

MemoCue: Empowering LLM-Based Agents for Human Memory Recall via Strategy-Guided Querying

Qian Zhao^a, Zhuo Sun^a, Bin Guo^a and Zhiwen Yu^a

^aNorthwestern Polytechnical University

Abstract. Agent-assisted memory recall is one critical research problem in the field of human-computer interaction. In conventional methods, the agent can retrieve information from its equipped memory module to help the person recall incomplete or vague memories. The limited size of memory module hinders the acquisition of complete memories and impacts the memory recall performance in practice. Memory theories suggest that the person’s relevant memory can be proactively activated through some effective cues. Inspired by this, we propose a novel strategy-guided agent-assisted memory recall method, allowing the agent to transform an original query into a cue-rich one via the judiciously designed strategy to help the person recall memories. To this end, there are two key challenges. (1) How to choose the appropriate recall strategy for diverse forgetting scenarios with distinct memory-recall characteristics? (2) How to obtain the high-quality responses leveraging recall strategies, given only abstract and sparsely annotated strategy patterns? To address the challenges, we propose a Recall Router framework. Specifically, we design a 5W Recall Map to classify memory queries into five typical scenarios and define fifteen recall strategy patterns across the corresponding scenarios. We then propose a hierarchical recall tree combined with the Monte Carlo Tree Search algorithm to optimize the selection of strategy and the generation of strategy responses. We construct an instruction tuning dataset and fine-tune multiple open-source large language models (LLMs) to develop MemoCue, an agent that excels in providing memory-inspired responses. Experiments on three representative datasets show that MemoCue surpasses LLM-based methods by 17.74% in recall inspiration. Further human evaluation highlights its advantages in memory-recall applications.

1 Introduction

Tip-of-the-Tongue (TOT) [3] is a phenomenon that usually occurs in our daily life, where one person has the memory but struggles to recall it immediately. As the pace of society accelerates, this issue becomes increasingly serious, and affects the quality of life. For example, when you need keys but can’t remember where you left them, it can be frustrated and helpless. Therefore, finding a solution to alleviate this issue becomes an urgent need.

With the advancements in the study of human cognition and memory [5], some studies have attempted to assist human memory recall through human-computer interaction. HippoCamera [28] is a smartphone-based memory recall system that helps reactivate memories by replaying memory cues from daily life. Schindler et al. (2022) developed a software to enhance the memory retention by using the social feedback, called interactional behavioral system. Georgiev et al. (2021) explored different types of virtual reality

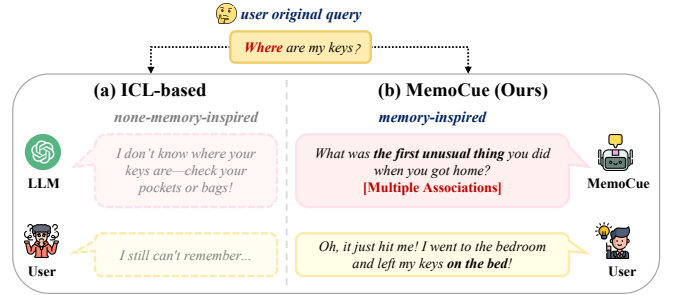


Figure 1: A comparison of forgetting scenario execution with traditional LLM method and our MemoCue agent. (a) Ineffective responses generated by LLM using zero-shot CoT prompting, one In-Context Learning (ICL) method. (b) Strategy-guided responses generated by MemoCue agent, which transforms an original query into a cue-rich one, thereby offering useful memory-inspired information to the user. The recall strategy used is marked in red color.

applications to improve the working memory [2] through immersive experience. These methods mainly rely on the explicit recording and the repeated playback of past experiences to assist the person to recall. The relationship among memory fragments is fixed and it lacks the evolution with adding new memory fragments. The rise of agents equipped with memory modules [45] and large language models (LLMs) [32] presents a promising solution to alleviate the issue. It can perceive the user’s memory context and respond to the user’s query by employing retrieval-augmented generation (RAG) techniques [34]. HippoRAG [12] integrated external information continuously inspired by the neocortex and hippocampus in human memory. LongMem [39] introduced a decoupled network architecture designed for storing and updating long contexts for agent retrieval. While these methods achieve the dynamic organization of stored memory fragments, they only rely on the passive retrieval of the agent’s stored memories to assist the person to recall. The recall performance is determined by the amount of stored memories.

However, when agents are applied to the real world, it is very challenging to acquire and store the entire memory data. First, the acquisition of memory data is limited by temporal and spatial constraints. Real-world memories arise from large-scale and multimodal data sources, such as text [47], video [36], which are often unpredictable in time and space. Furthermore, data collection relies heavily on the trust from users [19] in terms of legal and privacy considerations. Second, the limited storage space of the agent makes it difficult to store such a huge amount of memory information from the users over a large span of time and space. Thus, the incomplete memory

data significantly restricts the agent’s ability to help human recall.

Inspired by the discovery that human forgetfulness is often caused by the ineffective activation of memories rather than the memory loss [6, 30], we formulate a novel **Strategy-Guided Recall (SGR)** problem to achieve the agent-assisted proactive memory recall. Different from the traditional solutions, the agent can strategically guide the user to find associations across diverse memories and activate some memories progressively and achieve the memory recall in SGR, thereby achieving the memory recall. With the guidance strategy, this can alleviate the heavy reliance of memory recall performance on the amount of acquired and stored memories. However, there are various types of memory queries from the user in practice and it requires the unique recall strategy for each memory query type based on its characteristics. Moreover, it is necessary for the recall strategy to be adaptively adjusted to the user’s responses. In addition, the dataset for the SGR problem should include the diverse recall strategies and the appropriate evaluation metrics, which is missing. While the recent LLMs have shown remarkable capabilities in response generation across a wide range of open-domain tasks, their ability to effectively guide human memory recall is still limited. As shown in Figure 1, the response generated by LLM using ICL method exhibits a lack of specificity and recall cues.

To address the aforementioned challenges, we propose a Recall Router framework, as shown in Figure 2. This framework consists of a 5W based scenario query classification and a Monte Carlo Tree Search (MCTS) [21] based strategy response generation. The 5W based scenario classification uses the 5W (*What, Who, Where, When, and Why*) recall map to classify the user queries into five scenarios: Event, Person, Location, Temporal, and Decision. Based on memory theory [13, 10], we define a Recall Strategy Pool for the corresponding scenarios, which includes fifteen potential recall strategy patterns, as illustrated in Figure 3. We propose SGR-MCTS to generate high-quality recall strategy responses, where a fine-grained reward mechanism is designed to evaluate the effect of memory-recall guidance. To validate the effectiveness of proposed Recall Router, we construct MemoStrategy, an instruction tuning recall strategy dataset. At last, by fine-tuning on open-source LLM, we develop MemoCue, an LLM-based agent capable of generating strategy-driven cue queries and providing high-quality recall guidance.

