HMem: A Hawkes Process Memory Framework for Long-Horizon Agents

Zixuan Yan[1], Xuanyi Wu[2], Jinling Wei[2*]
[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
yanzixuan@zju.edu.cn
[2]School of Computer and Computing Science, Hangzhou City University, Hangzhou, China
2240101035@stu.hzcu.edu.cn, weijl@zucc.edu.cn

***Abstract*—In recent years, large language models (LLMs) have demonstrated increasingly remarkable capabilities, but they still have limitations in dealing with long context. Therefore, the LLMs-powered agents still have deficiencies in long-horizon scenarios. Recent studies on the memory of agents can alleviate this problem. In the context-aware agent memory, how to effectively organize and retrieve memories is a problem. Inspired by the Hawkes process, its self-excitation property can simulate the suddenness and associative mechanism of human reasoning, and its time-decay property can simulate human memory management, we proposed HMem, an agent memory framework, which combine Hawkes process and human cognitive mechanism. It consists of Dual Intensity Kernel Memory(DIKM), which distinguishes between working memory and factual memory and applies different intensity kernel functions, and a Hawkes-Enhanced Scoring Engine(HESE), which combines semantic similarity and temporal logic for better retrieval. In this way, HMem enables agents effectively filter out noisy memories and maintain behavioral consistency in long-horizon scenarios. Experiments on the LoCoMo dataset show that HMem outperforms baselines expecially in Single Hop and Multi Hop tasks, which demonstrated the effectiveness and practical value of our method.**

***Index Terms*—Large Language Model, Agent Memory, Hawkes Process**

I. Introduction

The rapid development of large language models(LLMs) has demonstrated remarkable capabilities [1]. People utilize their powerful generation and reasoning abilities to solve various problems. However, they are still constrained by the length of the context [2], [3]. While recent advancements in long-context LLMs have expanded context windows to 128k or even 1M tokens, they fundamentally suffer from quadratic computational complexity, the 'Lost in the Middle' phenomenon and length extrapolation. Even to this day, these issues have not been properly resolved.

As a way of using LLMs, LLMs-powered agents can more effectively utilize the capabilities of the LLMs [4]. At the same time, such agents are also subject to long context. To solve this problem, more and more research is beginning to focus on improving the performance of agents through the memory mechanism. On one hand, the context bottleneck of the LLMs difficult to be resolved in a short time. On the other hand, how to effectively organize the content generated by the interaction between the agent and the environment into the memory is also an ability that an agent should possess for a long period of time. Currently, the research on agent memory is mainly divided into two types: parametric and non-parametric [5]. The parametric approach fixes the memory management ability to the large model or external parameters through fine-tuning or reinforcement learning [6]–[8]. The advantage of this approach is that it runs faster and to some extent truly learns a certain memory organization ability or integrates context into model parameters directly, but the disadvantage is that it requires training time, and this method may lead to model degradation, catastrophic forgetting, etc. If the balance is not well managed, it may reduce the basic ability of the model. And even if the training time is optimized as much as possible, re-training is still necessary when switching between different models [9]. The advantage of the non-parametric approach [10]–[13] is that it can adapt to diverse environments without training, but the disadvantage is that there is a part of memory organization that is more time-consuming and has poor interpretability. There is no single approach that can unify community opinions. The methods and memory mechanisms [14]–[16] employed in other works have complex structures, such as high-dimensional organization. In contrast, our approach is simple in understanding and effective.

