Vector Storage Based Long-term Memory Research on LLM

Kun Li
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: 38190985@qq.com

Chengang Jing
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: jcg050980@163.com

Xin Jing
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: jingxin@xatu.edu.cn

***Abstract*—Current large language model (LLM) intelligences face the challenges of high inference cost and low decision quality when dealing with complex tasks, and are especially deficient in maintaining context coherence during long tasks. This research presents an innovative vector storage long-term memory mechanism model (VIMBank) to enhance the long-term context retention ability and task execution efficiency of LLM intelligences by storing and retrieving historical interaction data through a vector database. VIMBank utilizes a dynamic memory updating strategy and the Ebbinghaus forgetting curve theory to efficiently manage the memory of intelligences and reinforce critical information, forgetting unimportant data, and optimizing storage and reasoning costs. The experimental results show that VIMBank significantly improves the decision quality and efficiency of LLM intelligences in multi-tasking scenarios and reduces the computational cost. Compared with different agents, the success rate of task decision is increased by 10% to 20%, and the reasoning cost is reduced by about 23%, which provides an important theoretical basis and practical support for the future development of intelligences with long term memory and adaptive learning ability.**

Keywords-Large Language Model; Long Term Memory; Vector Storage

I. INTRODUCTION

Recent revolutionary advances in large language model-based intelligences have dramatically changed our interactions with AI systems, with LLM intelligences capable of autonomously fulfilling user commands and demonstrating impressive performance in a wide range of tasks. However, LLM intelligences still suffer from the fatal problems of high reasoning cost and low quality of decision making for complex problems. Long-term memory mechanisms aim to improve the decision quality and reduce the reasoning cost of LLM intelligences by storing external knowledge and historical interactions. For example, in conversations that require long interactions with the user, long-term memory helps the intelligent body to maintain contextual coherence, remember the user's historical behaviors and preferences, and provide more accurate or personalized answers, and it can also avoid repetitive reasoning on repeated historical tasks and obtain the reasoning results of similar tasks directly from long-term memory, which saves the arithmetic resources and improves the efficiency of task execution. Therefore, the long-term memory mechanism in LLM intelligent body systems is crucial for maintaining contextual understanding by storing information perceived from the environment and utilizing the recorded information to assist with future instruction tasks that will be performed. This mechanism is one of the important capabilities that have become indispensable for application scenarios such as

DOI: 10.2478/ijanmc-2024-0029

continuous dialogue systems, personalized recommendations, healthcare, and education. Therefore, in order to improve the quality of decision making and reduce the arithmetic resource consumption of LLM intelligences, it is necessary to study a more effective long-term memory mechanism.

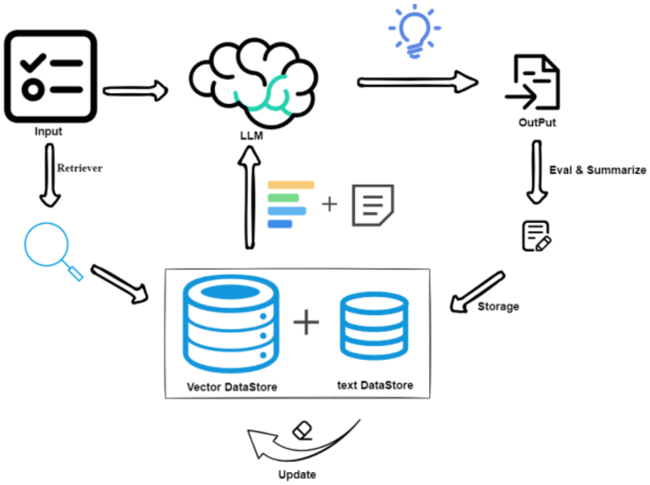


Figure 1. VIMBank Framework

Therefore, this research proposes a long-term memory mechanism model, VIMBank, which aims to improve the decision quality and decision efficiency of intelligences. As shown in Fig. 1, VIMBank uses a vector database as the underlying basic tool to support the storage of historical information, which enables LLM intelligences to effectively store different historical interaction information, such as knowledge information, dialog information, and related task information, which are sliced and vectorized before being stored as long-term memory. The LLM intelligences can effectively store different historical interaction information, such as knowledge information, dialog information and related task information, and different types of information from the interaction process of the intelligences are stored as long-term memory in the vector database after slicing and vectorization, which facilitates the subsequent retrieval and updating of related memory information. In order to realize the accurate and efficient retrieval of different types of information in the decision-making process of the intelligent body, different retrieval strategies are designed for different types of interaction information to effectively improve the recall rate of relevant information from long-term memory in the decision-making process of the intelligent body, so as to enhance the quality of decision-making of the intelligent body's task instructions. Meanwhile, based on Ebbinghaus's forgetting curve theory, which describes the law of human brain's forgetting of new things, VIMBank further combines human's own dynamic memory mechanism for new things, and VIMBank introduces this dynamic mechanism as a long-term memory updating strategy, realizing that the LLM is able to selectively forget the long-term memories with the passage of time and strengthen its memory of the LLM can selectively forget long-term memories over time and enhance the memorization of more frequent memory information. Overall, VIMBank is a memory mechanism that improves the quality and efficiency of decision-making of intelligent body tasks on the basis of storing, retrieving and updating long-term memories.