Our contributions can be summarized as follows:

- We propose a Recall Router framework for the formulated SGR problem, aiming to exploit the capacity of LLM-based agents to generate memory-inspired responses and strategically guide the user to activate some memories progressively.
- We design a 5W Recall Map, which classifies user queries to five typical forgetting scenarios, and defines the corresponding recall strategy patterns. Then we propose a SGR-MCTS algorithm to generate effective recall strategy responses, where a fine-grained reward mechanism is designed to evaluate the effect of memory-recall guidance.
- We construct an instruction tuning dataset and develop an LLM-based recall guidance agent MemoCue, which is able to generate strategy-driven cue queries and provide high-quality recall guidance. To evaluate reasonably, we design an evaluation metric which balances the query novelty and response accuracy.
- We conduct extensive experiments on three public memory datasets to validate the effectiveness of MemoCue. To evaluate reasonably, we design an evaluation metric which balances the query novelty and response accuracy. The results demonstrate the superiority of MemoCue in memory recall compared to baselines.

2 Related Work

Human memory, as a fundamental component of human cognition, has long been a topic across various fields [33, 12]. The process of human memory is divided into three stages: encoding, storage, and retrieval, with the retrieval stage refers to the memory recall [37]. Tip-of-the-Tongue (TOT) is frustrated phenomenon which refers to the situation that an individual unable to recall a certain memory immediately even though he know it. Actually, the phenomenon is not only a common occurrence in daily life but also a long-term research focus in cognitive psychology and neuroscience [4].

Previous studies have made some attempts to address the problem and improve human memory recall from the perspective of human-computer interaction [28, 35, 11]. However, these approaches primarily rely on the active recording and repetition of key memories, which fails to account for the unpredictability of forgetting scenarios. Furthermore, their reliance on stored data limits their capacity to guide users actively through the memory recall process.

Recently, LLM-based agent applications gain widespread attention in various fields, such as personal assistant [26], social simulation [24, 18], etc. However, the application of LLM-based agent in assisting human memory recall remains insufficiently addressed due to limited memory data. For example, for smart home agents, only partial event summaries can be stored due to privacy or limited device memory [20]. Many prominent memory theories [7, 10] suggest that forgetfulness arises from a lack effective cues to activate relevant memories, indicating that guiding users to activate key memories is more universally effective than passively relying on incomplete historical memory.

Large language models (LLMs) empowered by in-context learning (ICL) demonstrate remarkable capabilities in generating coherent and contextually relevant responses, even without task-specific fine-tuning [8, 31]. However, they often produce generic or irrelevant responses, as they lack tailored guidance or context-sensitive strategies that are essential for effective memory activation. This limitation becomes particularly evident when users struggle with vague or fragmented memories. Our work, however, considers the cognitive characteristics of human in the memory recall process, enabling agents to offer more targeted and cognitively aligned support in human memory recall, thereby providing a useful assistance and a promising direction for the application of agents in human-centered tasks.

3 Methodology

3.1 Overview

To alleviate the serious dependence of memory recall performance on the amount of stored memories, we formulate the **Strategy-Guided Recall (SGR)** problem, which aims to strategically transform the user’s original query into the memory-inspired cue query and guide the user to activate some memory progressively. For example, given an user original query Q_u , such as “Where are my keys?”, it can be transformed into a cue query Q_c with strategy s_i , such as “Did you do anything unusual after returning home?”, utilizing the “Multiple Associations” strategy. The mapping process can be expressed as Equation 1, where $S = \{s_1, s_2, \dots, s_n\}$ is the strategy set.

$$Q_c = \mathcal{T}(Q_u, S) \quad (1)$$

The essence of solving SGR problem is to optimize the recall strategy selection for the user’s query. To achieve this, we propose a Recall Router framework, which mainly consists of the 5W Recall Map and the SGR-MCTS algorithm, as shown in Figure 2. For

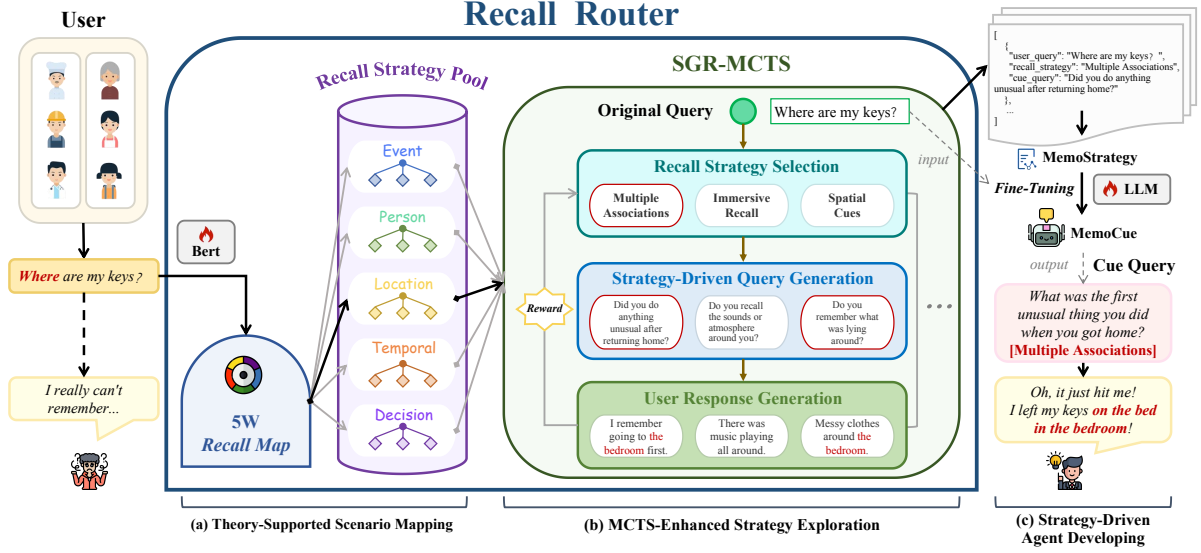


Figure 2: A schematic diagram of the overall memory-recall architecture for SGR problem.

the user’s memory query, the whole memory recall process includes three major steps, where the first two steps are executed in the Recall Router framework: (1) The Theory-Supported Scenario Mapping (Section 4.2) includes the 5W Recall Map and the Recall Strategy Pool. Specifically, when the user query (*none-memory-inspired*) arrives, it is first classified based on its semantic meaning using the fine-tuned RoBERTa model [25]. Details are shown in Appendix B. According to the classified forgetting scenario, the corresponding strategy pattern will be selected from the Recall Strategy Pool. (2) The MCTS-Enhanced Strategy Exploration (Section 4.3) is based on SGR-MCTS algorithm, which includes Recall Tree Construction and Recall Tree Traversal. The former is divided into the high-level strategy selection and the low-level cue query generation, and the latter consists of four key parts: Selection, Expansion, Simulation, and Backpropagation. (3) The Strategy-Driven Agent Developing (Section 4.4) is implemented by fine-tuning LLM with MemoStrategy dataset, which is constructed from the previous two steps (i.e., Recall Router). The developed agent is called MemoCue and capable of generating cue queries (*memory-inspired*) to help the user recall memory.

