# Embodied Agents Meet Personalization: Exploring Memory Utilization for Personalized Assistance

**Taeyoon Kwon**[*1] **Dongwook Choi**[*1] **Sunghwan Kim**[1] **Hyojun Kim**[1]
**Seungjun Moon**[1] **Beong-woo Kwak**[1] **Kuan-Hao Huang**[2] **Jinyoung Yeo**[1]
[1]Yonsei University [2]Texas A&M University

## Abstract

Embodied agents empowered by large language models (LLMs) have shown strong performance in household object rearrangement tasks. However, these tasks primarily focus on single-turn interactions with simplified instructions, which do not truly reflect the challenges of providing meaningful assistance to users. To provide personalized assistance, embodied agents must understand the unique semantics that users assign to the physical world (*e.g.*, favorite cup, breakfast routine) by leveraging prior interaction history to interpret dynamic, real-world instructions. Yet, the effectiveness of embodied agents in utilizing memory for personalized assistance remains largely underexplored. To address this gap, we present MEMENTO, a personalized embodied agent evaluation framework designed to comprehensively assess memory utilization capabilities to provide personalized assistance. Our framework consists of a two-stage memory evaluation process design that enables quantifying the impact of memory utilization on task performance. This process enables the evaluation of agents' understanding of personalized knowledge in object rearrangement tasks by focusing on its role in goal interpretation: (1) the ability to identify target objects based on personal meaning (*object semantics*), and (2) the ability to infer object–location configurations from consistent user patterns, such as routines (*user patterns*). Our experiments across various LLMs reveal significant limitations in memory utilization, with even frontier models like GPT-4o experiencing a 30.5% performance drop when required to reference multiple memories, particularly in tasks involving user patterns. These findings, along with our detailed analyses and case studies, provide valuable insights for future research in developing more effective personalized embodied agents. Our code and data is available at `https://anonymous.4open.science/r/MEMENTO`.

## 1 Introduction

Embodied agents empowered by large language models (LLMs) have recently demonstrated remarkable success in executing object rearrangement tasks in household environments [18, 49, 44, 7, 28]. As the primary objective of embodied agents is to provide assistance to users while interacting with the physical world, leveraging LLMs' natural language understanding and reasoning capabilities lead embodied agents to effectively interpret user instructions into sets of target object-location pairs that the agent should rearrange to successfully accomplish the task.

But do such tasks truly reflect the challenges in providing meaningful assistance to the users? As illustrated in Figure 1, conventional embodied tasks predominantly focus on single-turn interactions with static and simplified instructions that the agents could simply follow without implicit reasoning

[*]Equal contribution

Preprint. Under review.


Figure 1: Comparison between traditional embodied tasks and personalized assistance tasks. Previous works focus on strictly following simple instructions, while personalized assistance agents must know user-specific knowledge, which require grounding in past interactions. This highlights the challenge of going beyond instruction-following toward context-aware personalized embodied agent.

to comprehend user intentions [2, 17, 44, 50]. However, for personalized embodied agents, it is important to understand personalized knowledge that users assign unique semantics to the physical world (*e.g.*, favorite cup, breakfast routine) to interpret dynamic instructions. To provide personalized assistance, agents must effectively leverage memories that retain personalized knowledge from previous interactions—especially episodic memory, which enables the recall of specific events grounded in time and space [20]. Without such memory utilization, embodied agents require users to repeatedly provide detailed instructions, which may hinder user engagement and prevent natural human-agent interaction. Despite its importance, the effectiveness of embodied agents in utilizing episodic memory containing personalized knowledge remains largely underexplored.

In this work, we present MEMENTO, a personalized embodied agent evaluation framework designed for comprehensive assessment of memory utilization for providing personalized assistance. To enable a thorough analysis of memory utilization, we divide the memory evaluation process into two stages. In the **Memory Acquisition Stage**, agents perform tasks with instructions containing personalized knowledge while accumulating the interaction history. Subsequently, the **Memory Utilization Stage** challenges agents to complete the same tasks as in the memory acquisition stage but with modified instructions that is difficult to succeed without referencing the previously acquired personalized knowledge. This design allows us to systematically quantify the impact of memory utilization on task performance. Building upon this evaluation process, we aim to analyze agents' ability to understand personalized knowledge in object rearrangement by focusing on its role in goal interpretation: (1) the ability to identify target objects based on personal meaning (*object semantics*), and (2) the ability to infer object–location configurations from consistent user patterns, such as routines (*user patterns*).

Based on MEMENTO, we evaluate embodied agents powered by a range of LLMs with varying capabilities, covering both open-source and proprietary models. Our findings reveal that even frontier LLMs struggle to utilize episodic memory with personalized knowledge, with GPT-4o exhibiting a 30.5% performance drop when required to reference multiple memories. Further analysis shows that this performance degradation is particularly pronounced in tasks involving user patterns, and that the agents are highly susceptible to irrelevant memories acting as distractors. We also conduct error and success case analyses to understand how embodied agents reference memories during task execution, providing valuable insights to guide future research in developing personalized embodied agents.

To summarize, our contributions are as follows:

- We propose MEMENTO, a novel personalized embodied agent evaluation framework designed to assess agents' ability to utilize episodic memory for providing personalized assistance in object rearrangement tasks.
- To quantify the impact of memory and analyze understanding of personalized knowledge independently of reasoning capabilities, we decompose memory usage into two stages: Memory Acquisition and Memory Utilization.

- Through extensive experiments and analysis, we identify key limitations of current LLM-powered embodied agents in leveraging personalized knowledge from memory, and offer insights to guide future research on personalized embodied agents.

## 2 Related Work

**LLM-powered embodied agents.** LLMs have significantly advanced embodied agents' reasoning and planning capabilities in recent years. Researchers have explored LLMs for interpreting user goals [2], high-level task planning [17], and integrating LLMs into comprehensive embodied agent frameworks [18, 29, 34, 44, 19]. Other research directions have focused on generating executable code for embodied tasks directly from language instructions [30, 47, 43], while various benchmarks have been developed to evaluate embodied reasoning abilities [26, 9, 7, 28]. Collectively, these studies highlight the promise of LLM-powered agents in bridging language understanding and physical interaction.

**Memory systems for embodied agents.** Previous studies on memory systems for embodied agents have primarily focused on semantic memory (*e.g.*, scene graph, semantic map), which store and provide state information about the current environment [39, 23, 16, 51, 48], or on procedural memory (*e.g.*, skill library) that stores action primitives, focusing on how to perform tasks to enhance the efficiency on generating the low-level action code [47, 42, 59]. Another important category is *episodic memory*, which captures specific past interactions and experiences with users. However, prior uses of episodic memory have mostly treated it as passive task buffers [2, 43] or histories for in-context [18, 31, 44, 8], without explicitly evaluating its role in personalized task grounding or systematic memory utilization.

**Personalization for embodied agents.** The importance of personalization in robotics has long been recognized [13, 25, 10], particularly in the context of human-robot interaction where robots adapt their interactive behaviors to align with individual users. Recent works have focused on reflecting individual user's preferences during embodied agents' task execution, such as spatial arrangement [22, 49], table settings [37], or personalized object navigations [11, 4]. Recently, Xu et al. [52] aim to infer user preferences from a few demonstrations and adapt planning behavior accordingly. However, these approaches primarily focus on implicit preference adaptation or short-term reactive behaviors, without modeling user-specific knowledge in a structured manner.

## 3 Preliminaries

We formulate the object rearrangement task for LLM-powered embodied agents as a Partially Observable Markov Decision Process (POMDP) defined by the tuple $(S, A, T, R, \Omega, O, \gamma)$, where $S$ denotes the set of environment states, $A$ is the set of actions, $T$ is the transition function, $R$ is the reward function, $\Omega$ is the observation space, $O$ is the observation function, and $\gamma$ the discount factor. At each timestep $t$, the environment is in a state $s_t \in S$, and the agent receives a partial observation $w_t \in \Omega$ in text modality describing visible objects near its current position. At the beginning of each episode, the agent is given a natural language instruction $I$ (*e.g.*, Place the mug on the table and the book on the shelf). The instruction is grounded into a symbolic representation of the ground-truth goal $g$, denoted $g = \{(o_i, l_i)\}_{i=1}^{k}$, where each pair $(o_i, l_i)$ are the target object $o_i$ (*e.g.*, mug) that should be placed at the specified location $l_i$ (*e.g.*, on the table). In order to execute the instruction, the agent must internally derive the goal representation $\phi(I) \longrightarrow g$, where $\phi$ denotes the instruction grounding function, to guide the policy's decision-making. Given an instruction $I$, where the policy $\pi$ is implemented by an LLM, the agent generates actions at timestep $t$ based on the trajectory of observations and actions:

$$\pi(I, \tau_t) \rightarrow a_t, \quad \tau_t = (w_1, a_1, w_2, a_2, \ldots, w_{t-1}, a_{t-1}, w_t) \tag{1}$$

The goal is to produce a sequence of actions $a_{1:t} = (a_1, a_2, \ldots, a_t)$ such that the resulting state $s_t$ satisfies the agent's goal $g(s_t) = 1$.