VIMBank is generic in the sense that it is able to adapt closed-source large models such as ChatGPT, as well as open-source models such as Qwen2-7b, chatglm3-6b, and other models.

This research is based on the ALFWorld [1], HotpotQA [2] and KAgentBench [3] multi-task datasets, and tests the performance of open source large models such as chatglm3-6b and Qwen2-7b in planning and decision-making on different tasks, to understand the proficiency of the large models in understanding, planning and decision-making, and to analyze the problems existing in the large models in the reasoning and decision-making process. Furthermore, based on the large models, this research also considered the performance of the large models on various tasks after giving them the ability to decompose tasks and reflect based on intelligent agent models such as ReACT [4] and InterACT [5]. For example, the success rate of the chatglm3-6b model on the ALFWorld task set reached 62%, and after being equipped with ReACT, the task success rate increased to 70%. Although the large models themselves have certain understanding and reasoning capabilities, experiments have found that in the multi-round task process, due to the limitations of the large model context window, there is a semantic loss phenomenon caused by the overflow of the context window, which leads to the failure of task decision-making. Therefore, the necessity of long-term memory for large language models is verified.

In order to evaluate the effectiveness of VIMBank, this research uses ChatGPT4-o to generate new task datasets similar to existing tasks based on the characteristics of the existing datasets to expand them, so as to verify the effectiveness of long-term memory in the long-trajectory decision quality and multi-round reasoning efficiency of complex tasks in large language models. The experimental results demonstrate the ability of VIMBank in long-trajectory decision quality and multi-round execution efficiency of tasks. The main contributions of this research are summarized as follows:

This research verifies that the semantic environment of large language models is missing in the task reasoning process due to the limitation of context windows.

This research proposes a new long-term memory mechanism VIMBank, which improves the decision quality and efficiency of LLM in multi-round planning reasoning of complex tasks.

This research demonstrates the versatility of the VIMBank mechanism, which can be adapted to existing open-source large models such as Qwen2-7b and chatglm3-6b, as well as current mainstream closed source large models such as ChatGPT3.

II. RELATED WORKS

In recent years, the field of large language models (LLMs) has undergone significant transformation, demonstrating its powerful capabilities in a variety of natural language processing tasks. Models including GPT-3, OPT and FLAN-T5 have achieved outstanding results in multiple fields. At the same time, the latest closed-source models such as PaLM, GPT-4, and ChatGPT continue to demonstrate wide adaptability and gradually become an auxiliary tool for many people's daily decision-making. However, the closed-source nature limits researchers and companies from in-depth study of the internal mechanisms of LLM and hinders the development of applications adapted to specific fields. Therefore, many open source LLM projects have emerged in the community, such as LLaMa, ChatGLM, Alpaca, Vicuna, and Qwen. These models usually contain 6 billion to 14 billion parameters and have achieved remarkable results in multiple benchmark tests.

Nonetheless, these models still have some shortcomings. A significant drawback is that they lack strong long-term memory capabilities. This limitation hinders LLM's ability to maintain context over long periods of time and retrieve relevant information from past interactions. Therefore, in order to improve the decision-making quality and reasoning efficiency of LLM in complex tasks, it is particularly important to research and develop effective long-term memory mechanisms.

With the rapid progress of large language models, researchers have also conducted in-depth research on the context window limit of LLM. There are two main directions involved: one is to adjust the model to increase the context window limit, and the other is to introduce a long-term memory mechanism to enhance the ability of LLM to process long texts through retrieval enhancement. A representative application is the retrieval enhancement generation system based on LLM. For example, Wang et al. [6] proposed a self-knowledge guided retrieval enhancement, which aims to improve the reasoning and generation capabilities of the model by combining external knowledge and LLM's own knowledge, especially in the face of complex problems and task scenarios that require context understanding. Sun et al. [7] proposed a Think-on-Graph method that uses knowledge graphs to provide structured information to guide and optimize the reasoning process of LLM, thereby improving the accuracy and coherence of the generated results. However, the RAG method focuses on external knowledge and is used for tasks that require combining external knowledge to generate answers, but cannot be used to record and manage the historical information of LLM during user interaction. Long-term memory is more like an internal storage mechanism of LLM, which can store external knowledge, past conversation content, task execution history, and other information. This information can be stored and retrieved and used in subsequent interactions with the LLM, thereby helping the agent maintain contextual coherence, improve decision-making quality, and demonstrate higher intelligence in long-term interactions. At present, some research has made preliminary progress. For example, Liu et al. [8] pointed out that when LLM processes long-term tasks, due to the limitation of the context window, it cannot effectively maintain the memory of past information, resulting in poor reasoning and decision-making. Therefore, the research team proposed the Think-in-Memory method to enhance the long-term memory ability of the model through the mechanism of recall and post-thinking. In order to ensure the consistency and contextual coherence of LLM in long-term open dialogues, Lu et al. [9] introduced a memorandum mechanism to enhance the performance of LLM in long-term dialogues. However, the above studies all have high storage and retrieval overhead and correctness problems. At the same time, memory also needs to support dynamic updates to avoid LLM referencing outdated or irrelevant task-related information, thereby affecting the accuracy of task decisions.