3.2 Theory-Supported Scenario Mapping

Inspired by Lasswell’s “5W” model of communication [23], we propose the 5W Recall Map for memory recall, where the “5W” refers to What, Who, Where, When, and Why. These correspond to five typical forgetting scenarios: Event, Person, Location, Temporal, and Decision, as shown in Figure 1. Our model classifies user queries based on their type of forgetfulness, identifies recall difficulties, and guides them using appropriate memory-recall strategies. Details of definitions of strategies can be found in Appendix A.

3.3 MCTS-Enhanced Strategy Exploration

For the strategy exploration in SGR problem, we divide it into the high-level strategy selection and the low-level cue query generation. By modeling it as a Hierarchical Markov Decision Process (HMDP), we propose SGR-MCTS to solve it. The process of SGR-MCTS includes recall tree construction and recall tree traversal, which will be

presented in the following. Different from the conventional MCTS algorithm, we construct a hierarchical recall tree and design a fine-grained reward mechanism for SGR-MCTS based on the simulated user feedback. The goal is to obtain the high-quality strategy responses that can guide users to activate memories better.

3.3.1 Recall Tree Construction

The recall tree consists of nodes and edges, where nodes represent the current dialogue state and edges represent the current strategy actions. Based on the characteristics of SGR problem, the recall tree is divided into the high-level recall strategy selection and the low-level cue query generation.

High Level: Recall Strategy Selection. we define the high-level state space S^h as a current context of SGR. The high-level state s^h can be expressed as Equation 2.

$$s_t^h = (Q_u, H_t, M_t) \quad (2)$$

where Q_u is the user original query, H_t is dialog history, including the strategies used, the generated cue queries, and user responses, and M_t is the relevant memory retrieved from user memory bank up to turn t .

The high-level action space A^h consists of 15 recall strategy patterns, which we construct under five typical forgetting scenarios based on human memory theory. The strategy categories and examples are shown in Figure 3. An action a_t^h ($a_t^h \in A^h$) represents a specific recall strategy pattern, which aims to maximize the reward $R^h(s_t^h, a_t^h)$ based on the feedback from the low-level process, reflecting the effectiveness of the recall strategy in guiding user to activate target memories. We define the high-level state transition distribution as $\mathcal{P}(s_{t+1}^h | s_t^h, a_t^h)$, depending on the outcomes of the low-level process, as well as updates of H_t and M_t .

Low Level: Cue Query Generation. The low-level process focuses on generating memory-inspired cue query Q_c based on the selected high-level strategy pattern a_t^h . Therefore, we define the low-level state as $s_t^l = (s_t^h, a_t^h)$, including the high-level context and the selected strategy up at turn t .

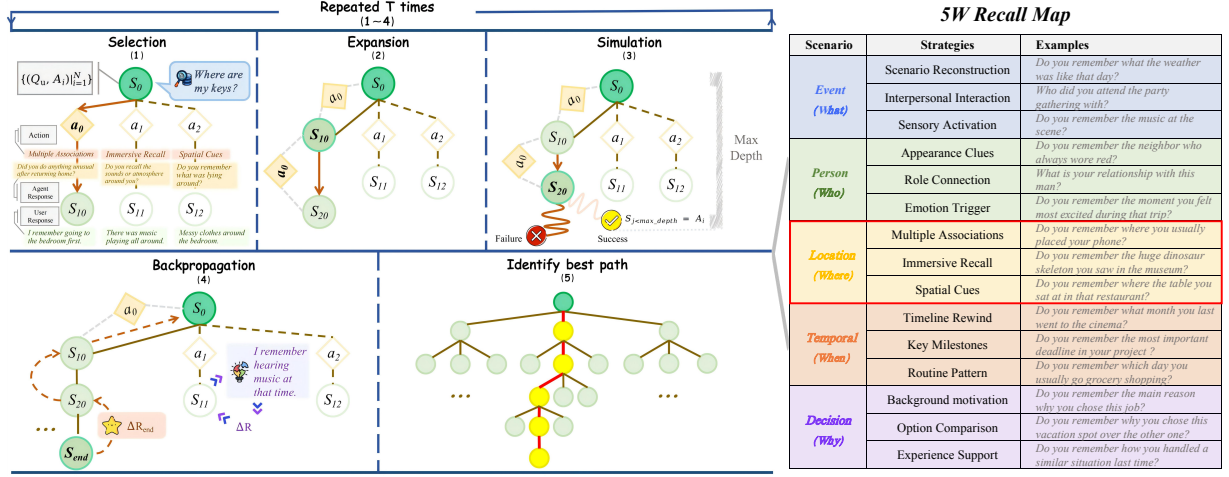


Figure 3: Overview of our proposed SGR-MCTS for memory-recall strategy responses generating. There are four key steps: Selection, Expansion, Simulation, and Backpropagation. Examples illustrating the 5W recall map including five typical forgetting scenarios and corresponding fifteen recall strategies are shown in the right, which constitutes the action space. We perform T iterations and collect the strategy response set on the top- k path for each query.

The low-level action a_t^l represents cue query Q_c , which is generated by LLM (*Qwen2.5-32B-Instruct*). The LLM itself has the characteristic of generating probabilities, where each cue query generation can be viewed as a sample drawn from a conditional probability distribution $a_t^l \sim \mathcal{P}(a | s_t^l)$, where $\mathcal{P}(a | s_t^l) = \text{LM}(Q_c | s_t^l)$. In order to achieve the best user recall performance, we select the $Q_c \in \mathcal{A}^l$ that maximizes the \mathcal{P} . This process can be expressed as Equation 3.

$$a_t^l = \arg \max_{Q_c \in \mathcal{A}^l} \mathcal{P}(Q_c | s_t^l) \quad (3)$$

3.3.2 Recall Tree Traversal

As shown in Figure 3, the recall tree traversal primarily focuses on the high-level process in selecting the optimal recall strategy for each query, which contains four key steps.

Selection. We define the root node as the Q_u (assuming it is none-memory-inspired). Each iteration starts from the root node, and then selects a child node guided by the recognized Upper Confidence Bound applied to Trees (UCT) [21] algorithm, which can balance exploration and exploitation and is expressed as Equation 4.

$$a = \arg \max_{a \in \mathcal{A}^h} (\hat{Q}_i + c \cdot \sqrt{\frac{\ln N(p)}{N(i)}}) \quad (4)$$

where \hat{Q}_i is the average reward for node i , $N(p)$ and $N(i)$ are the visit count for parent node p and node i , c is an exploration constant. The action a that maximizes the UCT value will be selected from the child node with the highest UCT value.