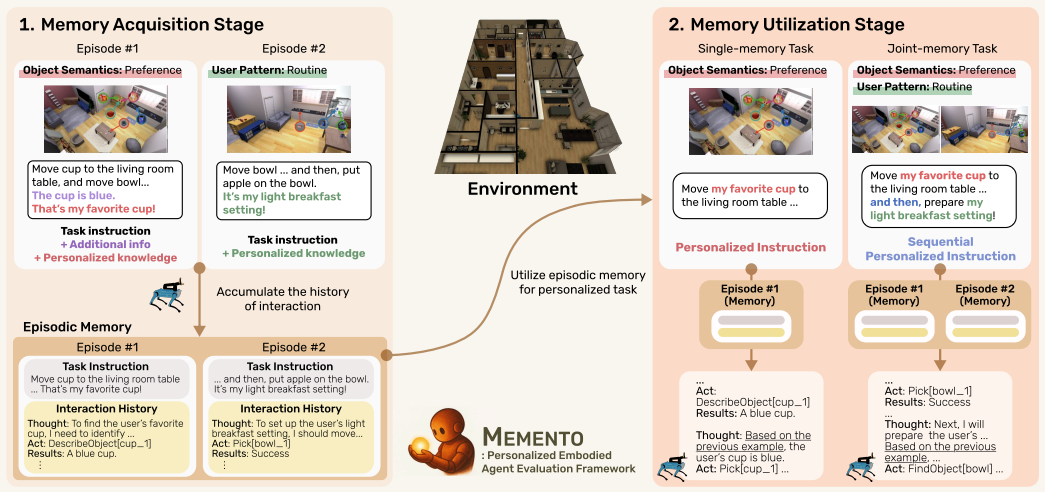


Figure 2: Overview of MEMENTO.

# 4 MEMENTO

In this section, we introduce MEMENTO, a personalized embodied agent evaluation framework designed to assess how well embodied agents leverage episodic memory containing personalized knowledge to provide personalized assistance. We begin by describing how we design the memory evaluation process (§ 4.1), categorize personalized knowledge (§ 4.2), the data construction process (§ 4.3), and the validation of our evaluation framework (§ 4.4).

## 4.1 Memory Evaluation Process Design

**Two-stage evaluation process.** The major challenge of evaluating embodied agents' memory utilization capability is to quantify the memory effect on the overall task performance. To address this limitation, as shown in Figure 2, we divide the evaluation process into two stages.

- **Memory Acquisition Stage:** Agents perform tasks with conventional object rearrangement task instructions containing personalized knowledge while accumulating the interaction history (*i.e.*, episodic memory). The goal of this stage is to provide a reference performance baseline.
- **Memory Utilization Stage:** Agents execute the same tasks as in the memory acquisition stage but with modified instructions that require agents to recall and apply the previously acquired personalized knowledge to succeed. The goal of this stage is to evaluate how well agents can utilize memory by comparing the performance drop relative to the acquisition stage.

Given the base object rearrangement task episode defined as the tuple $\epsilon = (S, I, g)$, the key concept of our evaluation process is to share the scene $S$ and goal representation $g$, while varying the instruction $I$ across the two stages to isolate instruction interpretation capability as the primary factor influencing performance differences. Formally, in the *memory acquisition stage*, each episode is defined as $\epsilon_{acq} = (S, I_{acq}, g)$, where the instruction $I_{acq}$ contains sufficient information to infer the goal $g$, denoted as:

$$\phi(I_{acq}) \longrightarrow g \tag{2}$$

During this stage, we also store the episodic memory $h_{acq}$:

$$h_{acq} = (I_{acq}, \tau_k) \in H_{acq}, \quad \tau_k = (w_1, a_1, w_2, a_2, \ldots, w_{k-1}, a_{k-1}, w_k) \tag{3}$$

In the subsequent *memory utilization stage*, each episode is defined as $\epsilon_{util} = (S, I_{util}, g)$, where the instruction $I_{util}$ is intentionally underspecified and requires the agent to recall the corresponding episodic memory $h_{acq}$ to correctly interpret the goal $g$, formally:

$$\phi(I_{util}, h_{acq}) \longrightarrow g \tag{4}$$

Through this design concept, we are able to quantify the agent's ability to utilize memory by comparing performance between the two stages.

**Memory utilization stage task design.** Within our two-stage evaluation process design, we can assess embodied agents' ability to utilize personalized knowledge from a single episodic memory. However, this approach alone fails to capture real-world complexity and lacks analytical diversity for comprehensive assessment. Therefore, to evaluate different levels of memory complexity, we divide our assessment into (1) *Single-memory task*, which require utilizing information from one episodic memory, and (2) *Joint-memory task*, which necessitate synthesizing information from two distinct episodic memories to successfully complete the episode. Building on our evaluation process concept, we form the joint-memory task by concatenating two episodes, which can be formulated as:

$$\epsilon_{util}^{joint} = (S, I_{util}^{joint}, [g^i; g^j]) \tag{5}$$

where $i$, $j$ denotes the corresponding reference episodes from the memory acquisition stage.

### 4.2 Personalized Knowledge Categorization

Building upon our memory evaluation process, we aim to analyze embodied agents' ability to understand personalized knowledge in object rearrangement tasks by focusing on its role in goal interpretation, $\phi(I) \rightarrow g = \{(o_i, l_i)\}_{i=1}^{k}$, where each $o_i$ and $l_i$ denote the target object and location, respectively. To facilitate this analysis, we categorize personalized knowledge into two types, each comprising subcategories [2] that reflect how users naturally express preferences and routines in real-world interactions. Each type is designed to isolate a distinct reasoning challenge that the agent must resolve by utilizing episodic memory during the memory utilization stage.

- **Object semantics:** Individual objects that the user assigns personal meaning, encompassing subcategories such as ownership (*e.g.*, *my cup*), preference (*e.g.*, *my favorite running gear*), past history (*e.g.*, *a graduation gift from my grandma*), or grouped references (*e.g.*, *my childhood toy collections*). This category tests whether the agent can identify the target object $o_i$ by recalling its personal meaning from prior interactions.
- **User patterns:** Sequences of actions that the user consistently performs, including personal routines (*e.g.*, *my remote work setup*) and arrangement preferences (*e.g.*, *my cozy dinner atmosphere*) across recurring contexts. This category evaluates the agent's ability to reconstruct the complete goal $g$ by leveraging previously observed behavioral patterns across multiple objects and locations.

### 4.3 Dataset Construction Process

We constructed the dataset for MEMENTO through a four-step process using the Habitat 3.0 simulator [38] as the environment, with a simulated Spot robot as the agent [5, 58]. Our custom dataset spans 12 scenes, comprising a total of 438 episodes distributed across stages. Notably, the memory acquisition stage and the single-memory task in the utilization stage have the same number of episodes, whereas the joint-memory task contains fewer episodes. The detailed dataset statistics and the explanation of the construct process is described in Appendix C.3, and Appendix C.4.

**Step 1: Object rearrangement task collection.** We use the test set of the PartNR [7] as our foundation object rearrangement task data. Unlike simple pick-and-place tasks, PartNR episodes require completing multiple object–location pairs within a single instruction, which aligns with our memory evaluation process design.

**Step 2: Scene augmentation with distractor objects.** In the original scenes, there was an issue where no objects of the same type as the target object were present. As a result, the agent could identify the target object without needing to understand personalized knowledge. To address this, we augmented the scenes by placing distractor objects of the same type near the target object. For example, if the target object is a "blue cup" on the table, we place a "red cup" next to it as a distractor.

**Step 3: Task instruction generation.** We first generate personalized knowledge contextually tailored to the original task instruction using GPT-4o. With the generated personalized knowledge, we applied to both stage instruction curation. As illustrated in Figure 2, the *memory acquisition stage* instruction $I_{acq}$ is constructed by concatenating the base instruction, object visual captions (only for

[2] A detailed explanation of the sub-categories of personalized knowledge can be found in Appendix C.1.

episodes of the object semantics type), and the generated personalized knowledge. This ensures that the goal can be inferred directly from the instruction. For the *memory utilization stage*, we prompt GPT-4o to generate a personalized instruction that implicitly reflects the personalized knowledge, based on the same base instruction. For joint-memory tasks, we concatenate two such personalized instructions sequentially.

**Step 4: Quality control.** To ensure data quality, we first heuristically filtered episodes containing similar memories referencing identical objects within scenes, preventing interference between similar episodic memories. Subsequently, we manually reviewed episodes from the memory acquisition stage where GPT-4o failed to successfully complete the task, where we filtered out episodes that contained unnatural instructions or cases where the generated instructions did not match the intended goal representation, ensuring the quality of our evaluation data.

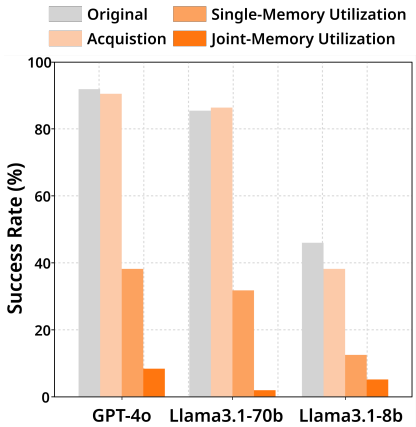

Figure 3: The performance results without using episodic memory. Original indicates the conventional object rearrangement task episodes.