In general, although significant progress has been made in the field of LLM in the past two years, long-term memory support is still needed to enhance LLM when persistent interaction is required and accuracy and efficiency are guaranteed in multi-task scenarios. This research uses VIMBank as a new approach to address this challenge.

III. VIMBANK MECHANISM

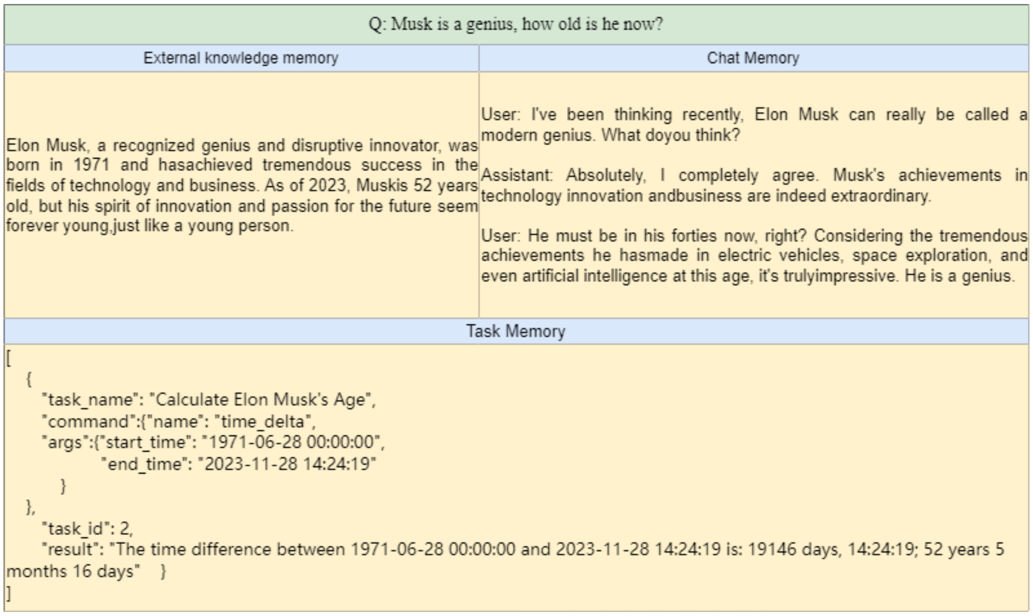
This section first introduces the overall workflow of the proposed long-term memory mechanism framework. Then each stage of VIMBank is described in detail, including the storage of long-term memory, the retrieval of stored information, and the principles of memory updating.

A. Overall framework

1) Task Definition: The purpose of long-term memory is mainly for context retention and decision optimization during the execution of complex tasks in LLM. For example, given a complex task query that requires N steps of reasoning to execute $Q=\left\{s_1,s_2,...,s_N\right\}$, where s_i denotes the first state of the task execution process state, $i\in\left(1,N\right)$; each task state is composed of a series of task context information $S_i=\left\{c_1,c_2,...,c_M\right\}$, where denotes the first state of a task state; each task state is composed of a series of task context information, where the first task context information of a task state, $j\in\left(1,M\right)$. At the same time, based on the long-term memory existing query $Q^{'}$ and historical decision responses $R^{'}$, the parameter pair $p=\left\{(Q_1^{'},R_1^{'}),(Q_2^{'},R_2^{'}),...\right\}$, the new task decision-making process is optimized and excited for each round. Then through the updating principle, the task context information of the current round is updated in the long-term memory, formalized as $F\left(s_i\mid(Q_i^{'},R_i^{'})\right)\rightarrow\left(Q_i,R_i\right)$, where $\left(Q_i^{'},R_i^{'}\right)\in P$.

2) Overview of the framework: Given a task query, the main goal is to enable the LLM to generate more accurate decisions based on previous knowledge and experience, while updating the information from the previous afternoon of the task in each round to form a long-term memory to ensure the contextual coherence of the new upcoming query. The proposed VIMBank enables LLM to retain useful historical information during the processing of multiple rounds of tasks. As shown in Fig. 1, VIMBank is a unified mechanism consisting of three core functional modules: (1) memory storage (2) memory retrieval and (3) memory update.

B. memory storage component



Q: Musk is a genius, how old is he now?

External knowledge memory	Chat Memory
Elon Musk, a recognized genius and disruptive innovator, was born in 1971 and hasachieved tremendous success in the fields of technology and business. As of 2023, Muskis 52 years old, but his spirit of innovation and passion for the future seem forever young,just like a young person.	User: I've been thinking recently, Elon Musk can really be called a modern genius. What doyou think? Assistant: Absolutely, I completely agree. Musk's achievements in technology innovation andbusiness are indeed extraordinary. User: He must be in his forties now, right? Considering the tremendous achievements he hasmade in electric vehicles, space exploration, and even artificial intelligence at this age, it's trulyimpressive. He is a genius.

