance.

C.2. Baselines

Here we introduce in detail the baseline methods included in the experiment:

1) RHP (Rule-based Hierarchical Planner) [8], which builds upon the existing robust baseline from the ThreeD-World traffic challenge. RHP introduces a Rule-Based Hierarchical Planning framework enhanced by a Frontier Exploration technique. It consists of a rule-driven high-level planner that selects among multiple strategic options based on predefined rules. Navigation is handled by an A*-based pathfinding planner, utilizing both occupancy and semantic maps that are continuously updated based on visual inputs.

2) MHP (MCTS-based Hierarchical Planner) [18], which represents the strongest baseline in the Watch-and-Help Challenge. It features a Hierarchical Planner combining a high-level planner powered by Monte Carlo Tree Search (MCTS) with a low-level planner based on Regression Planning (RP). This Modular Hierarchical Planner dynamically infers other agents' intentions and adjusts its subgoals accordingly, based on their observed actions.

3) MAT (Multi-Agent Transformer) [39], which is a multi-agent reinforcement learning model. We follows the MAT model trained in CoELA. It conceptualizes MARL as a sequence modeling challenge, employing a centralized decision transformer to orchestrate action generation. The input for the MAT is bifurcated. The primary segment encompasses a top-down semantic map of dimensions (12, 24), derived from oracle perception. The secondary input segment incorporates agent-specific information, such as container possession status and the count of held objects.

4) CoELA (Cooperative Embodied Language Agent) [43] is an advanced collaborative embodied agent that seamlessly integrates perception, memory, and execution for planning, communication, and collaboration. By harnessing the vast world knowledge and superior reasoning capabilities of LLMs, along with their expertise in natural language processing, CoELA effectively coordinates with other agents to tackle complex, real-world tasks.

5) RoCo (Dialectic Multi-Robot Collaboration) [24] is an embodied collaborative agent that utilizes LLMs for both high-level communication and low-level path planning. It facilitates agent collaboration through natural language dialogues. Using multi-round interactions, agents share mutual information and provide feedback. We implement RoCo in TDW-MAT and CWAH environments, where agents determine a unified collaboration plan through dialogue. Based on this plan, agents take actions, and any new progress triggers further discussions to refine the plan.

6) CaPo (Cooperative Plan Optimization) [22] is similar to RoCo, as both discuss cooperative plans through dialogue before executing them. CaPo consists of two key phases: meta-plan generation and meta-plan updating. During the meta-plan generation phase, agents exchange information to determine the plan, while in the meta-plan updating phase, the plan is revised when new items are discovered or sub-tasks are completed.

D. Examples

In this section, we present several examples of CoTS in experiments. First, Fig. 10 provides detailed node information for a specific path within the collaborative tree. Then, Fig. 11 presents two concrete examples from the plan evaluation module, representing cases where the current plan needs to be updated and where it does not.

E. Prompt Templates

We have listed the following templates: Alice Prompt (Fig. 12), Bob Prompt (Fig. 13), Plan Evaluation Prompt (Fig. 14), and Plan Parsing Prompt (Fig. 15).

Task: transport 2 iphones, 2 purses, 2 mouses, 2 lighters, 1 pen, 1 calculator to the bed

The Process in Each Node: Alice makes plans, Bob provides suggestions and rewards.

Plan from Alice: Step 1: Alice grasps the <mouse> (9068700). Bob explore his current room. Step 2: Alice explore the <Livingroom> (2000). If they find any target items, transport them to the bed.

Message to Bob: Hi Bob, I am currently in <Livingroom> (4000) where I found a <mouse> (9068700) which I plan to grasp. As I am not aware of your location or progress, please let me know if you have any suggestions or find any target objects or containers. Let's focus on exploring the rooms we are currently in and grasp any found target objects.

Message to Alice: Hi Alice, I am currently in the <Livingroom> (4000) and have found an <iphone> (14360913). It would be more efficient if I continue to explore the <Livingroom> (4000) while you explore a new room. Let's avoid exploring the same room together and utilize our abilities more efficiently. Can we adjust the plan accordingly?**Reward: 0.4**

Plan from Alice: Step 1: Alice moves to the <Bedroom> (8000). Bob grasps the <iphone> (14360913). Step 2: Alice explore the <Bedroom> (8000). Bob explores the <Livingroom> (4000). Step 3: If they find any target items, transport them to the bed.

Message to Bob: Hi Bob, thanks for the update. I will now shift to the <Bedroom> (8000) to avoid duplication of our efforts. Please continue exploring <Livingroom> (4000) and grasp the iphone> (14360913) you found. Let's keep each other updated on our progress.