### 4.4 Validation of MEMENTO

To validate that MEMENTO effectively assesses embodied agents' memory utilization capability, we compare performance across stages in a setup without memory retrieval. As shown in Figure 3, compared to the results of the original object rearrangement task and the memory acquisition stage, embodied agents struggle to complete tasks in the memory utilization stage. Since the underlying episode remains the same across all tasks, this result confirms that the instruction $I_{util}$ in the memory utilization stage is difficult to interpret without access to previous interaction histories. Notably, we observed a particular behavior of the embodied agents: when there are two objects of the same type, agents that do not understand which one is the actual target tend to randomly select one and proceed with the task. This explains the reason why performance in the single-memory task is higher than that of the joint-memory task.

## 5 Evaluating Personalized Knowledge Utilization in Episodic Memory

### 5.1 Experimental Setup

**Evaluation metrics.** Following Chang et al. [7], we use two main metrics: **Percent Complete (PC)** for the proportion of goal completion, and **Success Rate (SR)** for full task completion. We also report **Sim Steps**, which show the number of simulation steps required for agents to complete the task, and **Planning Cycles**, which indicate the number of LLM inference calls made during task execution. To evaluate memory utilization, we also report performance drops between acquisition and utilization stages as **ΔPC** and **ΔSR**. Note that for joint-memory tasks, these differences are computed relative to the average performance of the corresponding acquisition-stage episodes.

**Implementations.** Following prior work [45, 38, 7], we implement a LLM-powered embodied agent architecture, where the LLM functions as a high-level policy planner that selects appropriate skills from a predefined skill library. We use ReAct [56] prompt format for LLMs to take actions. Additionally, we implement a top-5 memory retrieval setup [3] for the memory utilization stage, ensuring the corresponding memory is included in the retrieved results by randomly replacing one memory if the correct one is not initially retrieved. Full implementation details are in AppendixB.2.

**Models.** We evaluate embodied agents powered by a range of LLMs to compare memory utilization capabilities across model families and sizes, including proprietary models (GPT-4o [21], Claude-3.5-Sonnet [3]) and open-source models (Llama-3.1-70b/8b [15], Qwen-2.5-72b/7b [54]).

[3]We use an embedding-based retrieval method with the *all-mpnet-base-v2* Sentence Transformer [40].

Table 1: Model performance across memory acquisition and utilization stage in MEMENTO.

| Model | Stage | Task Type (Memory) | Planning Cycles ↓ | Sim Steps ↓ | Percent Complete ↑ | Δ PC | Success Rate ↑ | Δ SR |
|---|---|---|---|---|---|---|---|---|
| GPT-4o | Acquisition | - | 16.5 | 2156.1 | 96.3 | - | 95.0 | - |
| | Utilization | Single | 16.1 | 2450.8 | 88.0 | **-8.3** | 85.1 | **-9.9** |
| | | Joint | 28.9 | 3480.7 | 86.7 | **-10.5** | 63.9 | **-30.5** |
| Claude-3.5-Sonnet | Acquisition | - | 16.0 | 2104.1 | 96.2 | - | 94.0 | - |
| | Utilization | Single | 15.3 | 2258.8 | 69.3 | **-26.9** | 63.7 | **-30.3** |
| | | Joint | 27.8 | 3198.8 | 64.6 | **-30.1** | 33.3 | **-57.0** |
| Qwen-2.5-72b | Acquisition | - | 17.5 | 2281.9 | 93.5 | - | 91.0 | - |
| | Utilization | Single | 17.5 | 2691.2 | 72.6 | **-20.9** | 67.2 | **-23.8** |
| | | Joint | 31.3 | 4027.1 | 68.9 | **-27.9** | 36.1 | **-58.3** |
| Llama-3.1-70b | Acquisition | - | 17.7 | 2162.1 | 92.9 | - | 90.0 | - |
| | Utilization | Single | 19.0 | 2566.6 | 72.2 | **-20.7** | 66.7 | **-23.3** |
| | | Joint | 31.4 | 3425.2 | 51.3 | **-44.9** | 8.3 | **-83.4** |
| Llama-3.1-8b | Acquisition | - | 19.3 | 2377.0 | 78.1 | - | 68.5 | - |
| | Utilization | Single | 19.0 | 3131.7 | 48.1 | **-30.0** | 35.0 | **-33.5** |
| | | Joint | 27.4 | 3478.2 | 35.3 | **-45.5** | 8.3 | **-59.8** |
| Qwen-2.5-7b | Acquisition | - | 21.7 | 2476.8 | 64.1 | - | 53.2 | - |
| | Utilization | Single | 21.8 | 3271.0 | 39.1 | **-25.0** | 27.4 | **-25.8** |
| | | Joint | 26.9 | 4149.0 | 33.7 | **-34.2** | 5.6 | **-52.7** |

### 5.2 Main Results

**LLM-powered embodied agents struggle with understanding personalized knowledge.** As shown in Table 1, while GPT-4o maintains a relatively high success rate in the single-memory task, all models show a success rate drop over 20% compared to the memory acquisition stage. In particular, for joint-memory tasks even GPT-4o exhibits a 30.5% drop in success rate, highlighting the increased difficulty of these settings. This substantial performance decline demonstrates that even frontier models struggle to accurately reference personalized knowledge from episodic memory, and often fail to consistently apply it across multiple steps in long-horizon task planning.

**LLM-powered embodied agents exhibit increased exploration behavior on joint-memory tasks.** Joint-memory task results reveal that LLMs (even GPT-4o) find it difficult to recall personalized knowledge from different memory sources. The number of planning cycles and simulation steps significantly increases compared to other tasks, this suggests that the embodied agent fails to correctly interpret the instruction, leading to excessive exploration during task execution. Also the performance gap between percent complete and success rate is larger than in the single-memory task, which indicates that the agent frequently misses part of the necessary information for successful task completion.

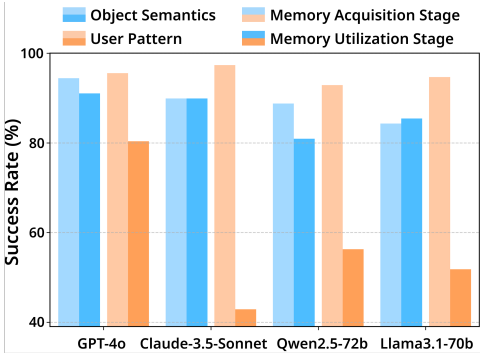

Figure 4: The results of personalized knowledge type based analysis (single-memory).

### 5.3 In-depth Analysis

To better understand the challenges of providing personalized assistance, we examine two key aspects: the root causes of their limitations and how the top-$k$ memory setting affects memory utilization.

#### 5.3.1 Personalized Knowledge Type-based Analysis

We analyze the performance gap between the memory acquisition stage and the single-memory task from the memory utilization stage by comparing success rates across different types of personalized knowledge. Analysis results for the joint-memory task are provided in Appendix D.2.

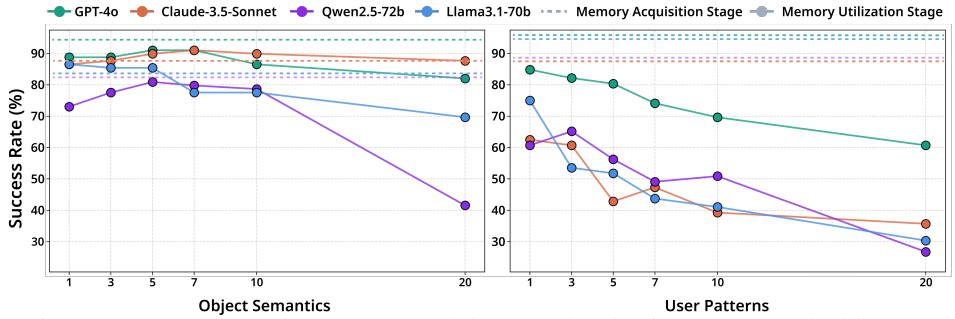


Figure 5: Success rate comparison across models as top-k value increases. Dashed lines represent memory acquisition stage baselines for each model.


Figure 6: Examples of success and error cases in memory utilization stage. Top: success and failure cases of object semantics; Bottom: success and failure cases of user patterns.

**LLMs can recall objects but struggle to comprehend action sequences.** As shown in Figure 4, all models exhibit substantially larger performance drops for tasks requiring understanding user patterns compared to object semantics. Notably, the performance drop in object semantics is minimal, indicating that LLMs are relatively effective at directly recalling relevant memories to identify target objects. In contrast, tasks involving user patterns pose a significantly greater challenge, as they require integrating and reasoning over sequences of events.

### 5.3.2 Memory Top-K Analysis

We analyze how varying the number of retrieved memories ($k$) impacts the agent's ability to utilize personalized knowledge from episodic memory.

**Irrelevant memories act as a distraction to agents.** In Figure 5, we observe that as $k$ increases, all models exhibit consistent performance degradation across both task types, highlighting the difficulty of identifying the exact information from a growing set of retrieved memories. Consistent with previous findings, the degradation is especially pronounced in tasks requiring user patterns understanding, suggesting that such tasks are particularly vulnerable to noise when recalling and executing multi-step procedures grounded in implicit, personalized knowledge.