In this paper, we propose a novel agent memory framework HMem, which has better interpretability and effectiveness in long-horizon scenarios. The motivation is inspired by the Hawkes process [17] and the human cognitive mechanism [18]. We use LLMs to distinguish different types of memories and then mark them. During the retrieval process, we employ a mechanism that combines semantic similarity and temporal logic to identify the memories that are more likely to be helpful for the current question. The Hawkes process is a self-exciting point process. The main idea is that each event occurrence increases the probability of related events (include itself) occurring in the near future. The self-exciting property makes the Hawkes process intermittent and clustered, which is similar to the associative memory mechanism of humans. That is, when we recall a certain concept, related concepts will emerge densely within a short period of time. The kernel function decay mechanism of the Hawkes process, such as the exponential decay kernel function, perfectly fits the Ebbinghaus forgetting curve [19], and can simulate the temporality of working memory. The power-law decay function have a certain long-tail effect that matches the long-term memory in

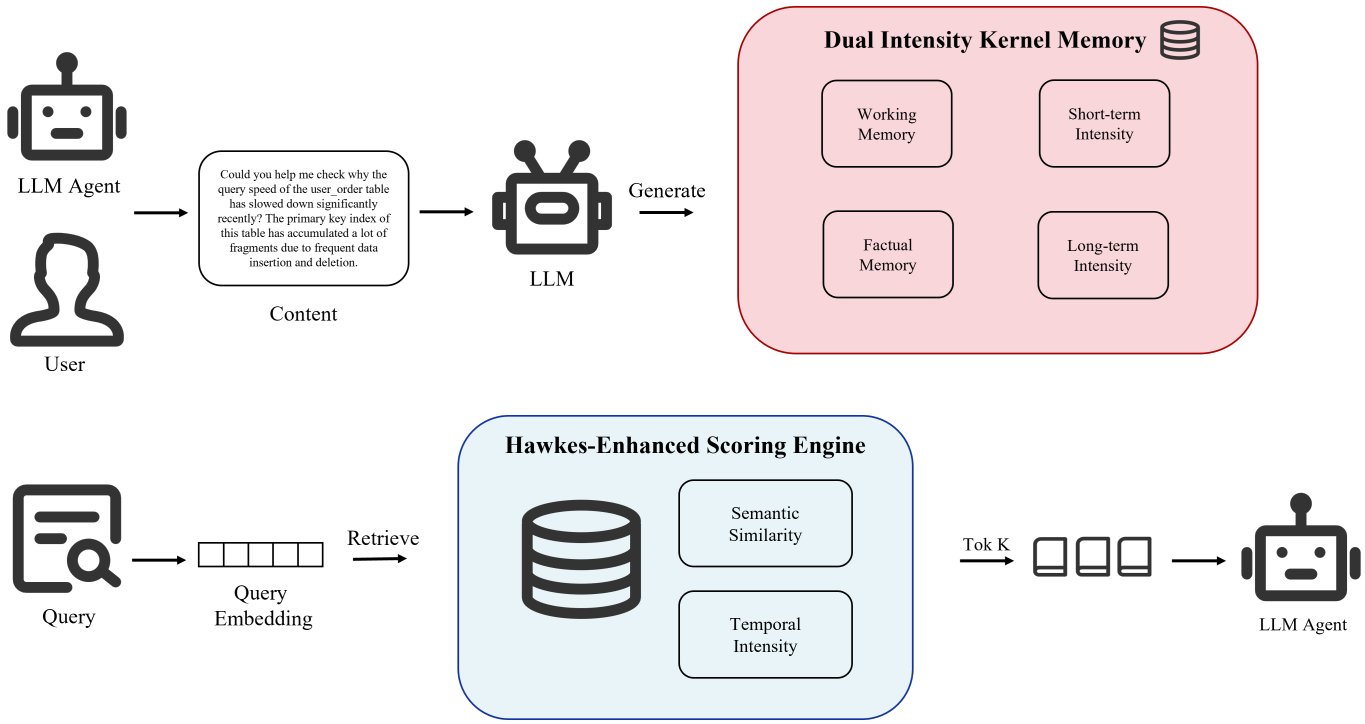

Fig. 1. HMem is capable of better handling dynamic memory by differentiating between working memory and factual memory, and it integrates semantic and temporal logic during retrieval.

cognitive mechanism. The conditional intensity function part of the Hawkes process has the effect of intensity superposition, which perfectly matches the repetition effect in human memory. The strength of memory existence is directly related to the number of awakenings.