Task Memory

```
[
   {
      "task_name": "Calculate Elon Musk's Age",
      "command":{"name": "time_delta",
      "args":{"start_time": "1971-06-28 00:00:00",
                  "end_time": "2023-11-28 14:24:19"
         }
   },
      "task_id": 2,
      "result": "The time difference between 1971-06-28 00:00:00 and 2023-11-28 14:24:19 is: 19146 days, 14:24:19; 52 years 5
months 16 days"    }
]
```

Figure 2. Different types of memory

The memory storage component caches task-related context information during the execution of task instructions by the LLM, which is categorized into three different types of memory banks: knowledge memory, dialog memory, and task-related memory. As shown in Figure 2, the knowledge memory mainly captures and stores task-related data and documents that need to be not trained or outdated by external search engines or LLM pre-training; the dialog memory records the query and response pairs in each round of dialogs between the LLM and the user to ensure the context coherence and consistency of the subsequent dialogs, and the task memory records the task decision-making process of the LLM. After each task instruction is given to the LLM, the LLM performs strategic planning through task planning, tool selection, and observations obtained from the tool, and task-related information is systematically stored in memory for effective utilization in subsequent dialogues. In order to improve the validity of historical information during the process of storing LLM interaction information in the memory bank, an evaluation strategy is introduced to ensure that all historical experiences memorized by the LLM are valid, and at the same time, Prompt is pre-set to activate the ability of the intelligent body to recognize positive and negative sample experiences, to ensure that the information memorized by the intelligent body is correct as much as possible. Specifically, when memorizing conversational experiences, positive samples refer to real multi-round conversations in which each round is logically interrelated. On the contrary, negative samples refer to pseudo-multi-round conversations, in which there is no logical correlation between the conversation history and the latest task query, and the intelligentsia does not need to refer to previous conversational experiences when answering the query. When constructing knowledge-based experiences, knowledge experiences need to synthesize possible conflicting, irrelevant and relevant information in the knowledge. Therefore, retrieved knowledge experiences are defined into three categories: relevant, irrelevant and conflicting information. Relevant knowledge refers to experiences from which the answer to a query can be found directly. For example, if a query is made about someone's age and the retrieved knowledge experience contains personal information about the person and his/her age, this part of the knowledge is relevant knowledge. Irrelevant knowledge experiences are those that are related to the task at hand but do not provide a direct answer. Conflicting type of knowledge refers to the existence of two contradictory historical experiences about the same task information in the knowledge experience base, for example, the retrieved knowledge contains two different ages of the same person, which the intelligent should be able to distinguish. The memory of task types is similar to external knowledge memory, which is also likely to have three types of memory: relevant, irrelevant, and conflicting, and the LLM's ability to utilize complex historical experience information to better adapt to different tasks is enhanced by this memory recognition method.

For each type of memorized information text, the text is divided into fixed-length segments. These segments are converted into vector representations on the basis of a vector database for subsequent efficient vector-based retrieval using FAISS. The specific formalization is:

First a piece of text is sliced and divided into n segments $T=\{S_1,S_2,...,S_N\}$ each of which is of length Each segment is of length n. For each text fragment, this research uses CoIBERT [10] as the encoder model to pre-code it into a vector representation, and its vector representation can be expressed in Equation 1:

$$v_i = E(S_i) = \{v_i^0, v_i^1, ..., v_i^m\}, v_i \in \mathbb{R}^d \qquad (1)$$

Where E is the function used to convert text segment into vectors, and v_i is the first i vector representation of the text fragment, $i \in (1 , m)$. Store all the fragment vectors in a shared vector database for subsequent retrieval.

The vector database as the basis of the whole VIMBank is mapped to the vectorized representation of each text using vector indexing, which is able to capture the semantic information of the text by mapping the text to a high-dimensional vector space, but the vector indexing approach is not able to support fast keyword matching, and when the LLM recalls the long-term memory information, on the one hand, the semantic information is very important, and on the other hand, the precision against the keyword matching to the historical memory information should also be captured. As a result, VIMBank needs to use Elasticsearch technology to map text segments and store them as sparse vectors, create an index for each keyword, and record all documents containing the keyword to realize fast processing and precise matching of large amounts of text data.

By mixing the two approaches, vector databases map a large number of text segments as dense vectors, and Elasticsearch stores text maps as sparse vectors, and in the subsequent retrieval phase, vector indexes are good at semantic understanding, while Elasticsearch-based backward indexes are good at fast and accurate keyword matching, while utilizing the advantages of both to obtain more comprehensive and accurate retrieval results.

C. Memory Retrieval Component

The quality of the relevance of information retrieved from long-term memory can have a critical impact on LLM reasoning and decision-making in the process of LLM executing reasoning and decision-making, so the retrieval mechanism is a crucial stage in long-term memory.