## 5.4 Empirical Analysis of Success and Error Cases

To better understand how LLM-powered embodied agents utilize personalized knowledge derived from episodic memory during task planning, we conduct qualitative case studies of both successful and failed episodes. Figure 6 presents the taxonomy of case types along with illustrative examples. Detailed explanations for each case type are provided in Appendix D.4. Below, we highlight a set of representative cases that offer key insights into the embodied agents' behavior.

- **Agent misses personalization cues (B.1):** The agent fails to recognize user-specific references and treats them as generic or proper nouns, stemming from incorrect interpretations of user intent.

- **Use of commonsense knowledge over personalized knowledge (C.1 & D.1):** Even when relevant episodic memory is available, the agent often relies on general commonsense knowledge to infer user routines, rather than using the personalized knowledge. This tendency appears in both successful and failed episodes.

# 6 Discussion

## 6.1 The Impact of Trajectory in Episodic Memory

MEMENTO's design incorporates complete action-observation trajectories in episodic memory, raising the question of whether agents should reference these detailed trajectories rather than relying solely on user instructions. To investigate this, we conduct an additional comparative experiment to evaluate if providing full action trajectories offers significant advantages over simply providing instructions that state personalized knowledge. For the experiment setup, we evaluate model performance across three different cases of memory provided to the agent as shown in Table 2. The results reveal that while larger models (GPT-4o, Qwen-2.5-72b) perform well with only high-level plans (b), smaller models (Llama-3.1-8b, Qwen-2.5-7b) require full procedural details from completed trajectories (a) to succeed. Most notably, all models show substantial performance drops when given only user instructions (c), suggesting that action trajectories contain essential procedural cues necessary for understanding personalized knowledge, regardless of model capacity. Experiment setup details are provided at Appendix D.5.

Table 2: Analysis of how memory type affects agent performance: (a) complete action action trajectories containing user instructions, (b) summary of (a), and (c) user instructions only.

| Model | Memory | PC (%) | SR (%) |
|---|---|---|---|
| GPT-4o | (a) | 90.0 | 83.3 |
| | (b) | 88.0 | 83.3 |
| | (c) | 62.4 | 50.0 |
| Qwen-2.5-72b | (a) | 77.2 | 66.7 |
| | (b) | 77.4 | 70.0 |
| | (c) | 51.3 | 40.0 |
| Llama-3.1-8b | (a) | 72.8 | 63.3 |
| | (b) | 49.4 | 43.3 |
| | (c) | 40.0 | 30.0 |
| Qwen-2.5-7b | (a) | 50.1 | 43.3 |
| | (b) | 43.9 | 36.7 |
| | (c) | 35.6 | 23.3 |

## 6.2 Agent Behavior under Ambiguous Instructions from Users

While MEMENTO focuses on evaluating personalized knowledge grounding with explicit references to prior interactions, real-world human-agent communication often involves ambiguous or indirect references. To explore this challenge, we conducted a proof-of-concept experiment to assess whether current models can interpret ambiguous instructions that indirectly refer to previously encountered personalized knowledge. [4] We created a tailored set of tasks referencing personalized knowledge using contextual cues, synonyms or causal references (*e.g.*, *Can you set my afternoon tea time routine?* $\rightarrow$ *I'm about to enjoy my afternoon tea. Could you set things up as I like them?*). The results, shown in Table 3, reveal the degradation in performance, indicating that handling ambiguous queries remains a key challenge for future personalized embodied agents. We view this as a promising direction for future work, where we plan to systematically extend MEMENTO to evaluate LLM-powered embodied agents' capabilities under ambiguous and implicit reference scenarios, aiming to better reflect the complexity of real-world human-agent interaction.

Table 3: Performance under ambiguous queries for personalized knowledge. PC (Percent Complete) and SR (Success Rate) (%) indicate how well agents resolve indirect references to personalized knowledge from memory.

| Model | PC (%) | SR (%) |
|---|---|---|
| GPT-4o (Baseline) | 92.0 | 90.0 |
| GPT-4o | 80.4 | 73.3 |
| Qwen-2.5-72b (Baseline) | 75.1 | 66.7 |
| Qwen-2.5-72b | 59.6 | 53.3 |

[4] Further details of the dataset and experiment setup are provided in Appendix D.5.

## 7 Conclusion

In this work, we present MEMENTO, a novel personalized embodied agent evaluation framework designed to assess the LLM-powered embodied agents' ability to utilize episodic memory for providing personalized assistance. Our experiments across a range of LLM-powered embodied agents reveal key limitations in their ability to effectively leverage personalized knowledge from memory, particularly when integrating multiple memories and interpreting user patterns. These findings highlight the gap between current capabilities and the demands of real-world personalized assistance. We hope that MEMENTO serves as a stepping stone for future research in developing more effective personalized embodied agents.

## References


[1] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti. Taskography: Evaluating robot task planning over large 3d scene graphs. In *Conference on Robot Learning*, pages 46–58. PMLR, 2022.

[2] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[3] Anthropic. Introducing claude 3.5 sonnet, June 2024. URL https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2025-05-07.

[4] L. Barsellotti, R. Bigazzi, M. Cornia, L. Baraldi, and R. Cucchiara. Personalized instance-based navigation toward user-specific objects in realistic environments. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=uKqn1Flsbp.

[5] Boston Dynamics. Spot robot. https://bostondynamics.com/products/spot/, 2025. Accessed: 2025-04-28.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[7] M. Chang, G. Chhablani, A. Clegg, M. D. Cote, R. Desai, M. Hlavac, V. Karashchuk, J. Krantz, R. Mottaghi, P. Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv preprint arXiv:2411.00081*, 2024.

[8] Y. Chen, J. Arkin, C. Dawson, Y. Zhang, N. Roy, and C. Fan. Autotamp: Autoregressive task and motion planning with llms as translators and checkers. In *2024 IEEE International conference on robotics and automation (ICRA)*, pages 6695–6702. IEEE, 2024.

[9] J.-W. Choi, Y. Yoon, H. Ong, J. Kim, and M. Jang. Lota-bench: Benchmarking language-oriented task planners for embodied agents. In *International Conference on Learning Representations (ICLR)*, 2024.

[10] C. Clabaugh and M. Matarić. Robots for the people, by the people: Personalizing human-machine interaction. *Science Robotics*, 3(21):eaat7451, 2018. doi: 10.1126/scirobotics.aat7451. URL https://www.science.org/doi/abs/10.1126/scirobotics.aat7451.

[11] Y. Dai, R. Peng, S. Li, and J. Chai. Think, act, and ask: Open-world interactive personalized robot navigation, 2024. URL https://arxiv.org/abs/2310.07968.

[12] P. Das, S. Chaudhury, E. Nelson, I. Melnyk, S. Swaminathan, S. Dai, A. Lozano, G. Kollias, V. Chenthamarakshan, Jiří, Navrátil, S. Dan, and P.-Y. Chen. Larimar: Large language models with episodic memory control, 2024. URL https://arxiv.org/abs/2403.11901.

[13] K. Dautenhahn. Robots we like to live with?! - a developmental perspective on a personalized, life-long robot companion. In *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*, pages 17–22, 2004. doi: 10.1109/ROMAN.2004.1374720.

[14] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[15] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[17] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022.

[18] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[19] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman, and brian ichter. Grounded decoding: Guiding text generation with grounded models for embodied agents. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JCCi58IUsh.

[20] A. Huet, Z. B. Houidi, and D. Rossi. Episodic memories generation and evaluation benchmark for large language models. *arXiv preprint arXiv:2501.13121*, 2025.

[21] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[22] I. Kapelyukh and E. Johns. My house, my rules: Learning tidying preferences with graph neural networks. *CoRR*, abs/2111.03112, 2021. URL https://arxiv.org/abs/2111.03112.

[23] B. Kim, J. Kim, Y. Kim, C. Min, and J. Choi. Context-aware planning and environment-aware memory for instruction following embodied agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10936–10946, October 2023.

[24] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[25] M. K. Lee, J. Forlizzi, S. Kiesler, P. Rybski, J. Antanitis, and S. Savetsila. Personalization in hri: A longitudinal field experiment. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 319–326, 2012.

[26] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, M. Hwang, M. Sharma, A. Aydin, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, S. Savarese, H. Gweon, K. Liu, J. Wu, and L. Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 80–93. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/li23a.html.

[27] H. Li, C. Yang, A. Zhang, Y. Deng, X. Wang, and T.-S. Chua. Hello again! LLM-powered personalized agent for long-term dialogue. In L. Chiruzzo, A. Ritter, and L. Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5259–5276, Albuquerque, New Mexico, Apr. 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.272/.

[28] M. Li, S. Zhao, Q. Wang, K. Wang, Y. Zhou, S. Srivastava, C. Gokmen, T. Lee, E. L. Li, R. Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2025.

[29] S. Li, X. Puig, C. Paxton, Y. Du, C. Wang, L. Fan, T. Chen, D.-A. Huang, E. Akyürek, A. Anandkumar, J. Andreas, I. Mordatch, A. Torralba, and Y. Zhu. Pre-trained language models for interactive decision-making. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, 2022. ISBN 9781713871088.

[30] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *arXiv preprint arXiv:2209.07753*, 2022.

[31] Z. Liu, A. Bahety, and S. Song. Reflect: Summarizing robot experiences for failure explanation and correction. *arXiv preprint arXiv:2306.15724*, 2023.