In summary, our main contributions can be summarized as follows:

- We propose HMem, an agent memory framework, which is mainly composed of two modules, Dual Intensity Kernel Memory and Hawkes-Enhanced Scoring Engine, and is designed to enhance the reasoning ability and behavioral consistency of long-horizon agents.
- We combined the Hawkes process with human cognitive mechanism to achieve this framework, which has excellent interpretability.
- We conducted thorough experiments and ablation study in the LoCoMo dataset, and the results demonstrated the practical value of our method.

II. Related Work

A. Hawkes Process & Human Cognitive Mechanism

Observations made over time reveal natural aggregation phenomena. For instance, earthquakes usually increase the geological tension in the area where they occur, and aftershocks are likely to follow. The Hawkes process is such a mathematical model used to capture these self-excited processes:

$$\lambda(t) = \mu + \sum_{t_i<t} \phi(t - t_i) \tag{1}$$

where $\lambda(t)$ denotes the conditional intensity function at time t, representing the instantaneous likelihood of a memory activation event. The parameter $\mu \geq 0$ is the base intensity, capturing spontaneous events that are exogenous to the process. The term t_i represents the timestamp of the i-th historical interaction ($t_i < t$), and $\phi(\cdot)$ is the kernel function that quantifies the decaying influence of these past events on the current state.

The core lies in the fact that the likelihood of subsequent events increases within a certain period after the initial event occurs. In past research, there have been Long Short-Term Memory networks based on the Hawkes process [20], which model how past events can influence the future in a complex and realistic manner. In our work, we also measured how past memory affects the future.

Although the exploration of the brain mechanism by humans is still at an early stage, there are some very practical scientific ideas. The spreading-activation theory of semantic processing [21] indicates that the semantics have a diffusive nature. The Ebbinghaus forgetting curve [22] shows that the decay pattern of the memory retention over time. The spacing effect [23] in learning shows that conducting spaced review when memory is about to be forgotten will slow down the decay rate.

B. LLMs and Agent Memory

Nowadays, the LLMs-powered agents can be classified into three types of memory in terms of storage form: token-level, parameter-level, and latent memory-level. In terms of the stored content, they can be divided into factual memory,

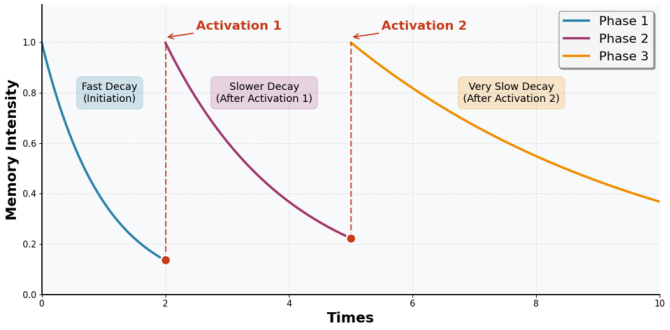

Fig. 2. Illustration of the spacing effect: repetition slows down memory decay as we used in β_{wm}. Each activation point resets Memory Intensity and reduces decay rate

experiential memory, and working memory. In our work, we will focus on token-level factual memory and working memory [5]. In this work, we primarily focus on token-level memory storage. Regarding content, we target two key categories: factual memory, which maintains interactional consistency (e.g., identities, stable preferences, and historical commitments), and working memory, which supports higher-order cognition by actively managing task-relevant information. Consequently, our memory system is organized into explicit and discrete units. These units remain externally visible and structured, allowing them to be individually accessed, modified, and reconstructed over time.