Expansion. When node i is not fully expanded, this phase will be executed. We use the LLM to generate low-level cue query Q_c based on the selected high-level strategy a_t , expressed as $Q_c = f(s_t, a_t)$, where s_t is the current state. After the simulated user generates response r_t based on Q_c , the new state s_{t+1} is updated. The terminal state is reached when r_t exceeds the accuracy threshold or the maximum number of dialog turn is exceeded, which is expressed as

Equation 5.

$$s_{end} = \begin{cases} \text{Success, if} & \text{acc}(r_t, r_{ans}) \geq \theta_{acc} \\ \text{Failure, if} & n_t \geq N_{max} \end{cases} \quad (5)$$

where $\text{acc}(\cdot)$ is the BERTScore [44] based similarity metric, r_{ans} is the true answer, n_t is the number of current dialog turn.

Simulation. To estimate the future rewards, the future situations will be simulated until terminal state after expansion. To improve the effectiveness of recall guidance, we design a fine-grained instant reward mechanism by considering three dimensions: recall accuracy, recall focus, and recall depth.

- **Recall Accuracy** R_{ra} : Evaluation of similarity between r_t and r_{ans} and calculated by BERTScore.
- **Recall Focus** R_{rf} : Evaluation of the consistency of the topic between r_t and r_{ans} , calculated by Jaccard distance[16], expressed as Equation 6.

$$R_{rf} = \frac{|r_t \cap r_{ans}|}{|r_t \cup r_{ans}|} \quad (6)$$

- **Recall Depth** R_{rd} : Evaluation of the level of detail of the user memory reflected in the answer, which is specifically manifested in the richness of the memory elements involved in r_t and can be formulated as Equation 7.

$$R_{rd} = \sum_{i=1}^n \mathbf{1}(e_i \in r_t) \quad (7)$$

where e_i represents the i -th type of memory element in user response r_t , n is the number of memory element types, including *Event*, *Person*, *Location*, *Temporal*, and *Decision*, evaluated by LLM (*Qwen2.5-32B-Instruct*). The indicator function $\mathbf{1}$ takes the value of 1 when the e_i appears in r_t , and 0 otherwise.

To avoid getting trapped in the local optimum, we introduce an exploration factor ε to control randomness. ε is the probability of selecting a random action, while $1 - \varepsilon$ represents the probability of selecting optimal action. We start with $\varepsilon = 1$ and gradually decrease it by 0.05 during the simulation.

Backpropagation. Once the simulation reaches a terminal state s_{end} , the final reward is backpropagated from the leaf node to the root node. During this process, the reward value $Q(s_t)$ and the visit count $N(s_t)$ of nodes will be updated, which can be expressed as Equation 8 and Equation 9.

$$Q(s_t) \leftarrow Q(s_t) + \left(\gamma^{T-t} \cdot R_{end} \right) \quad (8)$$

$$N(s_t) \leftarrow N(s_t) + 1 \quad (9)$$

where $\gamma \in [0, 1]$ is the discount factor, representing the degree of discount on future rewards, T is the total length of the path from the root to the leaf, and R_{end} is the final reward.

We perform T iterations ($T = 120$) on SGR-MCTS to construct and traverse the recall tree, ultimately identifying the top- k recall paths ($k = 5$) with the highest total value.

3.4 Strategy-Driven Agent Developing

To develop the strategy-driven memory-recall agent, we need to construct the appropriate dataset to fine tune LLMs, called MemoStrategy dataset. In particular, based MCTS-enhanced strategy exploration, we record the high-level action sequences, and low-level strategy responses corresponding to the user’s original query as the corpora in MemoStrategy dataset. In the MemoStrategy dataset, each sample contains three core elements: user original query Q_u , memory strategy s_i , and cue query Q_c . Here, Q_u represents the user query regarding forgetfulness, and s_i is determined based on the forgetfulness type identified from Q_u , guiding the subsequent generation of the Q_c using the relevant s_i . We collect a total of 5805 data samples, which are divided into training and testing sets, 5200 and 605 respectively. This instruction-tuning dataset includes three fields: *Instruction* is the task description and provides brief definitions for memory-recall strategies. *Input* is the user original query. *Output* represents the recommended strategy and the predicted cue query response. Details are shown in Appendix C.

We fine-tune LLMs using the MemoStrategy dataset and leverage the function calling capability of LLMs to obtain structured outputs. Based on this, we further develop the LLM-based agent capable of activating user memory through memory-inspired strategies, called MemoCue. Specifically, MemoCue can transform the user original query (*none-memory-inspired*) into a cue-rich one (*memory-inspired*) to activate the user’s memory progressively.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our MemoCue on three representative long-term memory datasets, all of which contain the user personal memory bank, as well as questions and answers for specific memories.

- **PerLTQA** [9]. This dataset includes 141 user memory banks and 8,593 memory question answer pairs. To retrieve memory streams more precisely, we transform each long query entry into multiple short sentences and obtain 35,179 memory streams in the end.
- **LoCoMo** [27]. LoCoMo is a dataset which comprises 50 high-quality ultra-long conversations, where each conversation contains approximately 300 turns and 9,000 tokens. LoCoMo offers the rich personal descriptions to ensure the coherence and long-term consistency of conversations.

- **MemoryBank** [47]. MemoryBank utilizes a dialogue memory storage system in the experiment created over 10 days by 15 virtual users, each possessing distinct personalities. Researchers then create 194 probing questions to assess whether the model could generate relevant responses and recall memories effectively.

Implementation Details. We perform the fine tuning on five open-source LLMs with our constructed MemoStrategy dataset, which includes Qwen2.5-14B-Instruct, Qwen2.5-7B-Instruct [42], LLaMA-3-8B-Instruct [29], Mistral-7B-Instruct [17] and Yi-9B [43]. During the phase, we use a batch size of 8 per GPU and a learning rate of $2e-5$ for all open-source LLMs. We train our model on 8 Nvidia Tesla A100-80GB GPUs with LoRA [14] on 5 epochs.

Baselines. We compare MemoCue with competitive open-source models and closed-source models enhanced by two types of In-Context Learning (ICL) methods: 1) *Standard Prompting*, including Zero-shot Prompting [40] and Few-shot Prompting, which provides the model with a direct prompt to perform a task. In Zero-shot Prompting, the model is directly generated with relying solely on its pre-trained knowledge. In Few-shot Prompting, a small number of demonstrations are provided to instruct the model to obtain the better performance. 2) *Chain-of-Thought Prompting* [41], including Zero-shot CoT [22], and Few-shot CoT [41], which encourages the model to break down complex tasks into intermediate reasoning steps, thereby improving the transparency of the model’s reasoning process and the accuracy of complex reasoning tasks.