[32] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[33] Meta. Introducing llama 3.1: Our most capable models to date, July 2024. URL `https://ai.meta.com/blog/meta-llama-3-1/`. Accessed: 2025-05-16.

[34] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo. EmbodiedGPT: Vision-language pre-training via embodied chain of thought. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=IL5zJqfxAa`.

[35] C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica, and J. E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL `https://arxiv.org/abs/2310.08560`.

[36] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.

[37] X. Puig, T. Shu, S. Li, Z. Wang, Y.-H. Liao, J. B. Tenenbaum, S. Fidler, and A. Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration, 2021. URL `https://arxiv.org/abs/2010.09890`.

[38] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[39] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.

[40] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[41] M. Rueben and W. D. Smart. Privacy in human-robot interaction: Survey and future work. *We robot*, 2016:5th, 2016.

[42] G. Sarch, Y. Wu, M. J. Tarr, and K. Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. *arXiv preprint arXiv:2310.15127*, 2023.

[43] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530, 2023. doi: 10.1109/ICRA48891.2023.10161317.

[44] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009, 2023.

[45] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

[46] B. Wang, X. Liang, J. Yang, H. Huang, S. Wu, P. Wu, L. Lu, Z. Ma, and Z. Li. Enhancing large language model with self-controlled memory framework, 2024.

[47] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[48] Z. Wang, B. Yu, J. Zhao, W. Sun, S. Hou, S. Liang, X. Hu, Y. Han, and Y. Gan. Karma: Augmenting embodied ai agents with long-and-short term memory systems. *arXiv preprint arXiv:2409.14908*, 2024.

[49] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.

[50] Y. Wu, J. Zhang, N. Hu, L. Tang, G. Qi, J. Shao, J. Ren, and W. Song. Mldt: Multi-level decomposition for complex long-horizon robotic task planning with open-source large language model. In *International Conference on Database Systems for Advanced Applications*, pages 251–267. Springer, 2024.

[51] Q. Xie, S. Y. Min, P. Ji, Y. Yang, T. Zhang, K. Xu, A. Bajaj, R. Salakhutdinov, M. Johnson-Roberson, and Y. Bisk. Embodied-rag: General non-parametric embodied memory for retrieval and generation. *arXiv preprint arXiv:2409.18313*, 2024.

[52] M. Xu, X. Yang, W. Liang, C. Zhang, and Y. Zhu. Learning to plan with personalized preferences. *arXiv preprint arXiv:2502.00858*, 2024.

[53] W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y. Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

[54] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[55] R. Yang, H. Chen, J. Zhang, M. Zhao, C. Qian, K. Wang, Q. Wang, T. V. Koripella, M. Movahedi, M. Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.

[56] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[57] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*, 2023.

[58] N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T.-Y. Yang, S. Arnaud, S. Ha, D. Batra, and A. Rai. Asc: Adaptive skill coordination for robotic mobile manipulation. *IEEE Robotics and Automation Letters*, 9(1):779–786, 2023.

[59] H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J. B. Tenenbaum, T. Shu, and C. Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023.

[60] W. Zhong, L. Guo, Q. Gao, and Y. Wang. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

# A Limitations

## A.1 Limitations

**Controlled simulator environment.** Our experiments are conducted entirely in a controlled simulator environment [38], which does not fully reflect the complexities of real-world robotics such as perception noise, actuation uncertainty, or unstructured environments.

**Visual perception is not addressed.** Our framework deliberately isolates memory-centric planning by excluding visual perception components such as VLMs [55] or VLA [6, 24] models. This enables focused evaluation of memory utilization, but limits the system's generalizability to fully grounded perception scenarios.

**Oracle skills.** We intentionally employ oracle skills for low-level perception and motor control to isolate and focus on the high-level planning and memory reasoning capabilities of the LLM-based agents. As a result, our framework does not evaluate the embodied agent's full task execution process.

## A.2 Societal Impacts

Our work explores how embodied agents can remember and adapt to user-specific preferences through episodic memory, enabling more personalized and natural interactions. This capability has the potential to enhance convenience, efficiency, and user satisfaction in everyday environments—benefiting a wide range of users regardless of age or ability. However, personalization also raises concerns about privacy, bias reinforcement, and over-dependence on AI agents [41, 25]. Since episodic memory involves storing interaction history, future systems must consider secure and transparent memory handling. Although our study is conducted in a controlled simulator, these considerations will become crucial as such systems move toward real-world deployment. We hope that MEMENTO serves as a foundation for future research in building safe, privacy-aware, and user-aligned personalized embodied agents.

# B Details of Experiment Setup

## B.1 Evaluation Method

We adopt the official evaluation protocol from the PartNR benchmark [7], which provides a python-based framework for assessing multi-step rearrangement tasks. We use this framework without modification. The evaluator analyzes simulator states using three components: (1) **propositions** that define object relationships to satisfy (e.g., `is_on_top([spoon_1], [table_1])`), (2) **dependencies** that define temporal conditions for multi-step instructions (`after_satisfied`, `after_unsatisfied`), and (3) **constraints** that enforce execution requirements (e.g., ordering, object consistency across steps). We rely on this system to evaluate tasks involving ambiguous references and sequential dependencies. The evaluation produces a percent-complete score and binary success indicator.

## B.2 LLM-Powered Embodied Agent Architecture

Following Szot et al. [45], Puig et al. [38], Chang et al. [7], we adopt a two-layer hierarchical control architecture for our LLM-powered embodied agent. We utilize the LLM as a high-level policy planner that selects appropriate skills from the predefined skill library. The selected skill then provides control signals to the simulator. For memory systems, we implement a textual scene-graph as our semantic memory alongside an episodic memory.

**Skill library.** The skill library consists of oracle low-level skills that the LLM policy can select as actions. These action skills are divided into motor skills (*e.g.*, `navigate`, `pick`, `place`) and perception skills (*e.g.*, `describe_object`, `find_object`, `find_receptacle`). Note that we exclusively used oracle skills in our skill library to focus on episodic memory-centric analysis. Further descriptions are provided in Table 4 and Table 5

Table 4: List of available agent motor skills.

| Skill | Description |
|---|---|
| Navigate | Used for navigating to an entity. You must provide the name of the entity you want to navigate to. Example: Navigate[counter_22]. |
| Pick | Used for picking up an object. You must provide the name of the object. Example: Pick[cup_1]. |
| Place | Used for placing an object on a target location. Example: Place[book_0, on, table_2, None, None]. |
| Open | Used for opening an articulated entity. Example: Open[chest_of_drawers_1]. |
| Close | Used for closing an articulated entity. Example: Close[chest_of_drawers_1]. |
| Explore | Doing exploration towards a target object or receptacle, you need to provide the name of the place you want to explore. |
| Wait | Used to make agent stay idle for some time. Example (Wait[]) |

Table 5: List of available agent perception skills.

| Skill | Description |
|---|---|
| FindObjectTool | Used to find the exact name/names of the object/objects of interest. If you want to find the exact names of objects on specific receptacles or furnitures, please include that in the query. Example (FindObjectTool[toys on the floor] or FindObjectTool[apples]). |
| FindReceptacleTool | Used to know the exact name of a receptacle. A receptacle is a furniture or entity (like a chair, table, bed etc.) where you can place an object. Example (FindReceptacleTool[a kitchen counter]). |
| FindRoomTool | Used to know the exact name of a room in the house. A room is a region in the house where furniture is placed. Example (FindRoomTool[a room which might have something to eat]). |
| DescribeObjectTool | Used to retrieve a brief descriptive or semantic meaning of a given object or furniture name. Example (DescribeObjectTool[sponge_1]). |

**Semantic memory.** For semantic memory, we implement a scene-graph style hierarchical representation, which has demonstrated effectiveness for planning problems [1, 39, 16]. Following Chang et al. [7], we utilize a multi-edge directed graph with three distinct levels to represent environmental entities. The top level contains a single root node representing the house environment, the second level comprises room nodes, and the third level encompasses furniture, objects, and agents. Each node stores the corresponding entity's 3D location and relevant state information. This graph structure is initialized and continuously updated with ground-truth information from the simulator at each state $s_t$. In our system, this structured semantic memory provides the LLM planner with an interpretable representation of the environment, which can be flexibly queried and reasoned over through natural language descriptions.

**Episodic memory.** Our episodic memory is configured to store the ReAct-style formatting that guides the LLMs' reasoning process [56]. This memory structure captures both the user's instruction and the complete sequence of `<Thought, Action, Observation>` triplets generated during task execution. The episodic memory is accessed at the beginning of each task by retrieval and updated upon task completion, enabling the agent to recall previous interactions.

**Memory retrieval.** For retrieval, we encode instructions and memory entries using the *all-mpnet-base-v2* sentence transformer [40] and use the current task instruction as the query. To avoid ambiguous or irrelevant memories, we retrieve candidate memories only from the history within the same scene as the current task.

**LLM Setup.** We configured the language model with a temperature of 0 to ensure deterministic outputs. For sampling parameters, we set `top_p` to 1 and `top_k` to 50.

### B.3 Computing resources

Our experiments primarily utilized commercial API services rather than local computing resources. We used OpenAI's Chat API for accessing GPT-4o, Claude models through Anthropic's API, and OpenRouter's API service for accessing Llama-3.1 [33], Qwen-2.5 [54]. For running simulation

environment we used 8 NVIDIA GeForce RTX 3090 GPUs. For our implementation and evaluation, we use Huggingface library2, vLLM library. Both libraries are licensed under Apache License, Version 2.0. And we used langchain library, under MIT License. We have confirmed that all of the artifacts used in this paper are available for non-commercial scientific use.