III. Methodology

A. Problem Formalization

In the interactions between the agent and the environment, we define the content of interactions as C. We define the current question or task issued by the agent as q_i. The process of generating content by the LLMs is referred as LLM, and the retrieval process of the memory framework is defined as $retrieve$, and the response of the agent is defined as $response_i$. Therefore, the content we need to optimize becomes the following formula:

$$response_i = LLM(retrieve(C, q_i), q_i) \tag{2}$$

B. Dual Intensity Kernel Memory

Dual-Intensity Kernel Memory(DIKM), a module that gracefully receives the two refined memory streams produced by the LLMs: the volatile working memory traces wm_j and the enduring factual memory fm_j. And unified under the symbol m_j, each memory shard is timestamped at its content instant t_k. For the working memory, we invoke an exponential decay kernel intensity function:

$$\lambda_{wm}(t) = \mu_{wm} + \alpha \sum_{t_i<t} exp(-\beta_{wm}(t - t_i)) \tag{3}$$

where $\mu_{wm} \geq 0$ is the base intensity (or background rate) of working memory, and $exp(-\beta_{wm}(t - t_i))$ is the kernel function that quantifies the decaying influence of past contents on the current state. In contrast, the power law kernel $exp(-\beta_{fm}(t - t_i))$ is tailored for the long-term memory. Its intensity function is defined as follows:

$$\lambda_{fm}(t) = \mu_{fm} + \sum_{t_i<t} K_{fm}/(C_{fm} + (t - t_i))^p \tag{4}$$

where $\mu_{fm} \geq 0$ is the base intensity (or background rate) of long-term memory, and $k_{fm} \geq 0$ is the scaling factor that controls the intensity of the power law kernel. $c \geq 0$ is the constant that ensures the kernel function is always non-negative, and $p \geq 1$ is the exponent that controls the rate of decay. Its Markovian nature mirrors the rapid fade of fleeting context. For the steadfast facts destined for long-term retention we instead adopt a power law kernel; its heavy tail echoes the persistence of human long-term recall, ensuring that salient truths remain luminous even after prolonged silence. These two complementary kernels give rise to a pair of time intensity functions that jointly orchestrate the ebb and flow of passed influence. The contrast of above two kernels can be illustrated in Figure 3.

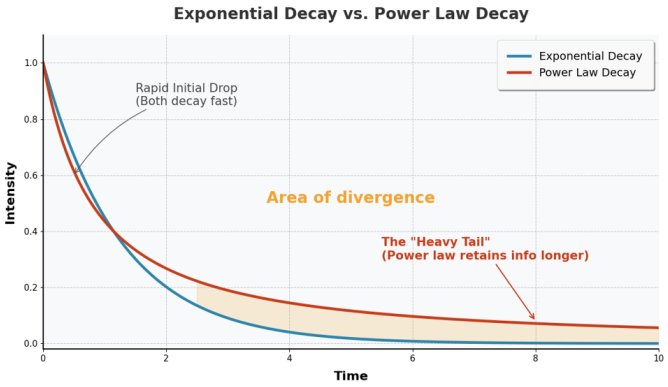

Fig. 3. Illustration of the exponential decay and power law decay

C. Hawkes-Enhanced Scoring Engine

When conducting the specific query for current question, the similarity between the query q_i and the memory m_i, as well as the specific time intensity function, will be taken into account to select the top k m_i. The final result formula is as follows:

$$Score_{ij} = \text{sim}(q_i, m_j) \times \lambda_{m_j}(t_k) \tag{5}$$

where $\text{sim}(q_i, m_j)$ is the semantic similarity between the query q_i and the memory m_j, and $\lambda_{m_j}(t_k)$ is the time intensity function of the memory m_j at time t_k. Specifically, we will convert the query into a vector form v_q through the embedding model, and then use cosine similarity to calculate the semantic similarity of the query and memory, which are also in vector form v_m. The formula is as follows:

$$\text{sim}(q_i, m_j) = \frac{v_q \cdot v_m}{||v_q|| \times ||v_m||} \tag{6}$$

D. Utilization of Memory

After obtaining the topk m_j with the highest scores, we concatenate them and q_i to form the final output based on the LLMs. Thus, the LLMs-powered agent generates the final response.

$$response_i = LLM(q_i, \bigcup_{j=1}^{k} m_j) \tag{7}$$

Prompt Template

Role: You are the Memory Encoder for the HMem system. Analyze the session and extract memory events into two categories based on decay patterns.