Message to Alice: Hi Alice, I think it would be more efficient to split up and explore different rooms. While I continue to explore the <Livingroom> (4000) and retrieve the <iphone> (14360913), you could head to the nearest room, the <Livingroom> (2000), to explore and look for more target objects. I suggest you grasp the <mouse> (9068700) in your current room before moving to <Livingroom> (2000). This way, we can cover more ground and find the target objects more quickly? **Reward: 0.7**

Plan from Alice: Step 1: Alice grasps the <mouse> (9068700). Bob grasps the iphone> (14360913).Step 2: Alice
moves to the <Livingroom> (2000). Bob explores the <Livingroom> (4000).Step 3: Alice explores the <Livingroom> (2000). If they find any target items, transport them to the bed.

Message to Bob: Hi Bob, I agree with you. I will move to the <Livingroom> (2000) to explore while you continue in the <Livingroom> (4000). This should help us cover more ground quickly. Please keep me updated on your progress.

Message to Alice: Hi Alice, your plan looks good. I will grasp the *iphone* (14360913) in my current room and continue exploring for other target objects or containers. If I find another object or container, I will grasp it and transport to thebed if I am holding two objects. **Reward: 0.9**

Figure 10. An example of node information in a path of the collaboration tree, selected from the experiment on TDW-MAT.

An Example of the Current Plan That Do Not Needs Updating

Current Plan

Step 1: Alice go grasp <orange> (14249767) and the container <tea_tray> (16224318). Bob continues to explore the Kitchen (5000).

Step 2: Alice put <orange> (14249767) into the container <tea_tray> (16224318). If Bob finds any target objects in the Kitchen (5000), he should go grasp up to two of them.

Step 3: Alice transport the container <tea_tray> (16224318) with the <orange> to the bed. If Bob is holding any target objects, he should transport them to the bed. If not, Bob should continue to explore the Kitchen (5000).

Plan Evaluation Reasoning

According to the latest progress, Alice and Bob are following the action plan. Alice has found the target object <orange> (14249767) and containers <tea_tray> (16224318), <plate> (1457820) in the <Livingroom> (4000) as planned. Bob is in the <Kitchen> (5000) and exploring. As per the plan, Alice needs to grasp the <orange> and one container, put the <orange> into the container, and then transport them to the bed. Bob is supposed to continue exploring the Kitchen and grasp target objects if found. No new target objects have been found by Bob and Alice is yet to execute the planned action. Therefore, there is no need to change the action plan immediately. Reward: [5].

Current Plan

An Example of the Current Plan That Needs Updating

Step 1: Alice go to the <Kitchen> (2000), Bob continues to explore the <Livingroom> (3000). Step 2: Alice explore the <Kitchen> (2000), Bob go grasp <orange> (7175626). Step 3: Alice go grasp up to two objects if found, Bob transport holding object to the bed.

Plan Evaluation Reasoning

According to the latest progress, Alice is in the Bedroom (6000) and has found a loaf of bread (13469629), which is one of the target objects, but she hasn't acted to grasp it yet. Meanwhile, Bob is in the Livingroom (3000) and has found an orange (7175626), a target object, but he hasn't acted to grasp it either. The current action plan is for Alice to proceed to the Kitchen (2000) and Bob to grasp the orange (7175626). However, since Alice has already found a loaf of bread (13469629) in Bedroom (6000), it would be more efficient for her to grasp that first before moving to another room. Similarly, Bob should also act to grasp the orange (7175626) he has found before continuing to explore. Therefore, the current action plan could be improved to be more efficient. Reward: [2].

Figure 11. Examples in the plan evaluation module.

I am Alice. My teammate Bob and I want to transport as many target objects as possible to the bed with the help of containers within 3000 steps. Remember Alice or Bob each can hold two things at a time and the 2 objects can be objects or containers. Alice or Bob can grasp containers and put objects into them to hold more objects at a time. Assume Alice is an expert in designing plan outlines. Given our shared goal, previous plan, dialogue history, latest progress, Bob's suggestion, please help me generate/refine the global plan for Bob and me during task execution, guiding us to achieve the goal collaboratively as soon as possible. Note that a container can hold three objects and will be lost once transported to the bed. I can only place objects into a container I am holding after grasping it. All objects are denoted as <name>(id), such as (712). Actions take several steps to complete. Note that it may be costly to go to another room or transport to the bed, use these actions sparingly.

The generated collaboration plan should strictly meet following requirements:

1. You should make full use of the agent's transport capabilities (up to two objects or containers per agent) to transport as much as possible efficiently. The collaboration plan should reasonably arrange the division of action between Alice and Bob in order to achieve the goal as soon as possible. Please be as detailed as possible in assigning actions to each agent, specific to a room or item.