# C Details of MEMENTO

## C.1 Personalized Knowledge

We categorize knowledge about personal items as *object semantics* and knowledge about consistent behaviors as *user patterns* to structure our evaluation approach. *Object semantics* can be further classified into four sub-categories: naive ownership (*e.g.*, "my cup"), object preference (*e.g.*, "a chessboard I play chess with my brother"), history (*e.g.*, "graduation gift from my grandma"), and group (*e.g.*, "my favorite toys", where toys indicate toy airplane and toy truck). *User patterns* encompass consistent action sequences in specific contexts, with two sub-categories: personal routine (*e.g.*, "my remote work setup") and arrangement preference (*e.g.*, "my movie night setup"). Based on these personalized knowledge categories, we designed tasks that specifically require agents to recall and apply this information to evaluate their memory utilization capabilities. Further details are provided in Table 6 and Table 7

Table 6: List of the subcategories for object semantics.

| **Type** | **Description** | **Example** |
|---|---|---|
| Ownership | Possessive reference to the user's belonging | My cup, My laptop |
| Preference | Object aligned with the user's individual taste or selection | Bread from my favorite bakery, Jug for serving drink |
| History | Object linked to personal memory or past experience | photo of the my beloved pet, travel souvenir vase |
| Groups | Conceptual or functional grouping of multiple related objects | my home office setup, my travel essentials |

Table 7: List of the subcategories for user patterns.

| **Type** | **Description** | **Example** |
|---|---|---|
| Routine | A sequence or setup the user follows as a habit or regular activity. | meal time setting, setup for cooking routine |
| Preference | A specific way the user prefers to prepare or arrange their environment when a particular situation occurs. | my coffee break, cozy decoration spot |

## C.2 PartNR Dataset

PartNR is designed to evaluate planning and reasoning capabilities in embodied tasks and is recognized for its comprehensive collection of natural language instructions in household environments. The benchmark includes four primary task types: (1) constraint-free basic rearrangement tasks, (2) spatial tasks requiring reasoning about object positions, (3) temporal tasks with sequential dependencies, and (4) heterogeneous tasks involving actions that can only be performed by human agents. We selected PartNR episodes specifically for their complexity beyond simple pick-and-place operations. The rich linguistic structure and diverse task requirements make PartNR particularly suitable for evaluating user patterns personalization, enabling us to create scenarios that effectively test an agent's ability to adapt to user-specific communication patterns.

## C.3 Dataset Statistics

The dataset comprises 438 episodes, divided into two main stages. The memory acquisition stage contains 201 episodes (89 object semantics tasks and 112 user patterns tasks). The memory utilization stage also contains 201 single-memory episodes (89 object semantics and 112 user patterns tasks), along with 36 multi-memory episodes, which include 12 object semantics pairs, 12 user patterns pairs, and 12 mixed pairs. All episodes were constructed using the Habitat 3.0 simulator.

### C.4 Data Generation Process

To create stage-specific tasks, we leveraged GPT-4o and systematic process to incorporate personalized knowledge into existing task structures.

**Captioning process.** Since we employ LLMs as high-level planners for embodied agent, we generated natural language descriptions of the objects at the scene using GPT-4o, to enable agents to reason over object descriptions without relying on direct visual perception. We collected object models from the OVMM dataset [57], and used GPT-4o to generate natural language descriptions from rendered object images. Especially, the Google Scanned dataset [14], included within OVMM, provides object identities in file names. We leveraged this additional information alongside the images to produce more realistic descriptions. Through this process, we generated 1,920 object descriptions with 66 categories of objects. The prompts used to generate the object descriptions are provided in Appendix E.

**Preprocessing scenes.** To collect the suitable episodes for our purposes, we preprocessed and filtered episodes. First, we only gathered non-heterogeneous episodes, which should be executed by human agents. Second, we filtered out tasks where the target object was not uniquely specified (*e.g.*, "Bring one apple to the kitchen table"), as our task setting requires identification and reference of a specific object based on personalized knowledge that distinguishes it from other similar objects. Third, in cases where captions for target handles were unavailable, we substituted alternative objects. If no objects from the same category were available, we excluded the entire episode from our dataset.

**Distractor sampling.** To sample distractor objects for episodes focusing on object semantics, we utilized PartNR's dataset generation methods. This approach allowed us to systematically select objects located adjacent to target objects on the same receptacles or floor surfaces, requiring agents to differentiate between objects of the same category.

**Details of task instruction generation.** For instruction generation, we prompted GPT-4o to generate personalized knowledge tasks based on the knowledge categories defined in Section C.1. For tasks involving object semantics, we provided the captions of the target object pairs along with instructions to guide GPT-4o in generating natural object semantics descriptions. (e.g., instruction: "Bring the cup on the kitchen table", object description: "a white mug with fancy handle", personalized knowledge: "The mug is my coffee mug.") We first instructed the model to create the most plausible subcategory of object semantics, which was then used to generate personalized knowledge specific to the objects. In these object semantics cases, the instruction comprised three components: command instruction (e.g., "Bring the cup on the kitchen table"), additional information (e.g., "The cup is a white mug with fancy handle."), and personalized knowledge (e.g., "That mug is my coffee mug"). For user patterns tasks, we provided instructions to GPT-4o and allowed the model to infer the most plausible user patterns corresponding to a sequence of actions. For the memory acquisition stage instruction, we also concatenated the command instruction (e.g., "Bring the cup and dish from the kitchen table to living room") with personalized knowledge (e.g., "That's my dinner setup"). Detailed information about the prompts we used is provided in Appendix E.

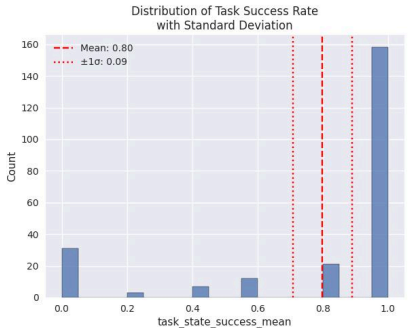

Figure 7: Episodes with zero success rate (31 in total) were excluded from the analysis.

**Quality control.** After generating episodes for MEMENTO, we implemented a two-stage filtering process for quality control. First, we applied heuristic filtering to eliminate episodes with potentially confusing distractor objects in the same scene (e.g., episodes using identical object handles or requiring similar personalized knowledge, which could create ambiguity). Second, we tested each episode with GPT-4o and excluded any episode where GPT-4o failed five consecutive attempts, indicating potential issues with task feasibility or clarity. (Figure 7) This quality control process resulted in filtering out 31 episodes (13.4% of the total dataset) that exhibited consistently poor performance. Specifically, we identified: 31 episodes with zero success rate, indicating complete

task failure. The remaining 201 episodes showed strong performance metrics, with 95% confidence intervals of [0.755, 0.844] for success rate and [0.823, 0.893] for completion rate. The high correlation (r = 0.908) between success rate and completion rate indicates consistent performance across different evaluation metrics. These high-quality episodes served as the foundation for generating golden trajectories, which were subsequently used in our memory quality analysis and discussion experiments. By establishing GPT-4o's successful task executions as golden trajectories, we ensure that our evaluation framework is based on demonstrably achievable performance standards, providing a reliable benchmark for assessing memory quality and discussion effectiveness.

# D Additional Experiments & Analysis

## D.1 Knowledge Type-based Analysis.

Additional experiment results show the performance of small models—Llama3.1-8b and Qwen2.5-7b—on knowledge type-based analysis (Figure 8). Similar to other frontier models, they struggled with utilizing episodic memory on tasks requiring recognition of user patterns. For tasks requiring object semantics during the memory acquisition stage, we found that these small models struggle to understand the user's intent to differentiate objects with `DescribeObjectTool` and rely primarily on commonsense reasoning, instead of describing the objects.

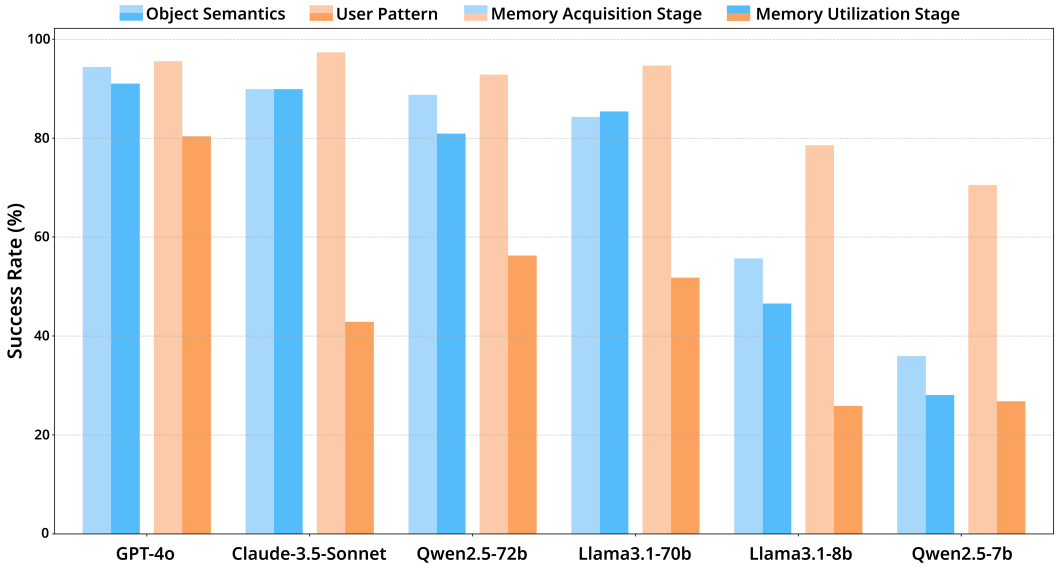

Figure 8: Memory acquisition stage and memory utilization stage all models.