1. Working Memory (Targeting Exponential Kernel):

- *Definition:* Transient context, current specific intent, and temporary constraints.
- *Decay:* Fades rapidly (Short-term relevance).

2. Factual Memory (Targeting Power-law Kernel):

- *Definition:* User attributes, explicit preferences, and enduring facts.
- *Decay:* Fades slowly (Long-term relevance).

Output Schema (JSON):

```
{
  "working_memories": [
    // List of immediate contexts (e.g.,
    "Debugging code related to Hawkes Process")
    "WM_Event_1", "WM_Event_2", ...
  ],
  "factual_memories": [
    // List of enduring facts (e.g., "User is a
    researcher using PyTorch framework")
    "FM_Event_1", "FM_Event_2", ...
  ]
}
```

Input Session:

Fig. 4. The prompt template used to generate working memories and factual memories for the DIKM.

IV. Experiments

A. Experimental Setup

In order to evaluate the effectiveness of HMem in long-term multi-conversation scenarios, we used LoCoMo dataset [24], which was designed to assess the capabilities of large models in long contexts. LoCoMo consists of 50 high-quality very long conversations, each containing an average of 300 turns and 9K tokens over up to 35 sessions. The questions in LoCoMo are classified into the following five types: (1) **Single-hop** questions require answers based on a single session; (2) **Multi-hop** questions require synthesizing information from multiple different sessions; (3) **Temporal** reasoning questions can be answered through temporal reasoning and capturing time-related data cues within the conversation; (4) **Open-domain** knowledge questions can be answered by integrating a speaker's provided information with external knowledge such as commonsense or world facts; (5) **Adversarial** questions are designed to trick the agent into providing wrong answers, with the expectation that the agent will correctly identify them as unanswerable.

TABLE I
Detailed HyperParameters Configuration

Description	Symbol	Value
Working Memory Exponential Decay	β_{wm}	dynamic
Working Memory Excitation	α	0.1
Factual Memory Power Decay	p	1.2
Factual Memory Offset	C	0.1
Factual Memory Scaling Factor	K_{fm}	0.1
Base Intensity	μ	1
Token K	$topK$	10

We use **LoCoMo**, **MemoryBank** and **AMem** as the comparison baselines. LoCoMo directly uses the LLMs to conduct the question answering task test without attaching any agent memory mechanism. MemoryBank introduces a memory management mechanism and performs dynamic memory updates based on the Ebbinghaus forgetting curve theory. Drawing inspiration from the Zettelkasten method, AMem introduces a agent memory system, enables LLMs-powered agents to dynamically organize and evolve their memories. We use F1 and BLEU-1 as evaluation metrics. F1 combines accuracy and recall rates and is more reliable:

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{8}$$

precision and recall are calculated as follows:

$$\mathrm{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{9}$$

$$\mathrm{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{10}$$

While BLEU-1 compares the responses word by word with the answers:

$$\text{BLEU-1} = BP \cdot p_1 \tag{11}$$

where p_1 is the unigram precision, calculated as:

$$p_1 = \frac{\sum_{r \in \{response\}} \sum_{gram_1 \in r} \text{Count}_{\text{clip}}(gram_1)}{\sum_{r \in \{response\}} \sum_{gram_1 \in r} \text{Count}(gram_1)} \tag{12}$$

and BP is the brevity penalty defined as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases} \tag{13}$$

TABLE II
RESULTS ON LOCOMO DATASET OF QA TASKS HAVE FIVE CATEGORIES: SINGLE HOP, MULTI HOP, TEMPORAL, OPEN DOMAIN, AND ADVERSIAL. OPTIMAL PERFORMANCE IS IN **BOLD**. RESULTS ARE BASED ON F1 AND BLEU-1 SCORES.