- **Open-Source Models.** In order to enhance the reproducibility of experimental results, we choose five open-sourced billion-level LLMs, including Qwen2.5-14B-Instruct, Qwen2.5-7B-Instruct, LLaMA-3-8B-Instruct, Mistral-7B-Instruct and Yi-9B.
- **Closed-Source Models.** In order to comprehensively compare the experimental results, we choose gpt-3.5-turbo, gpt-4 [1], o1-preview [46] and gpt-4o [15] as closed-source models in baselines.

Note that, due to the lack of real-world human memory recall datasets, it is difficult to directly evaluate the end-to-end recall success performance in this work. Instead, we demonstrate the performance of proposed system at the following three aspects: (1) How much can MemoCue improve recall effectiveness, excluding cases where responses merely repeat the original query? (2) How much can MemoCue improve the performance of response in terms of fine-grained content? (3) How much can MemoCue improve the performance of response in real-user experiences?

4.1.1 Evaluation Metrics.

We consider that the agent can guide users to recall the memory by generating memory-inspired cue queries. The quality of generated memory-inspired cue queries determines the memory recalling performance. On one hand, the generated cue query should avoid being too similar to the original query to prevent invalid repetition. On the other hand, it should not stray too far from the topic to avoid affecting the accuracy of the answer. Therefore, we present the metrics for the following three evaluation methods, to evaluate the quality of generated cue queries.

Automatic Evaluation. In order to evaluate the effectiveness of MemoCue, we design an evaluation metric **Balance of Recall Score (BRS)** to balance “the difference between the original query and the cue query” and “the accuracy of memory retrieval”. Specifically, the

Table 1: Performance comparison of different open-source models and settings. The best results are highlight in **bold** and the second-best results are underlined for three datasets, while our results are marked with shading.

Model	Setting	PerLTQA	LoCoMo	MemoryBank	Average
Qwen2.5-14B-Instruct	Zero-shot	<u>71.32</u>	68.04	66.18	<u>68.51</u>
	Few-shot	67.94	68.32	66.30	67.52
	Zero-shot CoT	69.78	<u>70.02</u>	64.13	67.98
	Few-shot CoT	68.77	69.64	<u>66.62</u>	68.34
	MemoCue	78.15	77.89	72.42	76.15
Qwen2.5-7B-Instruct	Zero-shot	69.33	65.00	64.56	66.30
	Few-shot	64.30	66.26	66.27	65.61
	Zero-shot CoT	<u>69.88</u>	67.43	67.11	68.14
	Few-shot CoT	68.88	<u>68.85</u>	<u>67.61</u>	<u>68.45</u>
	MemoCue	76.32	73.59	70.22	73.38
Yi-9B	Zero-shot	<u>70.86</u>	65.52	64.43	66.94
	Few-shot	68.00	67.51	65.13	66.88
	Zero-shot CoT	68.59	<u>70.50</u>	65.29	68.13
	Few-shot CoT	67.74	69.75	<u>66.94</u>	<u>68.14</u>
	MemoCue	73.58	75.71	70.07	73.12
Mistral-7B-Instruct	Zero-shot	58.45	59.13	62.74	60.11
	Few-shot	60.32	60.67	62.88	61.29
	Zero-shot CoT	62.87	64.11	63.41	63.46
	Few-shot CoT	<u>66.33</u>	<u>66.06</u>	<u>65.24</u>	<u>65.88</u>
	MemoCue	69.21	72.01	70.93	70.72
Llama-3-8B-Instruct	Zero-shot	63.78	<u>67.60</u>	65.31	65.56
	Few-shot	60.46	63.15	64.82	62.64
	Zero-shot CoT	<u>68.43</u>	66.81	<u>69.30</u>	<u>68.18</u>
	Few-shot CoT	67.33	66.45	70.31	68.03
	MemoCue	72.65	73.33	71.24	72.41

former is used as a penalty factor and the latter is used as a reward factor, which is mathematically expressed as Equation 10.

$$\text{BRS} = \frac{\text{Acc}(r_{res,i}, r_{ans,i})}{1 + \alpha \cdot \text{Sim}(Q_{u,i}, Q_{c,i})} \quad (10)$$

where $\text{Acc}(\cdot, \cdot)$ is the accuracy of response calculated by BERTScore, $\text{Sim}(\cdot, \cdot)$ is the cosine similarity of two queries. Here, α is a hyperparameter used to adjust the penalty intensity, which is set to 0.3 here.

LLM Evaluation. To achieve a more comprehensive assessment of recall performance, we also employ LLM-based evaluation. This provides the assessment across multiple dimensions, including *recall accuracy*, *recall coherence*, *recall logicity*, *recall inspiration*, and *human-likeness* of the generated cue queries. In particular, we employ OpenAI o1-Preview, the most powerful inference model yet, to evaluate the quality of cue queries from the above dimensions using few-shot prompts. In order to alleviate the position bias of LLM evaluation, we employ the balanced position calibration (BPC) strategy [38]. To ensure the reproducibility of the results, we set the temperature as 0.

Human Evaluation. To evaluate the performance of MemoCue in real-world scenarios, we employ human evaluators to assess the responses generated by MemoCue. To ensure the practicality of MemoCue across different age groups, we employ 9 evaluators from three age ranges: under 30, 30-50, and over 50. Specifically, we first provide the evaluators with specialized training. Then, we present 30 pairs of original queries alongside responses generated by both LLM and MemoCue, with the order randomized. Evaluators rank the response pair with the following dimensions: *recall inspiration*, *response intelligence*, and *cue consistency*.

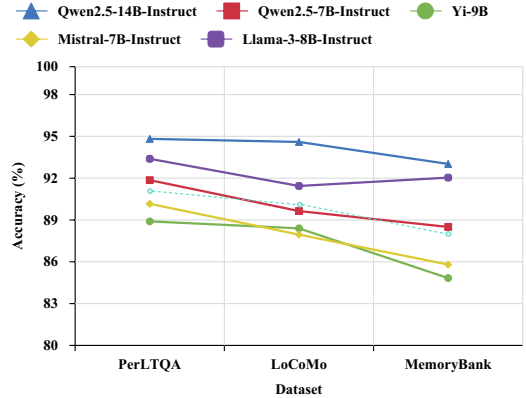


Figure 4: The evaluation results of the accuracy of MemoCue on memory-recall strategy prediction for different LLMs. The dashed line represents the average accuracy within each dataset.

4.2 Experimental Results

We first conduct a series of experiments to investigate the accuracy of MemoCue on memory-recall strategy prediction, which are fine-tuned on different LLMs across three long-term memory related datasets. It can be seen from Figure 4 that MemoCue can achieve the average accuracy of **90.36%** over considered five LLMs. Based on this, we conduct further evaluations.