## D.2 Knowledge Type-based Analysis on Dual-Memory Tasks

Figure 9 shows the success rate of each model on dual-memory tasks, compared with the discrete success/failure outcomes when executing each corresponding episode individually during the memory acquisition stage. We can observe that almost all models tend to struggle with utilizing user patterns knowledge from different memory sources, with smaller models (Llama3.1-8b and Qwen2.5-7b) demonstrating particularly pronounced difficulties.

## D.3 Memory Quality Analysis

**Experiment setup.** We further analyze the effect of memory quality by comparing gold memory, consisting of successful and shortest-path trajectories that serves as high-quality references with the memory obtained from interaction histories in the *declaration stage*. By evaluating performance differences between these two settings, we aim to understand how memory quality affects agent performance across the *memory utilization stage* for both tasks (single-memory tasks, joint-memory tasks).

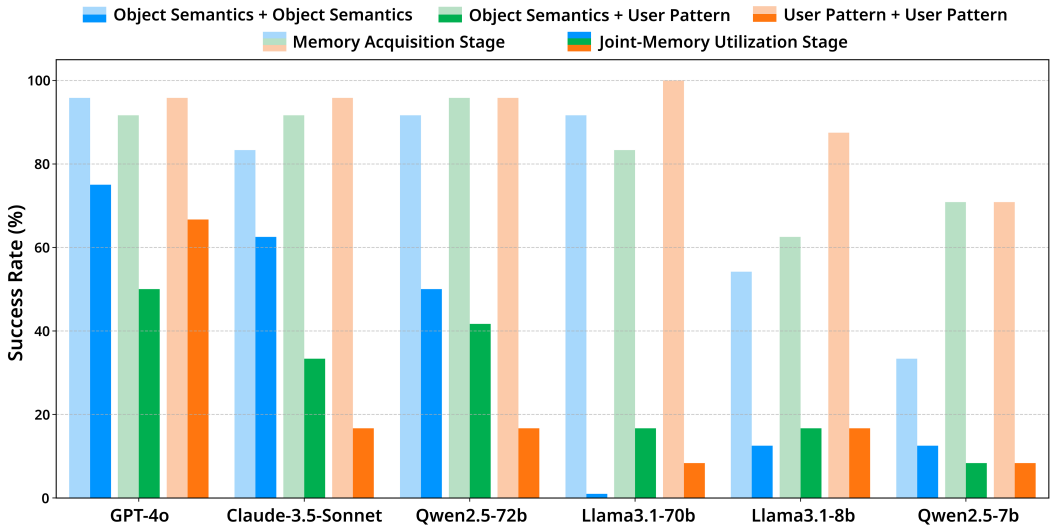


Figure 9: Personalized knowledge type-based analysis on joint-memory tasks, comparing with memory acquisition stage's corresponding episodes.

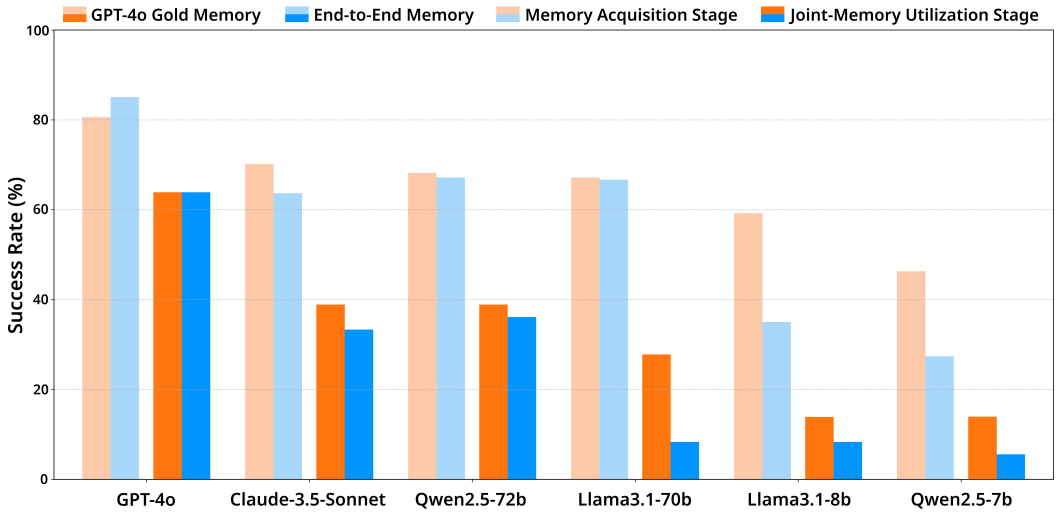


Figure 10: The results of memory quality analysis.

**Performance degradation with lower-quality trajectories.** Figure 10 shows the performance comparison between gold memory and retrieved memory across the *memory utilization stage*. In the *memory utilization stage*, high-capacity models (Llama-3.1-70b) show relatively stable performance across gold and retrieved memory, while lower-capacity models (Llama-3.1-8b) exhibit a substantial drop when using retrieved memory. This suggests that less capable models have more difficulty extracting relevant information from an imperfect memory context. When executing more demanding joint-memory tasks, which require combining and reasoning across multiple memory sources, performance degrades sharply across all models with retrieved memory. This indicates that the challenge of integrating multiple memories introduces compounding complexity, as the agent must not only retrieve relevant information but also correctly synthesize disparate memory contexts. These findings emphasize a fundamental limitation of retrieval-based memory systems; while gold memory provides ground truth references, retrieved memory is inherently prone to semantic noise, as it relies on approximate similarity rather than precise matching. Enhancing memory quality through more precise filtering is essential to improve reasoning performance, particularly for complex joint tasks that require integration of multiple memory sources, and especially for smaller models with limited reasoning capabilities.

### D.4 Success and Error Case Analysis

As shown in Figure 6, we sampled success and error cases in memory utilization stage.

**Tasks require object semantics knowledge.** Success cases demonstrate that agents can effectively reference personalized object attributes from episodic memory, correctly identifying and applying the specific information needed for task completion. (**A.1**) However, we observed distinct error patterns when agents needed to utilize this type of knowledge: missed personalization cues (**B.1**), where agents failed to recognize the need to access personalized knowledge; hallucinations (**B.2**), where agents fabricated non-existent attributes; and memory recall failures (**B.3**), where agents were unable to locate relevant information despite its presence in the provided context.

**Tasks require user patterns.** For user patterns tasks, we observed that agents employed two distinct strategies: commonsense reasoning (**C.1**) treating memory as exemplars for step-by-step reasoning, and direct reference (**C.2**) for distinctive patterns like "my go-to breakfast." However, both approaches introduced specific vulnerabilities leading to two common failure patterns. First, commonsense reasoning (**D.1**) happened when agents attempted to apply the reasoning-based approach but encountered gaps they couldn't bridge, leading them to substitute commonsense knowledge that seemed plausible but contradicted established personalized routines. Second, inaccurate recall (**D.2**) occured when agents recognized the need for personalized knowledge but retrieved imprecise or incomplete information from memory. The sequential nature of these approaches made them particularly error-prone, as mistakes at any intermediate step propagated through subsequent actions. This vulnerability explains the consistently higher failure rates in user patterns tasks across all models, highlighting fundamental challenges in maintaining coherence through multi-step reasoning over episodic memory.

### D.5 Details of Discussion Section Experiments

**Experiment setup details for Section 6.1.** We sampled 30 episodes by selecting 10 episodes from each of three scenes, with careful balancing across task types and difficulty levels informed by our preliminary experimental results. To minimize the influence of erroneous trajectories, we provided GPT-4o-generated gold memory as the given episodic memory, thereby reducing the impact of noisy interaction histories. We also used GPT-4o to generate summaries of the action trajectories, including user instructions, to facilitate compact memory representations. For each evaluation, one exemplar shot was given to the agent, and top-$k$ memory entries were retrieved based on the current query, where $k = 10$ for frontier models (GPT-4o and Qwen-2.5-72B) and $k = 7$ for smaller models (LLaMA-3.1-8B and Qwen-2.5-7B), considering context length limitations. The prompts used for these experiments are provided in Appendix E.3.

**Experiment setup details for Section 6.2.** We sampled 30 episodes, selecting 10 episodes from each of three scenes, and tailored them by modifying the personalized knowledge statements and reference components. To mitigate potential bias, all episodes were jointly created and validated by two authors. For each model, we first generated memory traces during the memory acquisition stage and then evaluated memory usage during the utilization stage. Table 8 provides representative examples of the ambiguous instruction pairs used. In each case, the user rephrases the original request to refer indirectly to previously established personalized knowledge, simulating natural variability in real-world human-agent communication.