Based on text memory vectorization and keyword storage, VIMBank adopts FAISS [11] retrieval method for densely embedded vectors and BM25 retrieval model for sparse vectors [12] for these two storage methods respectively. The core idea of FAISS is to generalize, partition, or quantize high-dimensional vectors so as to reduce the search space and improve the retrieval speed, which can be achieved through a combination of various indexing techniques such as inverted files, product quantization, HNSW, and GPU acceleration to achieve efficient retrieval on large-scale datasets. And keyword storage uses the Elasticsearch search engine to represent these text segments as sparse vectors through inverted indexing, of which the BM25 retrieval model is the most commonly used inverted indexing-based retrieval model, which not only considers word frequency and inverse document frequency, but also introduces factors such as lexical item saturation, which better balances the impact of high-frequency words. VIMBank improves the information retrieval speed by integrating the vector retrieval and the keyword retrieval results to improve the overall effectiveness of information retrieval. This process involves merging and de-duplicating the results from the two retrieval strategies. With these two retrieval techniques, LLM is able to achieve more comprehensive and accurate information recall.

On the other hand, in order to ensure the recall of information related to LLM retrieval and the current task, VIMBank designs different retrieval strategies based on three different types of memory information, namely external knowledge, dialog, and task. Each memory type has its own adapted retrieval mechanism during each round of interaction in LLM.

For knowledge experience, this research wants to increase the number of recall results to access more relevant information. In a vector database, documents and instructions are represented as high-dimensional vectors. Vector databases enable semantic similarity search by storing and retrieving these high-level vectors. The vector similarity of different texts can be defined by approximate nearest search method using cosine similarity distance function, which is implemented as:

$$dist(q,d)=\frac{q\cdot d}{\|q\|\|d\|} \tag{2}$$

In Equation 2, q is the query vector, d is the document vector, $\|q\|$ is the parameter of vector q . and $\|d\|$ is the parameter of vector d . Suppose that $q=[q_1,q_2,...,q_n]$ and $d=[d_1,d_2,...,d_n]$, then the distance function of cosine similarity is specified as:

$$dist(q,d)=\frac{\sum_{i=1}^{n}q_i d_i}{\sqrt{\sum_{i=1}^{n}q_i^2}\cdot\sqrt{\sum_{i=1}^{n}d_i^2}} \tag{3}$$

Cosine similarity can measure whether two vectors are in the same direction or not without considering the size of the two vectors, this method focuses more on the relative importance of the words and can maximize the retrieval of relevant document information for similar words. In contrast, knowledge-experience ES retrieval can increase the number of returned results by tuning the retrieval parameters.

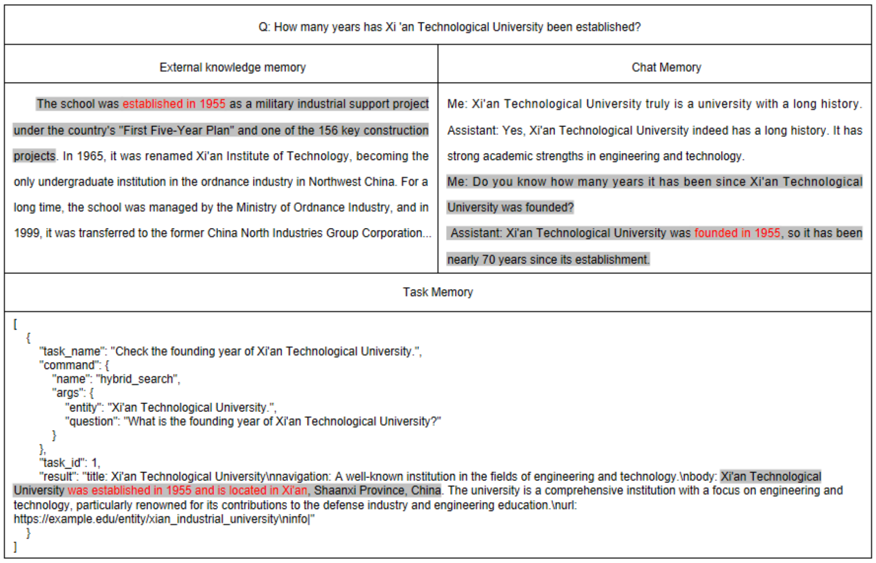
For conversational experience, ensure recall of conversation rounds that are relevant to the query context. The main performance is semantically identical and logically coherent. Therefore, for conversational memory retrieval focuses more on pre-temporal weighting and content semantic similarity. ES retrieval introduces temporal weighting, while semantics can be realized directly based on vectorized retrieval, and the quality of retrieval recall for conversational information needs to consider both kinds of retrieval results comprehensively:

$$S_{call}=\omega\cdot\alpha+(1-\omega)\cdot s \tag{4}$$

In Equation 4, α and s are ES scores and similarity scores, ω is the time weights, assuming that the current time is T and the timestamp of the dialog round is t , and the timestamp of the conversation round as:

$$\omega=e^{-\lambda(T-t)} \tag{5}$$

For task memory, the focus is on recalling recent task-relevant information with increased temporal weighting. The retrieval strategy is similar to conversational memorization. For example, for the query "What year was Xi'an University of Technology founded?", the information retrieved from the memory bank with different memory types is shown in Figure 3. The background color marked in gray indicates the part of the retrieval process related to the query, while the text marked in red indicates the information related to the text and the answer to the query. As the dialog progresses, the session memory and task memory are continuously updated to ensure that the LLM can effectively handle dynamically changing dialog situations and task requirements. This design not only enhances the flexibility of the LLM, but also improves the efficiency and accuracy of its application in practical tasks.