2. Please keep your reasoning process, but the final collaboration plan should be brief, reliable, authentic, and consistent with the latest progress of Alice and Bob. Don't make random and meaningless plans.

3. There are only 5 allowed actions you can use to construct the collaboration plan. 1) 'go to': move to a specified room. 2) 'explore': explore a room for underlying target objects. 3) ' go grasp': go to grasp a specified target object. 4) 'put': Place an object into a specified container. 5) 'transport': Transport holding objects or containers to the bed and drop them on the bed.

4. The collaboration plan should be detailed to each Agent. The collaboration plan only needs to consider three steps at most at current time. When there is not much known information or content to be planned, the collaboration plan can have only one or two steps. The collaboration plan must be structured strictly in the format: {Action Plan: Step 1: Alice xxx, Bob xxx; Step 2: Alice xxx, Bob xxx; Step 3: Alice xxx, Bob xxx}. 'xxx' represents one or multiple allowed actions.

5. In order to let Bob know about Alice's situation, you need to generate a short message to Bob. The message has to be concise, reliable, and helpful for assisting Bob and Alice to make an efficient and consistent collaboration plan, and transport as many objects to the bed as possible. Don't generate repetitive messages.

6. Alice and Bob act separately and can only exchange information and they cannot exchange items.

Here is an example for Alice:

{Goal: [Transport 2 oranges, 3 apples, 1 banana, 3 breads, 1 burger to the bed.]

Reasoning: [According to Bob's suggestions and progress, the current plan is partially reasonable and needs minor adjustments. Bob found <orange> (5345043) and
banana> (6381322) in <Livingroom> (4000), but he can't take more things in his hand. He needs to transport them to the bed first. Alice currently is not far from <Livingroom> (4000), so Alice can go to <Livingroom> (4000) to get <orange> (5345043) and

banana> (6381322).]

Collaboration plan: [Step 1: Alice go to the <Livingroom> (4000). Bob transport holding objects <bread> (547795) and

 (521494) to the bed and drop them on the bed.

Step 2: Alice go grasp <orange> (5345043) and
shanana> (6381322). Bob goes to the <Kitchen> (5000) to explore. Step 3: Alice transports holding objects to the bed. If any target objects or containers are found by Bob, Bob go grasp objects.]

Message: [Hi Bob, I have received your message and will adjust our collaboration plan. You continue to transport the target items in your hand to the bed. I will transport the <orange> (5345043) and
banana> (6381322) in <Livingroom> (4000) you found. Do you have any new suggestions for the updated plan?]

Following are provided information for Alice: Goal: \$GOAL\$ Previous Collaboration Plan: \$PREVIOUS_PLAN\$ Dialogue History: \$DIALOGUE_HISTORY\$ Alice's Progress: \$ALICE_PROGRESS\$, it's need to consider Alice's latest action progress.

Think step by step, and generate the response:

Figure 12. Alice Prompt.

I am Bob. My teammate Alice and I want to transport as many target objects as possible to the bed with the help of containers within 3000 steps. Remember Alice or Bob can hold two things at a time, and they can be objects or containers. Alice or Bob can grasp containers and put objects into them to hold more objects at a time. Because Alice may not understand Bob's current progress and information, and may not consider the plan comprehensively and perfectly, which wastes our action time. Given our shared goal, collaboration plan, dialogue history, progress, and my previous actions, please help me analyze and reward Alice's proposed collaboration plan, point out the shortcomings of Alice's plan and reflect on it and finally generate a message to send to Alice, at the beginning of the message, I should first explain my findings. Alice and Bob actseparately and they cannot exchange items. It may be costly to go to another room or transport to the bed, use these actions sparingly.

The content Bob generate mainly consists of two parts: reasoning and message sent to Alice. Please strictly follow the following format:

Reasoning: [the reasoning process, analyze the unreasonableness of the current plan and consider how to make it more efficient]

Dis_Reward: [the reward between 0 and 5: 1, No consideration of distance 2, Minimal consideration of distance, overlooking key factors 3, Distance is considered but not entirely accurate 4, Distance is sufficiently considered, with only minor oversights 5, Comprehensive and accurate consideration of distance. Only output the reward.] Task_Reward: [the reward between 0 and 5: 1, No consideration of work distribution between Alice and Bob 2, Minimal consideration of work distribution, leading to unreasonable allocation 3, Work distribution is considered but not entirely accurate 4, Work distribution is reasonable, with only minor oversights 5, Work distribution is highly effective, making full use of Alice and Bob's abilities. Only output the reward.] Message: [the message sent to Alice, you need to first tell Alice about your findings]

Here is an example for your reference: :

Reasoning: [In the current plan, Alice may not know our progress, so some of Bob's plans are vague, which needs to be improved. In addition, it is a waste of time for Alice and Bob to explore the same room together.] Dis_Reward: [4]

Task_Reward: [2]

Message: [Hi, Alice, I dont find any objects and we are in the same room, it is not efficient for us to explore the same room together. Your plan needs to be adjusted and describe my actions as detailed as possible.]