Automatic Evaluation Results. Table 1 shows the main evaluation results of our experiments. It can be seen that MemoCue achieves the highest BRS among all considered mainstream open-source LLMs across three representative datasets. For example, MemoCue fine-tuned based on Qwen2.5-14B-Instruct improves the BRS by **11.24%**

(compared to zero-shot CoT) on LoCoMo dataset, **9.58%** (compared to zero-shot) on PerLTQA dataset, and **8.71%** (compared to few-shot CoT) on MemoryBank dataset, demonstrating an enhanced ability to generate cue queries that balance the novelty and response accuracy.

To more convincingly demonstrate the superiority of our MemoCue, we compare MemoCue (fine-tuned on Qwen2.5-14B-Instruct) with the leading closed-source LLMs enhanced with zero-shot CoT and few-shot CoT prompting. As shown in Table 2, MemoCue outperforms these models in terms of BRS.

Table 2: Performance comparison of different closed-source models.

Model	Setting	PerLTQA	LoCoMo
gpt-3.5-turbo	Zero-shot CoT	65.43	67.82
	Few-shot CoT	66.34	68.11
gpt-4	Zero-shot CoT	68.54	68.02
	Few-shot CoT	69.76	69.90
gpt-4o	Zero-shot CoT	69.32	68.81
	Few-shot CoT	70.11	69.53
o1-preview	Zero-shot CoT	70.81	68.14
	Few-shot CoT	71.32	69.94
MemoCue		76.32	73.59

LLM Evaluation Results. Figure 5 shows the LLM-based performance evaluation. It can be seen that MemoCue outperforms base-lines generally. For example, MemoCue improves the recall inspiration of response expression by **17.74%**. For the metric of recall logicity, the performance of MemoCue is slightly worse than that of Qwen2.5-14B-Instruct model. This is because the larger-scale parameters enable the model to capture richer logical relationships.

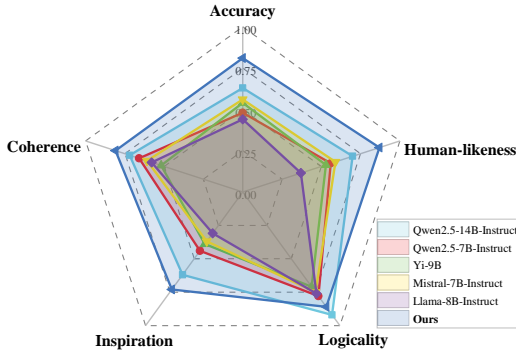


Figure 5: The LLM-based evaluation results.

Human Evaluation Results. Table 3 shows the final win rate distribution on human evaluation. The results show that MemoCue generally outperforms competitive closed-source models in terms of recall inspiration, response intelligence, and cue consistency.

Table 3: Win Rate of MemoCue over competitive closed-source models with HumanEval serving as evaluator.

Evaluator	v.s. gpt-4	v.s. gpt-4o	v.s. o1-preview
HumanEval	83%	85%	78%

4.3 Ablation Study

We also conduct a series of ablation studies to investigate the impact of each module in our proposed framework, as shown in Table 4. The whole framework, incorporating both 5W recall map and SGR-MCTS, achieves the best performance of all three datasets. Removing 5W recall map generally leads to a score drop by 3~4, indicating that distinguishing different scenarios and using corresponding strategies can facilitate the generation of more effective responses. Removing SGR-MCTS module decreases the score by around 4~5, showing that SGR-MCTS can generate high-quality responses by exploiting the simulated user feedback and reward mechanism.

Table 4: The evaluation results of ablation study.

Method	PerLTQA	LoCoMo	MemoryBank	Average
MemoCue	78.15	77.89	72.42	76.15
w/o 5W	↓ 3.54	↓ 3.27	↓ 3.02	↓ 3.28
w/o MCTS	↓ 4.98	↓ 5.13	↓ 4.78	↓ 4.96
w/o 5W&MCTS	↓ 6.83	↓ 9.85	↓ 6.24	↓ 7.64

4.4 Effect of Number of Iterations in SGR-MCTS

We further explore the effect of different number of iterations T in Figure 6, a crucial hyperparameter influencing both the accuracy of the search results and the computational resource consumption. As T increases, the computational time cost grows. Therefore, we set T as 30, 60, 90, 120, 150, and 180 to explore the best. With an increasing number of T , the average score gradually rises and then drops. The peak score occurs when $T = 120$.

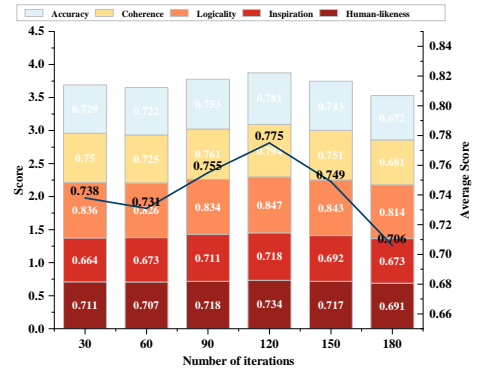


Figure 6: Impact of specifying the number of iterations in SGR-MCTS algorithm on PerLTQA dataset.

5 Conclusion

In this paper, we formulate a novel SGR problem, which aims to empower LLM-based agent for human memory recall. Specifically, we propose a Recall Router framework based on 5W recall map and SGR-MCTS algorithm. We also design a reward mechanism to explore optimal recall strategies. In addition, we construct the MemoStrategy dataset and obtain MemoCue for memory-recall guidance. Furthermore, we define an evaluation metric and conduct extensive experiments on three representative datasets compared with competitive LLM-based methods. The evaluation results show that our MemoCue has improved performance in assisting human memory recall. In the future, we will explore adaptive iteration strategies to improve scalability and conduct a pilot study with real users to further validate the practical utility of our system.