Q: How many years has Xi 'an Technological University been established?	
External knowledge memory	Chat Memory
The school was established in 1955 as a military industrial support project under the country's "First Five-Year Plan" and one of the 156 key construction projects. In 1965, it was renamed Xi'an Institute of Technology, becoming the only undergraduate institution in the ordnance industry in Northwest China. For a long time, the school was managed by the Ministry of Ordnance Industry, and in 1999, it was transferred to the former China North Industries Group Corporation...	Me: Xi'an Technological University truly is a university with a long history. Assistant: Yes, Xi'an Technological University indeed has a long history. It has strong academic strengths in engineering and technology. Me: Do you know how many years it has been since Xi'an Technological University was founded? Assistant: Xi'an Technological University was founded in 1955, so it has been nearly 70 years since its establishment.
Task Memory	

```
[
   {
      "task_name": "Check the founding year of Xi'an Technological University.",
      "command": {
         "name": "hybrid_search",
         "args": {
            "entity": "Xi'an Technological University.",
            "question": "What is the founding year of Xi'an Technological University?"
         }
      },
      "task_id": 1,
      "result": "title: Xi'an Technological University\nnavigation: A well-known institution in the fields of engineering and technology.\nbody: Xi'an Technological
University was established in 1955 and is located in Xi'an, Shaanxi Province, China. The university is a comprehensive institution with a focus on engineering and
technology, particularly renowned for its contributions to the defense industry and engineering education.\nurl:
https://example.edu/entity/xian_industrial_university\ninfo|"
   }
]
```

Figure 3. Retrieval memory

D. Memory Update Component

Through the memory storage and retrieval mechanism, LLM can break through the limitations of the context window and obtain contextually coherent or relevant task information from long-term memory, and the LLM memory capacity is greatly enhanced. However, as time goes by, there may be a large amount of redundant information in the LLM long-term memory, resulting in too much data in the long-term memory, which will greatly affect the retrieval efficiency of the entire VIMBank mechanism, and may even cause LLM decision errors due to outdated and inaccurate information. Therefore, in order to ensure the correctness and efficiency of information in long-term memory, a memory update mechanism needs to be designed.

To address the above two problems, this research was inspired by the forgetting curve theory proposed by Ebbinghaus, and designed the human brain's law of forgetting new things in the updating strategy of LLM long-term memory. According to the forgetting curve theory, clearing those secondary memory segments that occurred long ago and have not been frequently recalled can avoid the LLM memory module from occupying too much memory, and also avoid the influence of old and outdated information on the LLM decision-making. The LLM long-term memory updating strategy is mainly guided by the following principles: (1) memory forgetting; (2) speed of forgetting; and (3) review effect. Memory forgetting aims to simulate the decline of human memory over time, forgetting speed aims to reflect the rate of forgetting of different frequency and importance of information over time, and the review effect aims to express that the steepness of the forgetting curve can be effectively slowed down when the learning content is regularly reviewed or memorized over and over again to enhance the durability of memory.

The Ebbinghaus forgetting curve can be described by an exponential decay model:

$$R(t) = R_0 \cdot e^{-\frac{t}{\lambda}} \tag{6}$$

In Equation 6, $R(t)$ denotes the memory retention rate at time t, that is, the proportion of information retained, t denotes the time that has elapsed since the information was learned, e is approximately equal to 2.71828. λ denotes the parameter of forgetting rate, which controls the speed of memory decline and is affected by factors such as learning depth and number of repetitions. In order to simplify the process of memory updating. The λ modeled as a discrete value and initialized to 1 at the time of the first memory. when a memory segment is retrieved by the LLM decision-making process will increase the forgetting rate parameter of the segment pair by 1, and will reset to 0, which makes the segment less likely to be forgotten in the future and thus retained in memory for a longer period of time.

In summary, VIMBank builds a more comprehensive LLM long-term memory mechanism through the entirety of these key components. This mechanism improves the decision quality of LLM while reducing the reasoning cost, providing new possibilities for LLM applications.

IV. Experiments

A. Experimental Environment

The experimental environment is shown in Table 1.

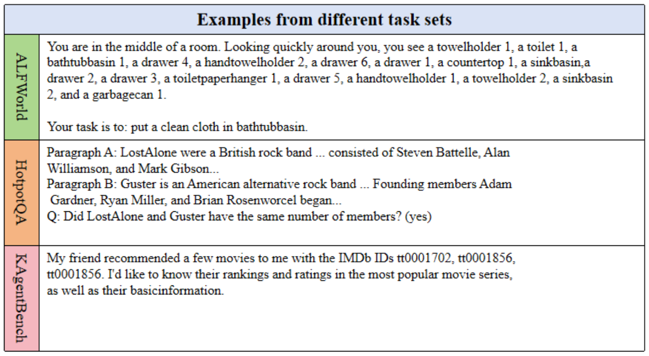
TABLE I. EXPERIMENTAL ENVIRONMENT

Experimental Environment	Version
CPU	Intel Core i9-10900K
GPU	NVIDIA Tesla V100 PCIe 32G
Language	Python 3.9
Framework	LangChain

B. Task Set

In order to verify the effectiveness of the long-term memory mechanism of VIMBank, this research conducted experiments on three different task sets, ALFWorld, HotpotQA, and KAgentBench.