The following is the information of Bob currently:

Bob's Previous Action: \$ACTION_HISTORY\$ Bob's Progress: \$BOB_PROGRESS\$, it's need to consider Bob's latestprogress and the distance is a dimensionless relative measure.

The following is the relevant information when Alice is planning her collaboration plan, which can be used as a reference for Bob:

User: {user input}\n

Alice's response: {candidate_content}

Think step by step, and generate the content sent to Alice:

Figure 13. Bob Prompt.

I am Bob. My teammate Alice and I want to transport as many target objects as possible to the bed with the help of containers within 3000 steps. Important: Alice or Bob can hold two things at a time, and they can be objects or containers. I can grasp containers and put objects into them to hold more objects at a time. Note that a container can contain three objects, and will be lost once transported to the bed. I can only put objects into the container I hold after grasping it. All objects are denoted as <name> (id), such as (712). Actions take several steps to finish. Before, Alice has generated an collaboration plan based on our progress and discussion to coordinate Alice and Bob to complete the task efficiently. Now some steps have passed since the last discussion of the collaboration plan, given our shared goal, collaboration plan, dialogue history, progress, and our progress can only reflect whether the action has been completed, but not whether the action is being taken.

Your response needs to include your thought process and scoring. Scoring is mainly based on the following criteria: PA_Reward: [the reward between 0 and 5: 1, The current plan is completely unreasonable and needs to be revised 2, The current plan requires significant adjustments 3, The current plan needs improvement, its core structure is sound, though details are unclear 4, The current plan requires only minor adjustments 5, The current plan is fully reasonable and needs no changes. Only output the reward.]

Here are two examples for you:

Reasoning: According to the latest progress, There is currently no plan to execute, so a plan needs to be generated. Plan_Reward: [1]

Reasoning: According to the latest progress, Bob and Alice are both acting according to plan. Alice is in the process of taking action and Bob found new target items <ipod> (2189332) in the room <Office> (3000), which was not considered in the previous collaboration plan. We should update our collaboration plan to take them into account. Plan_Reward: [3]

Following are provided information for you:

Goal: \$GOAL\$ Collaboration Plan: \$ACTION_PLAN\$ Dialogue History: \$DIALOGUE_HISTORY\$ Previous Action: \$ACTION_HISTORY\$ Alice and Bob's Progress Desc: \$PROGRESS\$, it's important to consider Alice and Bob's latest action progress and the distance is a dimensionless relative measure.

Think step by step, and generate the message:

Figure 14. Plan Evaluation Prompt.

I am \$AGENT_NAME\$. My teammate \$OPP_NAME\$ and I want to transport as many target objects as possible to the bed with the help of containers within 3000 steps. I can hold two things at a time, and they can be objects or containers. I can grasp containers and put objects into them to hold more objects at a time. All objects and rooms are denoted as <name>(id), <Living room> (3000) and <Living room> (1000) are not the same room.

Assume that you are an expert decision maker. Given our shared goal, collaboration plan, my progress, and previous actions, please help \$AGENT_NAME\$ choose adjustments and select my next available action. Note that a container can contain three objects, and will be lost once transported to the bed. I can only put objects into the container I hold after grasping it. Actions take several steps to finish. It may be costly to go to another room or transport to the bed, use these actions sparingly.

The collaboration plan has the highest priority, if the number of steps is close to 3000, it is most important to get the things in hand or the nearest things to the bed as soon as possible, rather than continue to look for.

Because I need to complete the task step by step, you only need to help me to choose the next step that needs to be taken. Your response must be one of the Available actions.

Goal: \$GOAL\$ Collaboration Plan: \$ACTION_PLAN\$ Available Actions: \$AVAILABLE_ACTIONS\$ \$AGENT_NAME\$ Previous Action: \$ACTION_HISTORY1\$ \$OPPO_NAME\$ Previous Action: \$ACTION_HISTORY2\$ Progress: \$PROGRESS\$

Think step by step, and choose the next step of \$AGENT_NAME\$:

Figure 15. Plan Parsing Prompt.