References

- [1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] A. Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.
- [3] R. Brown and D. McNeill. The “tip of the tongue” phenomenon. *Journal of verbal learning and verbal behavior*, 5(4):325–337, 1966.
- [4] D. M. Burke, D. G. MacKay, J. S. Worthley, and E. Wade. On the tip of the tongue: What causes word finding failures in young and older adults? *Journal of memory and language*, 30(5):542–579, 1991.
- [5] S. Chandra, S. Sharma, R. Chaudhuri, and I. Fiete. Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, pages 1–13, 2025.
- [6] A. M. Collins and E. F. Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975.
- [7] F. I. Craik and R. S. Lockhart. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior*, 11(6):671–684, 1972.
- [8] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
- [9] Y. Du, H. Wang, Z. Zhao, B. Liang, B. Wang, W. Zhong, Z. Wang, and K.-F. Wong. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and synthesis in question answering. *arXiv preprint arXiv:2402.16288*, 2024.
- [10] H. Eichenbaum, P. Dudchenko, E. Wood, M. Shapiro, and H. Tanila. The hippocampus, memory, and place cells: is it spatial memory or a memory space? *Neuron*, 23(2):209–226, 1999.
- [11] D. D. Georgiev, I. Georgieva, Z. Gong, V. Nanjappan, and G. V. Georgiev. Virtual reality for neurorehabilitation and cognitive enhancement. *Brain sciences*, 11(2):221, 2021.
- [12] B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*, 2024.
- [13] R. S. Herz. The effects of cue distinctiveness on odor-based context-dependent memory. *Memory & cognition*, 25:375–380, 1997.
- [14] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.
- [15] 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.
- [16] P. Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.
- [17] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- [18] Z. Kaiya, M. Naim, J. Kondic, M. Cortes, J. Ge, S. Luo, G. R. Yang, and A. Ahn. Lyfe agents: Generative agents for low-cost real-time social interactions. *arXiv preprint arXiv:2310.02172*, 2023.
- [19] T. Kandappu, V. Subbaraju, and Q. Xu. Privacyprimer: Towards privacy-preserving episodic memory support for older adults. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–32, 2021.
- [20] A. T. Khan, S. Li, and X. Cao. Human guided cooperative robotic agents in smart home using beetle antennae search. *Science China Information Sciences*, 65(2):122204, 2022.
- [21] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.
- [22] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- [23] H. D. Lasswell. The structure and function of communication in society. *The communication of ideas*, 37(1):136–139, 1948.
- [24] Y. Li, Y. Zhang, and L. Sun. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500*, 2023.
- [25] Y. Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.
- [26] J. Lu, S. An, M. Lin, G. Pergola, Y. He, D. Yin, X. Sun, and Y. Wu. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. 2023.
- [27] A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y. Fang. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*, 2024.
- [28] C. B. Martin, B. Hong, R. N. Newsome, K. Savel, M. E. Meade, A. Xia, C. J. Honey, and M. D. Barense. A smartphone intervention that enhances real-world memory and promotes differentiation of hippocampal activity in older adults. *Proceedings of the National Academy of Sciences*, 119(51):e2214285119, 2022.
- [29] A. Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2024.
- [30] A. S. Meyer and K. Bock. The tip-of-the-tongue phenomenon: Blocking or partial activation? *Memory & Cognition*, 20(6):715–726, 1992.
- [31] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.
- [32] R. OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.
- [33] F. Sarfraz, E. Arani, and B. Zonooz. Sparse coding in a dual memory system for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9714–9722, 2023.
- [34] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024.
- [35] S. Schindler, R. Vormbrock, and J. Kissler. Encoding in a social feedback context enhances and biases behavioral and electrophysiological correlates of long-term recognition memory. *Scientific Reports*, 12(1):3312, 2022.
- [36] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.
- [37] L. R. Squire. Mechanisms of memory. *Science*, 232(4758):1612–1619, 1986.
- [38] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, Q. Liu, T. Liu, and Z. Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023.
- [39] W. Wang, L. Dong, H. Cheng, X. Liu, X. Yan, J. Gao, and F. Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, 2023.
- [40] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [41] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [42] 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.
- [43] A. Young, B. Chen, C. Li, C. Huang, G. Zhang, G. Zhang, H. Li, J. Zhu, J. Chen, J. Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.
- [44] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.
- [45] Z. Zhang, X. Bo, C. Ma, R. Li, X. Chen, Q. Dai, J. Zhu, Z. Dong, and J.-R. Wen. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*, 2024.
- [46] T. Zhong, Z. Liu, Y. Pan, Y. Zhang, Y. Zhou, S. Liang, Z. Wu, Y. Lyu, P. Shu, X. Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024.
- [47] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731, 2024.

A Definitions of Memory-Recall Strategies

We define 15 memory-recall strategies under five typical forgetting scenarios based on human memory theory. Details of the description of the 15 memory-recall strategies are shown in Table 6.

B Details of User Query Classification by fine-tuning ReBERTa Model

We employ GPT-3.5-turbo and utilized few-shot prompting to annotate 2,000 data samples, with the prompt template shown in Figure 7. These samples were manually verified and then used as training data to fine-tune a pre-trained RoBERTa model. The fine-tuning is performed on an Nvidia Tesla A100-80GB GPU, with 500 epochs, a batch size of 32, and the key parameters detailed in Table 5.

Table 5: key parameters set in RoBERTa model

Parameter Name	Value
model_type	roberta
num_hidden_layers	24
hidden_size	1024
num_attention_heads	16
intermediate_size	4096
hidden_act	gelu
attention_probs_dropout_prob	0.1
hidden_dropout_prob	0.1
initializer_range	0.02
layer_norm_eps	1e-5
vocab_size	50265
max_position_embeddings	514
type_vocab_size	1
pad_token_id	1
bos_token_id	0
eos_token_id	2

C Details of MemoStrategy Dataset

C.1 Construction of MemoStrategy Dataset

Strategic Corpus Collection We employ SGR-MCTS algorithm to select appropriate memory-recall strategies and generate corresponding cue queries, as shown in Figure 8. First, by analyzing the user original query, we identify the type of forgetfulness and then choose a suitable memory-recall strategy to generate cue query aimed at activating user’s memory. For instance, when the original query is “When did I take my medication?”, which is related to person, a cue query such as “What time do you usually take your medication?” by using the “routine pattern” memory-recall strategy will be generated accordingly, which activate memory from the perspective of daily habits. The user response will serve as feedback, and its effectiveness is quantified through an immediate reward mechanism. In this way, we can efficiently simulate the memory-recall guidance effect and provide strong support for the optimization of memory-recall strategies data.

Data Quality Optimization. To ensure high standards of data quality, we conduct strict screening and quality control on all memory-recall corpora. By comparing user responses and true answers, we delete invalid samples that failed to successfully activate user memory-recall, that is, scenarios where user responses were unclear or failed to provide valid information. For example, users fail to answer key information in the response (such as dates, people, etc.) in some cases, which are marked as failed samples and remove from the dataset.

Formatting and Instruction Creation. In constructing the MemoStrategy dataset, we first standardize and format the data. Each sample contains three core elements: user original query Q_u , memory strategy s_i , and cue query Q_c . Q_u represents the user query regarding forgetfulness, while s_i is determined based on the forgetfulness type identified from Q_u , guiding the subsequent generation of the Q_c using the relevant s_i . This instruction-tuning dataset includes three fields: `Instruction` is the task description and provides definitions for various memory-recall strategy types. `Input` is the user original query and the recommended strategy. `Output` represents the predicted cue query response. We collect a total of 4805 data samples, which are divided into training and testing sets, 4500 and 305 respectively.

C.2 Examples of MemoStrategy Dataset

Details of the Instruction-tuning MemoStrategy dataset are shown in Figure 9, which takes the user query “What did Zhang Xiaohong and Li Yang admire when they walked around the farm?” for example.

You are an experienced memory-recall guide, and the following is a well-defined set of examples:

- **Event Recall (What):** "What was the game I played at my friend's birthday party?"
- **Person Recall (Who):** "Who was sitting next to me at the wedding?"
- **Time Recall (When):** "When was the last time I went for a medical checkup?"
- **Location Recall (Where):** "Where did I put my keys?"
- **Decision Recall (Why):** "Why did I choose this option at that time?"

