Think-in-Memory: Metacognition-Augmented LLM with Long-Term Memory

Anonymous EMNLP submission

Abstract

Memory-augmented Large Language Models (LLMs) can recall and reason on recalled past contexts (named recall-reason step). However, multiple recall-reason steps may produce biased thoughts, *i.e.*, inconsistent reasoning paths over the same recalled results. Motivated by that humans only memorize the metacognition thoughts in mind rather than event details, we propose a novel memory-augmented framework called Think-in-Memory (TiM) to flexibly utilize the historical context. Concretely, we formulate a self-organizing memory mechanism equipped with a metacognition space and stationary operation actions, leveraging role-playing LLM agents to achieve thought generator, retriever, and organizer. Supported by such multi-agent self-organization, TiM can imitate human-level metacognition to memorize and update history context as metacognition thoughts without suffering from reasoning inconsistency. TiM can process ultra-long history context in a plug-and-play paradigm to benefit downstream interactions. To conduct evaluations under more complex tasks, clinical diagnosis is adopted as the evaluation task: (1) we formulate a role-play simulator to simulate long-term interactions between the doctor and patient. (2) we collect a multi-turn medical consultations dataset from the real-world hospitals. Besides, two daily conversation datasets are also involved. Experiments demonstrate that our method achieves remarkable improvements on memory-augmented long-term dialogues about both daily and medical topics.

1 Introduction

Impressive advancements in Large Language Models (LLMs) have revolutionized the interaction between human and intelligence systems, as demonstrated by ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023). These advancements have particularly showcased superior performance from finance (Yang et al., 2023) and healthcare (Zhang et al., 2023c) to business and customer service (Eloundou et al., 2023). Nevertheless, it is well-known that existing LLMs suffer from the inability to process long-form inputs (Liu et al., 2022), preventing them from generalizing to real-world scenarios beyond fix-sized inputs (Wang et al., 2024).

Contextual information is particularly critical in LLM-based interactions, *e.g.*, medical AI assistants (Zhang et al., 2023c) may struggle to provide accurate clinical diagnosis due to forgetting crucial medical information of the long-term history. Various studies are conducted to improve the capabilities of LLMs to handle long-term inputs, which can be roughly divided into two types:

△ **Internal Memory** aims to reduce the computational costs of self-attention for expanding the sequence length (Fournier et al., 2023). To accommodate longer input texts, special positional encoding should be exploited to learn relative positions. For example, (Phang et al., 2022) explored a block-local Transformer with global encoder tokens, combined with additional long input pre-training.

△ **External Memory** generally utilizes a physical space as a memory cache to store historical information. Then relevant history can be read from the memory cache to augment LLMs without forgetting. In particular, both token and raw text can be maintained as history in the memory. For instance, (Borgeaud et al., 2022) demonstrated a significant performance improvement by augmenting LLMs with an external memory cache containing trillions of tokens assisted by BERT embeddings (Kenton and Toutanova, 2019). Token-based memory mechanism requires to adjust the LLM's architecture for adaption with additional costs.

By accessing an external memory cache, the augmented LLMs have achieved new state-of-the-art records in various language modeling benchmarks (Wang et al., 2024), generally outperforming internal memory. Therefore, this work focuses on designing an external memory mechanism to en-