Method	Single Hop		Multi Hop		Temporal		Open Domain		Adversarial	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Backbone: Qwen3-1.7B										
LoCoMo	32.73	28.46	30.78	27.64	21.53	17.84	40.95	37.17	5.35	4.61
MemoryBank	29.35	24.26	27.86	29.21	23.77	19.64	45.41	39.18	5.31	4.12
AMem	41.35	34.26	39.86	32.21	**37.45**	**39.31**	42.53	37.07	7.36	4.92
HMem	**46.25**	**39.16**	**47.84**	**41.43**	25.83	21.36	**47.29**	**43.94**	**8.56**	**5.95**
Backbone: Qwen3-4B										
LoCoMo	29.49	24.39	23.91	19.78	15.54	11.97	45.84	39.67	10.88	9.73
MemoryBank	31.26	27.86	27.21	21.77	17.64	12.76	45.41	39.18	7.31	4.95
AMem	40.96	32.47	42.86	33.87	**29.77**	**22.64**	48.65	39.18	5.31	4.12
HMem	**41.64**	**35.13**	**34.14**	**27.78**	19.89	14.58	**57.55**	**48.73**	**14.22**	**11.73**

TABLE III
WE CONDUCTED AN ABLATION STUDY TO EVALUATE HMEM. THE NOTATION 'W/O' INDICATES EXPERIMENTS WHERE SPECIFIC MODULES WERE REMOVED.

Method	Single Hop		Multi Hop		Temporal		Open Domain		Adversarial	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Backbone: Qwen3-8B										
LoCoMo	39.47	34.67	33.61	29.89	17.45	13.98	59.95	53.13	22.77	19.55
w/o HESE	42.23	37.56	35.81	31.04	18.35	14.96	62.87	56.67	28.48	24.29
w/o DIKM	40.42	36.96	33.37	29.92	18.57	14.71	64.05	58.39	29.03	26.27
HMem	**47.83**	**41.98**	**39.92**	**32.61**	**21.73**	**18.55**	**66.58**	**61.92**	**31.93**	**26.84**

B. Implementation Details

We use **Qwen3-1.7B/4B** [25] as the base model. For memory generation, we set the temperature to 0.4, and for question answering, we set the temperature to 0.7. In the process of generating memories, we use Qwen3-8B as the LLM backbone. In the retrieval, we select the tok-10 memories with the highest scores as the context for the agent. To balance efficiency and performance, we use **all-MiniLM-L6 v2** [26] as the embedding model. For the hyperparameters in DIKM, based on experience and for the convenience of research, our specific settings are as shown in the tableI.

C. Main Results

In our experiment, the HMem framework was compared with LoCoMo, MemoryBank and AMem on the LoCoMo dataset. On the model based on Qwen3-1.7B/4B, HMem outperformed the LoCoMo and MemoryBank in all categories of tests. Our HMem outperformed AMem in the Single Hop and Multi Hop task and show similar performance in Temporal, Open Domain and Adversarial. From a holistic perspective, the experimental results demonstrated the effectiveness of our memory mechanism, especially in Single Hop and Multi Hop tasks, which were remarkable. Next, we will analyze from the perspective of tasks. For Single Hop and Multi Hop tasks, this effectiveness stems from the improvements we made to the generation of memory and retrieval. For the Open Domain tasks, these methods we tested did not show any significant results. For the Temporal and Adversarial tasks, HMem did not significantly outperform AMem. Overall, HMem significantly enhances the performance of LLM-powered agents in evaluation by applying the Hawkes process to the agent memory. In addition, we also find that for Qwen3, with larger parameters does not necessarily perform better in long-horizon scenario.