Based on the 10 high-quality typical examples provided, please annotate the corresponding user queries for various types of forgetfulness problems. The annotations should be provided *in CSV format*, with the first column containing the questions for different forgetfulness types, and the second column containing the memory scenarios label (*labels include: what, who, when, where, why*).

Example Format:

```
"Query", "Memory Scenario Label"
"What was the game I played at my friend's birthday party?", "what"
"Who was sitting next to me at the wedding?", "who"
"When was the last time I went for a medical checkup?", "when"
"Where did I put my keys?", "where"
"Why did I choose this option at that time?", "why"
"Who was the professor delivering the lecture?", "who"
"Which room did I leave my phone in at home?", "where"
"Why did I decide to turn down that job offer?", "why"
"Where was the location of the last gathering?", "where"
"When did I purchase the car I bought last year?", "when"
```

Now, please annotate the following user queries:

[Here are 2000 user queries need to be annotated (processed in batches of 50)]

Figure 7: The prompt template of user query map to forgetting scenario.

```
Action: Interpersonal Interaction -> Response: [
  state: What did Zhang Xiaohong and Li Yang admire when they walked around the farm?,
  action: Interpersonal Interaction,
  output: What did Zhang Xiaohong and Li Yang say to each other while they were exploring the farm?
]
Action: Sensory Activation -> Response: [
  state: What did Zhang Xiaohong and Li Yang admire when they walked around the farm?,
  action: Sensory Activation,
  output: Do you remember any particular scents, like the smell of fresh hay or earth, that were noticeable during your walk around the farm?
]
Action: Interpersonal Interaction -> Response: [
  state: What did Zhang Xiaohong and Li Yang admire when they walked around the farm?
  action: Interpersonal Interaction
  output: How did Zhang Xiaohong and Li Yang respond when they saw each other's reactions to different parts of the farm?
]
Action: Scenario Reconstruction -> Response: [
  state: What did Zhang Xiaohong and Li Yang admire when they walked around the farm?,
  action: Scenario Reconstruction,
  output: Can you remember if there were any particular moments during the walk when you noticed the play of light through the trees, perhaps casting beautiful patterns on the ground or highlighting certain features of the farm?
]
Action: Scenario Reconstruction -> Response: [
  state: What did Zhang Xiaohong and Li Yang admire when they walked around the farm?
  action: Scenario Reconstruction
  output: Do you remember any specific farm structures, like an old barn or a silo, that stood out and caught your attention during the walk?
]
...
```

Figure 8: The original data generated by the SGR-MCTS algorithm.

Table 6: The description of 15 strategies on five forgetting scenarios.

Scenario	Strategies	Description
Event (What)	Scenario Reconstruction	Activate and enhance memory by simulating or recreating relevant situations, helping user reconstruct memory scenes by recalling details such as surroundings, emotions, time, space, and other contextual elements.
	Interpersonal Interaction	Help user recall specific details of their interactions with others by triggering memory cues linked to particular people or social situations, using both verbal and visual information.
	Sensory Activation	Activate memory by using sensory cues like sight, sound, and smell, particularly when these stimuli are closely linked to the memory of a specific event or situation.
Person (Who)	Appearance Clues	Activate memory by using external features of another person, such as facial expressions and body language, to help recall memories related to that person through recognizing and interpreting their behavior.
	Role Connection	Recall memories linked to an individual based on the user's specific relationship with them in social interactions, such as teacher-student, superior-subordinate, family member, friend, coworker, etc.
	Emotion Trigger	Use emotional responses tied to specific memories, such as happiness, sadness, anger, or surprise, to activate memory by leveraging the strong connection between emotions and memories.
Location (Where)	Multiple Associations	Establish connections between multiple memory points to guide the user, using objects, specific locations, or unrelated events to broaden associative thinking and help recall.
	Immersive Recall	Guide the user to re-experience a situation by using landmarks, paths, or scene details to help recall specific events or experiences from that time and place.
	Spatial Cues	Use spatial information in the environment to trigger memory, helping the user more accurately recall events and details by recalling the layout of objects, buildings, directions, distances, or specific landmarks at a location.
Temporal (When)	Timeline Rewind	Activate memory through simulated time travel, allowing user to retrace past events by recreating or simulating a sequence of occurrences.
	Key Milestones	Guide user to identify key events or turning points in memory, using them as anchors to help locate and recall information along the timeline.
	Routine Pattern	Use regular and habitual behaviors in daily life to trigger memories, linking the user's repeated actions at specific times, places, or situations to memory cues, helping them recall events and details associated with those behaviors.
Decision (Why)	Background motivation	Explore the internal psychological drivers behind the user's actions in a given situation, including intrinsic needs like self-concept, values, and goals, as well as extrinsic factors such as rewards, punishments, or societal expectations.
	Option Comparison	Involve comparing different options during decision-making, helping the user recall their decision-making process by evaluating the pros and cons of each option.
	Experience Support	Assist in remembering the reasons for current decisions by guiding the user to analyze past experiences.

```
[
  {
    "instruction": "Please generate an appropriate memory-recall cue query based on the provided State: ",
    "input": "What did Zhang Xiaohong and Li Yang admire when they walked around the farm? ",
    "output": "What did Zhang Xiaohong and Li Yang say to each other while they were exploring the farm?"
  },
  {
    "instruction": "Please generate an appropriate memory-recall cue query based on the provided State: ",
    "input": "What did Zhang Xiaohong and Li Yang admire when they walked around the farm? ",
    "output": "Do you remember any particular scents, like the smell of fresh hay or earth, that were noticeable during your walk around the farm?"
  },
  {
    "instruction": "Please generate an appropriate memory-recall cue query based on the provided State: ",
    "input": "What did Zhang Xiaohong and Li Yang admire when they walked around the farm? ",
    "output": "How did Zhang Xiaohong and Li Yang respond when they saw each other's reactions to different parts of the farm?"
  },
  {
    "instruction": "Please generate an appropriate memory-recall cue query based on the provided State: ",
    "input": "What did Zhang Xiaohong and Li Yang admire when they walked around the farm? ",
    "output": "Can you remember if there were any particular moments during the walk when you noticed the play of light through the trees, perhaps casting beautiful patterns on the ground or highlighting certain features of the farm?"
  },
  {
    "instruction": "Please generate an appropriate memory-recall cue query based on the provided State: ",
    "input": "What did Zhang Xiaohong and Li Yang admire when they walked around the farm? ",
    "output": "Do you remember any specific farm structures, like an old barn or a silo, that stood out and caught your attention during the walk?"
  },
  ...
]
```

Interpersonal Interaction

Sensory Activation

Interpersonal Interaction

Scenario Reconstruction

Scenario Reconstruction

Figure 9: Examples of the Instruction-tuning MemoStrategy dataset.