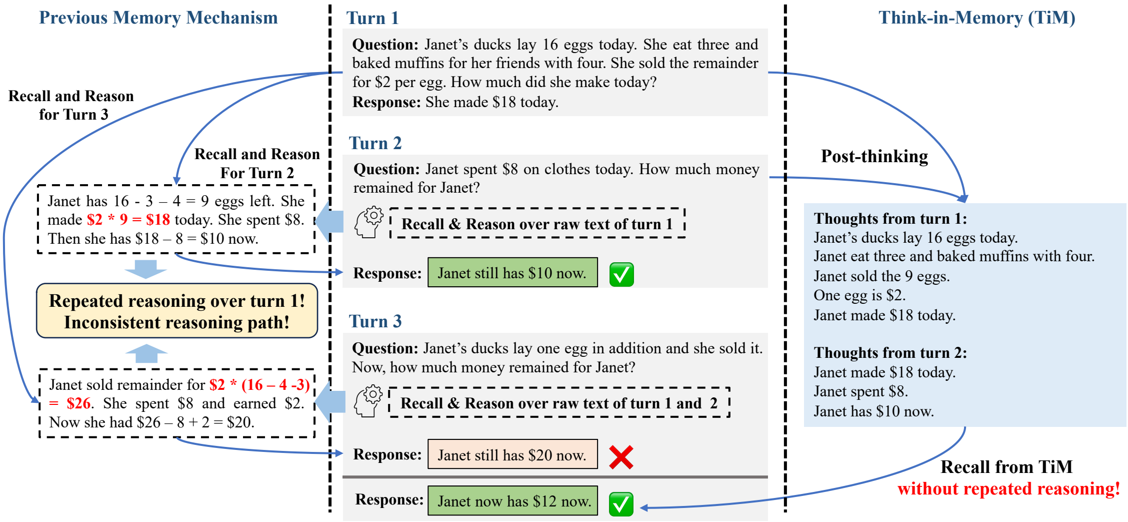

Figure 1: Illustration of Inconsistent Reasoning Path. **(Left)**: Existing memory mechanisms mainly save raw text of previous turns, which require repeated reasoning over the same history. This easily leads to the inconsistent reasoning path (*i.e.*, left red part) with wrong response. **(Right)**: The proposed TiM stores the thoughts of LLMs for previous turns, which can avoid such inconsistency without repeated reasoning (*i.e.*, right red part).

hanced the memorization capacity of LLMs.

Unfortunately, although external memory based methods afford the advantage to processing long inputs, their applications are still hindered by the potential biases emerged during recall-reason step. First one is about inconsistent reasoning paths. As illustrated in Figure 1, taking long-term conversation as an example, the same history context is recalled twice. The LLM may produce two different reasoning results and generate a wrong response. Such phenomena is also demonstrated by recent studies (Adiwardana et al., 2020; Wang et al., 2022). Second one is for overlooking crucial information of recalled raw text due to the noisy context, as indicated by (Barnett et al., 2024). Both issues could easily lead to a performance bottleneck in real-world LLM applications.

To address these concerns, we would like to learn from the typical process of metacognition (Dunlosky and Metcalfe, 2008), where the brain only perceives post-processed thoughts rather than the full details of events. Motivated by this, we propose a **T**hink-**i**n-**M**emory (TiM) framework to utilize long-term history context in the form of thoughts. This framework enables language models to memorize long-term past context into an external memory bank, then model self-organization to mitigate the issue of inconsistent reasoning. To achieve a self-organizing memory mechanism, we design a metacognition space in conjunction with role-playing LLM agents with stationary actions, encompassing 3 critical capabilities: (i) Thoughts Generator: utilize a post-think step to extract metacognition thoughts from the memory bank. (ii) Thoughts Organizer: conduct self-organization operations (insert, merge, and forget) to simplify the metacognition space. (iii) Thoughts Retriever: quickly retrieval relevant thoughts from the metacognition space. To further facilitate the self-organization of the metacognition space, we introduce Locality-Sensitive Hashing to afford efficiency for the metacognition space. Through self-organization evolution to memorize pure and indecomposable thoughts, the memory-augmented LLM can effectively leverage long-term past memory for improving downstream interactions. The key contributions are summarized as follows:

(i) We propose a novel TiM framework to memorize past context as metacognition thoughts, where a self-organizing memory mechanism is designed to mitigate the issue of inconsistent reasoning. TiM can process ultra-long history context in a plug-and-play paradigm to benefit long-term interactions.

(ii) We design a metacognition space in conjunction with role-playing LLM agents with stationary actions, which can formulate self-organization with thoughts generator, organizer, and retriever. Self-organization evolution of the metacognition space can maintain pure and indecomposable thoughts for improving downstream interactions.

(iii) We develop a simulated environment for clinical consultations to evaluate the effectiveness of the proposed method. Also, extensive experiments are conducted on extensive multi-turn dialog datasets. The results indicate that TiM can substantially enhance LLM's performance across various dimensions: (1) It enables diverse topics ranging from open to specific domains; (2) It supports bilingual languages in both Chinese and English; (3) It improves response correctness and coherence.

2 Related Work

Large Language Models. LLMs have attracted significant attention for their superior performance on diverse NLP tasks (Zhang et al., 2023a,b; Guo et al., 2023). Existing LLMs can roughly divided into two types: (1) Closed-source LLMs, *e.g.*, PaLM (Chowdhery et al., 2022), GPT-4 (OpenAI, 2023), and ChatGPT (OpenAI, 2022); (2) Open-source LLMs, *e.g.*, LLaMa (Touvron et al., 2023), ChatGLM (Zeng et al., 2022), and Alpaca (Taori et al., 2023). Recent developments of LLMs cover a broad range of topics, including model architecture (Zeng et al., 2022), training methods (Korbak et al., 2023), fine-tuning strategies (Hu et al., 2021), as well as ethical considerations (Chowdhery et al., 2022). All these methods aim to enhance the understanding capabilities of LLMs for real-world applications. However, these powerful LLM models still have some shortcomings. One notable limitation of LLMs is their lack of a strong long-term memory, which hinders their ability to process lengthy context and retrieve relevant historical information.