The ALFWorld dataset combines task execution in virtual environments and natural language instruction comprehension, and contains about 8,000 test tasks covering a wide range of scenarios and complex tasks. HotpotQA is a dataset for studying and evaluating complex question-answering tasks, especially multi-step reasoning tasks, and contains 7405 test samples and is commonly used as a benchmarking dataset for testing models on complex problems. KAgentBench is a benchmarking tool for evaluating the capabilities of knowledge-based intelligences, which contains multiple tasks that cover different knowledge-based reasoning capabilities, decision-making capabilities, as well as natural language understanding and generation capabilities. The above three textual task sets are shown in Figure 4:



	Examples from different task sets
ALFWorld	You are in the middle of a room. Looking quickly around you, you see a towelholder 1, a toilet 1, a bathtubbasin 1, a drawer 4, a handtowelholder 2, a drawer 6, a drawer 1, a countertop 1, a sinkbasin,a drawer 2, a drawer 3, a toiletpaperhanger 1, a drawer 5, a handtowelholder 1, a towelholder 2, a sinkbasin 2, and a garbagecan 1. Your task is to: put a clean cloth in bathtubbasin.
HotpotQA	Paragraph A: LostAlone were a British rock band ... consisted of Steven Battelle, Alan Williamson, and Mark Gibson... Paragraph B: Guster is an American alternative rock band ... Founding members Adam Gardner, Ryan Miller, and Brian Rosenworcel began... Q: Did LostAlone and Guster have the same number of members? (yes)
KAgentBench	My friend recommended a few movies to me with the IMDb IDs tt0001702, tt0001856, tt0001856. I'd like to know their rankings and ratings in the most popular movie series, as well as their basicinformation.

Figure 4. Examples from different task sets

C. Benchmark Evaluation

In order to assess the effectiveness of VIMBank, the evaluation follows the principle of "Unity of knowledge and action", which integrates planning decisions and corresponding actions at each step.

$$S_{plan} = \frac{1}{M}\sum_{j=1}^{M} \max EM\left(T_{n,i}, T_{n,j}^{'}\right) \cdot \Gamma\left(T_{h,i}, T_{h,j}^{'}\right)\left(1 \le i \le N\right) \tag{7}$$

$$S_{action} = \frac{1}{M}\sum_{j=1}^{M} \max EM\left(T_{n,i}, T_{n,j}^{'}\right) \cdot \sum_{k=1}^{K_i} EM\left(a_{k,i}, a_{k,j}^{'}\right) \cdot \Gamma\left(v_{k,i}, v_{k,j}^{'}\right) \tag{8}$$

In Equation 7 and Equation 8, M is the number of complex tasks decomposed into subtasks by the LLM and N is the number of real decisions made by the LLM. $T_{n,i}$ is the number of i true decision, and $T_{n,j}^{'}$ is the prediction result of the j-th subtask. Γ is the ROUGE-L evaluation metric function to measure the similarity between the decision results and the true results. Specifically, ROUGE-L is based on the principle of the longest common subsequence, and measures the recall and precision of the decision results through F1-Score comprehensively.

$$R=\frac{LCS\left(T_{n,i},T_{n,j}^{'}\right)}{|T_{n,i}|} \tag{9}$$

$$P=\frac{LCS(T_{n,i},T_{n,j}^{'})}{|T_{n,j}^{'}|} \tag{10}$$

$$F1=\frac{2\times R\times P}{R+P} \tag{11}$$

The calculation based on the longest common subsequence can well capture the order information between sequences and is suitable for measuring the similarity between the generated sequences and the desired results in the decision-making process.

EM (Exact Match) is another evaluation metric to assess the exact match between the generated text and the reference text. When evaluated using EM, it tends to 1 if the generated text (R) and the reference text (G) are basically the same, and tends to 0 if most of the text is different from the reference text, which can be expressed as:

$$EM(R,G)=\begin{cases}1, if\ \ R\approx G\\0, if\ \ R\neq G\end{cases} \tag{12}$$

In order to comprehensively assess the performance of LLM in decision making and execution, a penalty factor is introduced p ∈ {0 , 1}, the comprehensive assessment formula is as follows:

$$S_{total}=(1-p)\cdot(0.7\cdot S_{plan}+0.3\cdot S_{action}) \tag{13}$$

D. Results

In order to evaluate the effectiveness of VIMBank in improving the decision quality and reducing the reasoning cost on multi-tasks, this research replicates the performance of some intelligences on task datasets based on some open source LLMs and compares VIMBank with the replication results. The performance of various LLMs on task reasoning decisions is shown in Table 2.