D. Ablation Study

We conducted ablation studies by removing components from the HMem framework to evaluate the effectiveness of DIKM and HESE. As shown in Table III, when these two key components were removed, the system performance declined to varying degrees in evaluation metrics. The performance degradation was more significant in Single Hop and Multi Hop tasks, confirming their crucial role in the memory mechanism. And in other tasks, this decline was less significant, indicating that the improvements in these tasks are not as significant as those mentioned above.

To verify the effectiveness of the DIKM in HMem framework more directly, we conducted a t-SNE visualization analysis on the memory snapshots during the framework's operation. The figure5 showed that DIKM can perform a more granular division of memories, enabling the system to focus more on the key information than LoCoMo.

V. CONCLUSION

In this study, we proposed HMem, a novel framework for organizing the memory of LLMs-powered agents, aiming to alleviate the limitations present in long-horizon scenarios. Starting from the Hawkes process and human cognitive

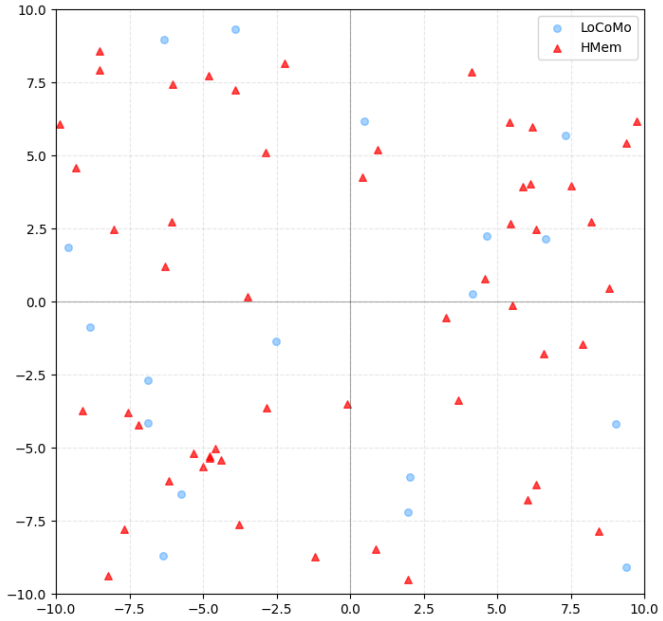

Fig. 5. T-SNE visualization of text embeddings demonstrate that HMem has more detailed memories and presented a certain clustered distribution structure.

mechanism, we innovatively propose Dual Intensity Kernel Memory and Hawkes-Enhanced Scoring Retrieval. This design effectively enhances the performance of agents in long-horizon scenarios from the perspective of agent memory framework. Experimental results show that HMem outperforms the baselines on various tasks on the LoCoMo dataset. This study demonstrates that our well-designed agent memory framework can effectively alleviate the weaknesses of LLMs in long context, as the bottleneck of LLMs-powered long-horizon agents.

VI. Acknowledgments

This work is partially supported by the Second Batch of Undergraduate Provincial Teaching Reform Project of the 14th Five-Year Plan of Zhejiang Province, China (JGBA2024676), and the Teaching Reform Project of Zhejiang Province, China (No. JGCG2025196).

References

[1] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2025. [Online]. Available: https://arxiv.org/abs/2402.06196

[2] J. Liu, D. Zhu, Z. Bai *et al.*, "A comprehensive survey on long context language modeling," 2025. [Online]. Available: https://arxiv.org/abs/2503.17407

[3] DeepSeek-AI, "Deepseek-v3 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2412.19437

[4] J. Luo, W. Zhang, Y. Yuan *et al.*, "Large language model agent: A survey on methodology, applications and challenges," 2025. [Online]. Available: https://arxiv.org/abs/2503.21460

[5] Y. Hu, S. Liu, Y. Yue *et al.*, "Memory in the age of ai agents," 2026. [Online]. Available: https://arxiv.org/abs/2512.13564