Long-term Memory. Numerous efforts are conducted to enhance the memory capabilities of LLMs. For example, one approach is memory-augmented networks (MANNs) (Meng and Huang, 2018) with an external memory cache, which can well handle tasks of long-term period by interacting with memory. Recently, many studies focused on long-term conversations (Xu et al., 2021, 2022; Zhong et al., 2023; Liang et al., 2023). For example, Xu *et al.* (Xu et al., 2021) introduced a new English dataset consisting of multi-session human-human crowdworker chats for long-term conversations. Zhong *et al.* (Zhong et al., 2023) proposed a MemoryBank mechanism inspired by Ebbinghaus' forgetting curve theory. However, these methods still face some great challenges to achieve a reliable and adaptable long-term memory mechanism for Language and Learning Models (LLMs). Concretely, these methods only considered storing the raw dialogue text, requiring repeated reasoning of the LLM agent over the same history. Besides, these models need to calculate pairwise similarity for recalling relevant information, which is time-consuming for long-term interactions.

3 Methodology

In this section, we first formulate the problem of memory-augmented language modeling. Then, we introduce the architecture of our self-organizing mechanism with a metacognition space. Next, we provide the designed workflow of role-playing LLM agents to generate, organize, and retrieve the metacognition thoughts in the metacognition space.

3.1 Metacognition-Augmented LLMs

Problem Formulation. Given the wide exploration of pretrained LLMs, our TiM framework is built on a LLM backbone $\mathbf{f}_\theta$ parameterized by θ. Each round of the interaction consists of two parts, *i.e.*, a user's query $\mathbf{Q}$ and the LLM's response $\mathbf{R}$. Specifically, in the i-th round of interaction, given the inquiry $\mathbf{Q}_i$, the LLM generates a response $\mathbf{R}_i = \{r_0^{(i)}, r_1^{(i)}, \cdots, r_{|R_i|}^{(i)}\}$ based on the interaction history in an autoregressive manner:

$$p_\theta\left(\mathbf{R}_i \mid \mathbf{Q}_i, \mathbf{M_b}\right) = \prod_{k=1}^{|R_i|} p_\theta\left(r_k^{(i)} \mid \mathbf{Q}_i, \mathbf{M_b}\right), \tag{1}$$

where $\mathbf{M_b} = \mathbf{Q}_{0:i-1}, \mathbf{R}_{0:i-1}$ is to cache history contexts for memory augmentations. However, with gradually increasing round number i or with the very lengthy history context, *i.e.* i or $|R_{k\in[0,i-1]}|$, only the partial segment of the history sequence can be normally processed due to the limitation of fix-sized inputs for most existing LLMs. Thus, existing memory-augmented mechanisms mainly focus on recalling relevant history contexts for augmentations instead of full history:

$$p_\theta\left(\mathbf{R}_i \mid \mathbf{Q}_i, \mathbf{M}_b\right) \approx \prod_{k=1}^{|R_i|} p_\theta\left(r_k^{(i)} \mid \mathbf{Q}_i, \mathbf{F}(\mathbf{Q}_i, \mathbf{M}_b)\right), \tag{2}$$

where $\mathbf{F}(\cdot)$ is a metric function to retrieve relevant history contents. For $k \in [0, i-1]$, when some history (Q_k, R_k) is recalled twice or more, LLMs may produce different reasoning paths (*e.g.*, different CoT prompts (Wei et al., 2022)), inevitably resulting in unexpected responses.

Metacognition Augmentation. To address the above issue, text-embedding pairs of previous inputs are stored in Memory Bank $\mathbf{M}_b$, which are se-

Table 1: Comparisons of memory mechanisms. KG denotes the knowledge graph and Q-R is question-response pair.