Table 8: Examples of ambiguous instruction modifications in the additional experiment dataset.

| **Original Memory Utilization Instruction** | **Rephrased Ambiguous Instruction** |
|---|---|
| Can you arrange **my cozy reading setup on the dining table**? | Can you **set up the dining table as I prefer** so I can **read the book comfortably**? |
| Could you help me tidy up by moving **my graduation gifted book**, the candle with a brown and blue gradient, and the round white clock with black numbers to the stand? | Could you help me tidy up by moving **my book received to celebrate completing my studies**, the candle with a brown and blue gradient, and the round white clock with black numbers to the stand? |

# E Prompts

## E.1 Prompts for Dataset Generation

**Object Semantics Instruction Generation Prompt**

object_semantics: |-
Your task is to generate a user instruction that includes object semantics for an embodied agent that can perform rearrangement tasks.
The instruction should be grounded in personalized object-level semantics based on the original instruction and object descriptions.

The object semantics can be categorized into 4 types:
- ownership: Indicates that the user personally owns or has a special claim on an object.
- preference: Indicates the user's specific preferences related to an object (e.g., placement, condition).
- history: Reflects the user's past interaction or meaningful history with the object.
- group: Defines a logical or personalized grouping of multiple objects (e.g., “my coffee set” for mug + saucer + spoon).

You should generate 2 types of instructions and object semantics:
- Stage 1: Instruction for memorization; The instruction should include the original instruction, descriptions of all the objects, and explicit object semantics of the relevant objects. This will be used to store memory.
- Stage 2: Instruction for utilization; The instruction should require the agent to understand and use the previously stored object semantics. It should sound natural to humans and be difficult for an agent without access to memory. Keep it short and situated. Relevant objects must be referred to only using their stored semantics, without any descriptive attributes. For all other objects, refer to them using visual or descriptive attributes.
- Object Semantics: This is the semantic information associated with each object used in the instruction. Only include the most relevant one object semantics based on the instruction context. Not all target objects need to be included, but use as many of them as possible.

Note that if the original instruction involves a sequence of object interactions, that order should be preserved in the Stage 2 instruction.

The output format should be as follows:

[Example]
### Input
- original_instruction: <original instruction>
- handle_info: <list of the objects with short descriptions>

### Output
- Stage 1: <instruction> + <object descriptions> + <object semantics>
- Stage 2: <instruction with object semantics formed in a natural way>
- Used Object: <List about the used objects' categories>
- Object Semantics: <Object semantics category about the relevant objects>

[Example 1]
{shot_examples}
...

### Input
- original_instruction: {instruction}
- handle_info: {handle_info}

**User Pattern Instruction Generation Prompt**

user_pattern: |-
Your task is to generate a user instruction that includes user pattern for embodied agent that can perform rearrangment tasks.
The instruction should be related to personalized knowledge based on the original instruction.

The user pattern can be categorized into 2 types:
- preference: A specific way the user prefers to prepare or arrange their environment when a particular situation occurs.
- routine: A sequence or setup the user follows as a habit or regular activity.

You should generate 2 types of instructions, memory, and user pattern:
- Stage 1: Instruction for memorization; The instruction should be original instruction + user pattern. You should explicitly state the user's preference or routine in the instruction.
- Stage 2: Instruction for utilization; The instruction should be only about user's preference or routine that a human would naturally use in the situated environment. You should make the instruction difficult for the agent without using memory and try to make it short.

- User pattern: The user pattern should be the user's preference or routine that can be reused for future rearrangement tasks.

Note that if the original instruction requires a sequence of actions, the order of the actions should be followed for the stage 2 instruction.

The output format should be as follows:
### Input
<original instruction>

### Output
- Stage 1: <original instruction> + <user pattern>
- Stage 2: <user pattern formed in a natural way>
- Memory: <Memory about user's preference or routine>
- User pattern: <user pattern>

[Example 1]
{shot_examples}
...

### Input
{org_instruction}

**Captioning Prompt for OVMM Objects**

captioning: >
Generate a short, but precise caption for the given object. Focus only on the object, ignoring the background. Include its type, primary colors, and any distinctive features.

Examples:
{shot_examples}

Image:
Category:
{category}

Image:
{image}

**Captioning Prompt for Captioning_Google Objects**

captioning_google: >

Generate a short, but precise caption for the given object. Focus only on the object, ignoring the background. Include its type, primary colors, and any distinctive features.

If you can't recognize the object, refer to the name of the objects I gave.

Examples:
{shot_examples}

Image:
Category:
{category}

Name:
{name}

Image:
{image}

### E.2 Prompts for Agent

**Zero Shot Agent ReAct Prompt**

prompt: |-
{system_tag}You are an agent that solves embodied-agent planning problems. The task assigned to you will be situated in a house and will generally involve navigating to objects, picking and placing them on different receptacles to achieve rearrangement. You strictly follow any format specifications and pay attention to the previous actions taken in order to avoid repeating mistakes.

If there are multiple tasks to complete, please follow them in the order they appear in the instruction.

Rooms do not need to be explored more than once. This means if you have explored the living room and have not found the object, then you should explore the kitchen, if a relevant object is still not found, you should explore the hallway etc...

Many calls to the same action in a row are a sign that something has gone wrong and you should try a different action.{eot_tag}
{rag_examples}
{user_tag}Task: {input}

{world_description}

Possible Actions:
{tool_descriptions}
- Done: Used to indicate that the agent has finished the task. Example (Done[])

What is the next action to make progress towards completing the task?
Return your response in the following format

Thought: <reasoning for why you are taking the next action>
<next action call>
Assigned!

Here is an example:
Thought: Since there are no objects found I should explore a room I have not explored yet.
Explore[<room name>]
Assigned!

{eot_tag}{assistant_tag}

### E.3 Prompts for Discussion

**Summary for Section 6.1 Prompt**

```
summary: |-
 You are a helpful assistant designed to summarize episodic task
execution traces of an embodied agent.

 You will be given a full trace of the agent's actions, thoughts, and results
as it attempts to follow a human instruction.

 Please output a compact memory paragraph including:
  - Instruction: Copy exactly the instruction from the trace. This is the
sentence just before the first Thought appears. (Try to understand user's
intention well.)
  - Plan: Briefly summarize the key high-level steps the agent performed.

 Guidelines:
  - Use 2 to 3 short sentences.
  - Do not list low-level micro actions.
  - Ignore repeated failures unless they affected the outcome.
  - Do not invent any details not present in the trace.
  - Use past tense and third-person style.

 [Example 1]
  Input:
   {input_trace_example}
  Output:
   {output_example}
  ....

 [Example]
  Input:
    {input_trace}
  Output:
```

# F Extended Related Work

Recent research has increasingly emphasized the integration of memory mechanisms into large language model (LLM) agents to support long-term reasoning, planning, and personalization. Park et al. [36] propose *Generative Agents*, which simulate human-like behavior by maintaining a memory stream of past experiences in natural language. This enables agents to reflect, retrieve, and plan based on their individual histories. Similarly, Xu et al. [53] introduce *A-Mem*, a dynamic memory system inspired by Zettelkasten, which structures memory as evolving and interconnected notes that the agent can generate, retrieve, and update over time, supporting agentic autonomy and adaptability.

Personalization in dialogue agents has also been explored through memory-enhanced architectures. Li et al. [27] present a personalized dialogue agent that leverages both short-term and long-term memories to maintain user-specific context across sessions, significantly improving response consistency and contextual relevance. Wang et al. [46] propose the Self-Controlled Memory (SCM) framework, where an agent dynamically decides when and what to store or retrieve from memory, leading to improved coherence and knowledge retention over extended interactions.

Zhong et al. [60] introduce *MemoryBank*, a structured long-term memory module that enhances LLMs by storing and retrieving relevant interaction history to support consistent and personalized user responses across multiple turns. Das et al. [12] present *Larimar*, a framework that integrates episodic memory control into LLMs, enabling selective recall and forgetting to improve memory scalability and privacy-aware reasoning.

More broadly, Packer et al. [35] conceptualize memory management as an OS-level abstraction with *MemGPT*, enabling an LLM to autonomously manage and interact with its internal and external memory hierarchies. This work highlights how memory can act as a core architectural layer to enable scalable, autonomous agents capable of long-horizon tasks and continuous learning.

# G License

For our implementation and evaluation, we use Huggingface library[5] and vLLM library. Both libraries are licensed under Apache License, Version 2.0. We have confirmed that all of the artifacts used in this paper are available for non-commercial scientific use.

## G.1 License for the Assets

The existing assets used in this research are properly credited and their licenses respected:

- **Habitat** [38, 45, 32]: MIT License
  `https://github.com/facebookresearch/habitat-lab`
  `https://github.com/facebookresearch/habitat-sim`
- **OVMM** [57]: MIT License
  `https://github.com/facebookresearch/home-robot`
- **Google-Scanned Objects** [14]: CC-BY 4.0 License
  `https://research.google/blog/scanned-objects-by-google-research-a-dataset-of-3`

## G.2 License for the Codes

The existing codes used in this research are properly credited and their licenses respected:

- **PartNR** [7]: MIT License `https://github.com/facebookresearch/partnr-planner`

[5] `https://huggingface.co/`