[6] Y. Wang, X. Liu, X. Chen, S. O'Brien, J. Wu, and J. McAuley, "Self-updatable large language models by integrating context into model parameters," in *ICLR*, 2025. [Online]. Available: https://openreview.net/forum?id=aCPFCDL9QY

[7] H. Pouransari, D. Grangier, C. Thomas, M. Kirchhof, and O. Tuzel, "Pretraining with hierarchical memories: separating long-tail and common knowledge," 2025. [Online]. Available: https://arxiv.org/abs/2510.02375

[8] X. Yu, C. Xu, G. Zhang, Z. Chen, Y. Zhang, Y. He, P.-T. Jiang, J. Zhang, X. Hu, and S. Yan, "Vismem: Latent vision memory unlocks potential of vision-language models," 2025. [Online]. Available: https://arxiv.org/abs/2511.11007

[9] J. Cao, J. Wang, R. Wei, Q. Guo, K. Chen, B. Zhou, and Z. Lin, "Memory decoder: A pretrained, plug-and-play memory for large language models," 2025. [Online]. Available: https://arxiv.org/abs/2508.09874

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

[11] W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y. Zhang, "A-mem: Agentic memory for LLM agents," in *NeurIPS*, 2025. [Online]. Available: https://openreview.net/forum?id=FiM0M8gcct

[12] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "Memorybank: enhancing large language models with long-term memory," in *AAAI*, ser. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29946

[13] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, "Mem0: Building production-ready ai agents with scalable long-term memory," 2025. [Online]. Available: https://arxiv.org/abs/2504.19413

[14] M. Hu, T. Chen, Q. Chen, Y. Mu, W. Shao, and P. Luo, "HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model," in *Proceedings of ACL 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., 2025, pp. 32 779–32 798. [Online]. Available: https://aclanthology.org/2025.acl-long.1575/

[15] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," 2025. [Online]. Available: https://arxiv.org/abs/2404.16130

[16] Z. Wang, Z. Li, Z. Jiang, D. Tu, and W. Shi, "Crafting personalized agents through retrieval-augmented generation on editable memory graphs," in *Proceedings of EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., 2024, pp. 4891–4906. [Online]. Available: https://aclanthology.org/2024.emnlp-main.281/

[17] P. J. Laub, T. Taimre, and P. K. Pollett, "Hawkes processes," 2015. [Online]. Available: https://arxiv.org/abs/1507.02822

[18] L. R. Squire, "Memory systems of the brain: A brief history and current perspective," *Neurobiology of Learning and Memory*, vol. 82, no. 3, pp. 171–177, 2004.

[19] J. M. J. Murre and J. Dros, "Replication and analysis of ebbinghaus' forgetting curve," *PLOS ONE*, vol. 10, no. 7, pp. 1–23, Jul. 2015.

[20] H. Mei and J. Eisner, "The neural hawkes process: a neurally self-modulating multivariate point process," in *NeurIPS*, ser. NIPS'17, 2017, p. 6757–6767.

[21] A. COLLINS and E. LOFTUS, "Spreading activation theory of semantic processing," *PSYCHOLOGICAL REVIEW*, vol. 82, no. 6, pp. 407–428, 1975.

[22] J. T. W. K. Carpenter;, "The wickelgren power law and the ebbinghaus savings function," *Psychological Science*, vol. 18, no. 2, pp. 133–134, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1467-9280.2007.01862.x

[23] P. J. Pavlik and J. Anderson, "Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect," *COGNITIVE SCIENCE*, vol. 29, no. 4, pp. 559–586, JUL-AUG 2005.

[24] A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y. Fang, "Evaluating very long-term conversational memory of LLM agents," in *Proceedings of ACL 2024*, 2024, pp. 13 851–13 870. [Online]. Available: https://aclanthology.org/2024.acl-long.747/

[25] A. Yang, A. Li, B. Yang *et al.*, "Qwen3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

[26] N. Reimers and I. Gurevych, "all-minilm-l6-v2 sentence transformer," https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, 2021, accessed: 2025-4-25.