TABLE II. EXPERIMENTAL RESULTS ON VARIOUS DATASETS

DataSet	Model	NoAgent	ReAct	InterAct	VIMBank
ALFWorld	Qwen2-7b	48.8	54.7	60.1	72.3
	ChatGLM3	46.2	49.2	55.8	64.9
HotpotQA	Qwen2-7b	51.6	57.3	63.4	76.3
	ChatGLM3	45.9	51.8	59.7	71.5
KAgentBench	Qwen2-7b	34.2	48.5	52.6	58.7
	ChatGLM3	32.6	44.7	46.3	54.2

The experimental results show that compared to ReAct and InterAct, which incorporate short-term memory, the LLM with the introduction of VIMBank's long-term memory mechanism improves the quality of decision making on different tasks, which is still a gap compared to the original paper that uses a closed-source LLM such as ChatGPT, which may be due to the overall poorer capability of the open-source models compared to ChatGPT (Qwen2-7b and ChatGLM) are less capable overall. In order to analyze the effectiveness of long-term memory in enhancing LLM decision-making, experiment observed the ALFWorld task environment when LLM performs the pick 2 task during which LLM needs to find two identical items, such as "find two books and put them in bookshelf. "Since only short-term memory can forget the previous position due to the limitation of the context window, which leads to task failure, especially in the case of performing the same task in the same environment, it is more likely to cause inefficiency in task execution, and the advantage of long-term memory mechanism in the similar task environment comes to the forefront. token test on different LLMs, with NoAgent as the benchmark, this research recorded the token consumption of LLMs with the process of multiple rounds of tasks, which effectively proved the effectiveness of VIMBank in reducing the cost of LLMs' reasoning for multiple rounds of tasks, as shown in Table 3.

TABLE III. REASONING COST OF ALFWORLD ENVIRONMENT

	200	600	1000
NoAgent	63.2K	164.7K	334.7K
VIMBank	56.8K	142.6K	258.3K

In addition, this research observes the reasoning trajectories of different intelligences in the ALFWorld environment as shown in Fig. 3, which is analyzed to show that the introduction of the long-term memory mechanism can guide the new reasoning steps through previous experiences, thus avoiding repeated reasoning and improving the retrieval efficiency to a large extent.

V. CONCLUSIONS

In this paper, this research presents a new long-term memory mechanism, VIMBank, to enhance LLM context-awareness without model fine-tuning, to improve decision quality, and to reduce the inference cost of similar tasks with the help of long-term memory. This research evaluates this mechanism in a variety of different task environments, and it significantly improves in decision accuracy compared to some baseline models, by more than 10% compared to the InterAct model built on the ReAct model, and by more than 20% compared to the NoAgent. Meanwhile, in terms of LLM reasoning cost, NoAgent reasoning cost basically grows in a linear trend, while our VIMBank mechanism reduces the reasoning cost by nearly 25% as the task is executed, through the accumulation of long-term memories and specific retrieval and update mechanisms. This highlights the great potential of VIMBank in LLM-driven AI systems.

REFERENCES

[1] SHRIDHAR M, YUAN X, CÔTÉ M A, et al. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning[J]. 2021. arXiv:2010.03768.

[2] YANG Z, QI P, ZHANG S, et al. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering[C]Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium. 2018:2369-2380.

[3] PAN H, ZHAI Z, YUAN H, et al. KwaiAgents: generalized information-seeking agent system with Large Language Models[J]. 2023. arXiv:2312.04889.

[4] YAO S, ZHAO J, YU D, et al. ReAct: Synergizing Reasoning and Acting in Language Models [J]. 2023. arXiv:2210.03629.

[5] CHEN P L, CHANG C S. InterAct: Exploring the Potentials of ChatGPT as a Cooperative Agent [J]. 2023. arXiv:2308.01552.

[6] WANG Y, LI P, SUN M, et al. Self-Knowledge Guided Retrieval Augmentation for Large Language Models [J]. 2023. arXiv:2310.05002.

[7] SUN J, XU C, TANG L, et al. Think-On-Graph: Deep and Responsible Reasoning of Large Language Model with Knowledge Graph [J]. 2023. arXiv:2307.07697.

[8] SHEN Y. Think-in-Memory: Recalling and Post-Thinking Enable LLMs with Long-Term Memory [J]. 2023. arXiv:2311.08719.

[9] LU J, AN S, LIN M, et al. MemoChat: Tuning LLMs to Use Memos for Consistent Long-Range Open-Domain Conversation [J]. 2023. arXiv:2308.08239.

[10] KHATTAB O, ZAHARIA M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT[J]. SIGIR, 2020:39-48.

[11] JOHNSON J, DOUZE M, JEGOU H. Billion-scale similarity search with GPUs [J]. IEEE Transactions on Big Data, 2021: 535-547.

[12] AKLOUCHE B, BOUNHAS I, SLIMANI Y. BM25 Beyond Query-Document Similarity[M/OL]//Lecture Notes in Computer Science. String Processing and Information Retrieval. 2019: 65-79.