Method	Content	LLM-agnostic	Insert	Forget	Merge
SCM (Liang et al., 2023)	Q-R	✓	✓	✗	✗
RelationLM (Liu et al., 2022)	KG	✗	✓	✗	✗
LongMem (Wang et al., 2024)	Token	✗	✓	✗	✗
MemoryBank (Zhong et al., 2023)	Q-R	✓	✓	✓	✗
Ours (TiM)	Q-R, Thoughts	✓	✓	✓	✓

quentially transformed as thoughts in the metacognition space $\mathbf{M}_s$. Embeddings of the raw text are retained for retrieval. Given the current input, top relevant previous thoughts are recalled to augment the language modeling for response generation. The metacognition space can be viewed as a self-organizing system (*i.e.*, like a human brain) to organize historical thoughts. The overall language modeling can be denoted as:

$$p_\theta\left(\mathbf{R}_i|\mathbf{Q}_i, \mathbf{M}_b\right) \approx \prod_{k=1}^{|R_i|} p_\theta\left(r_k^{(i)}|\mathbf{Q}_i, \underbrace{\mathbf{F}(\mathbf{Q}_i, \mathbf{M}_s)}_{\text{TiM}}\right), \tag{3}$$

where $\mathbf{M}_s = \{\mathbf{T}_s, \mathbf{A}_s\}$ and $\mathbf{F}(\cdot)$ is to retrieve relevant thoughts of $\mathbf{Q}_i$. $\mathbf{T}_s$ is the thought set and $\mathbf{A}_s$ is the self-organizing action set. Here we provide explicit definitions for *Metacognition* and *Thought*.

Definition 1 ***Metacognition*** *is originally defined to as the knowledge about and regulation of one's cognitive activities in learning processes (Flavell, 1979). One metacognition space* $\mathbf{M}_s$ *consists of metacognition thoughts* $\mathbf{T}_s$ *and a set of self-organizing actions* $\mathbf{A}_s$*.*

Definition 2 ***Thought*** *is defined as the minimum unit in the metacognition space. One thought can be basiclly represented as a relation triple* (E_h, R, E_t)*, where* E_h *is head entity connected with tail entity* E_t *via the relation* R*.*

Framework Architecture. Here, we implement the proposed framework based on a multi-agent system. As illustrated in Figure 3, our framework comprises the following components, working together to provide more accurate and coherent responses for long-term interaction: (1) Agent Core, which is a pre-trained LLM backbone $\mathbf{f}_\theta$ to facilitate dynamic interactions, such as ChatGPT (OpenAI, 2022) and ChatGLM (Zeng et al., 2022). (2) Cache Module, which contains continually growing $\mathbf{M}_b$ and $\mathbf{M}_s$ as memory cache. (3) Self-organization Module, which imitates the human brain to organize the thoughts in the metacognition space according to certain rules.

3.2 Memory Cache

TiM's memory cache aims to store the history contexts of the long-term interactions. The memory cache consists of a Memory Bank $\mathbf{M}_b$ and a Metacognition Space $\mathbf{M}_s$.

Memory Bank. $\mathbf{M}_b$ is utilized to preserve the raw texts from the interactions. Each data instance of $\mathbf{M}_b$ is in the format of the text-embedding pair $(\mathbf{U}_k^{\text{txt}}, \mathbf{U}_k^{\text{emb}})$, where $\mathbf{U}_k^{\text{txt}}$ denotes the raw text of $(\mathbf{Q}_k, \mathbf{R}_k)$ and $\mathbf{U}_k^{\text{emb}}$ denotes the sentence-level embedding of $\mathbf{U}_k^{\text{txt}}$ from the LLM backbone $\mathbf{f}_\theta$. Here, the memory bank is a long-term cache to store fixed text-embedding pairs without any modifies.

Metacognition Space. $\mathbf{M}_s$ is designed to save high-level thoughts (Definition 2). Similarly, each data instance of $\mathbf{M}_s$ is in the format of the thought-embedding pair $(\mathbf{T}_k^{\text{tho}}, \mathbf{T}_k^{\text{emb}})$, where $\mathbf{T}_k^{\text{tho}}$ denotes the thoughts from the $(\mathbf{Q}_k, \mathbf{R}_k)$. Different with the memory bank, the metacognition space is a short-term cache, where thought-embedding pairs would be evolved via self-organizing behaviors.

Notice that human can conduct association among the relevant memories. Following this rule, the semantically similar pairs should be cached in the same group for both memory bank and metacognition space. Their cache structures are the same due to shared embeddings, increasing the cache efficiency. To achieve this, we adopt a hash-based structure for TiM's memory cache, where similar pairs are assigned with the same hash index. Given a newly coming memory pair, we propose to quickly search its nearest thoughts in a high-dimensional embedding space, which can be solved by the locality-sensitive hashing (LSH) method. The hashing scheme of LSH is to assign each d-dimension embedding vector $x \in \mathbf{R}^d$ to a hash index $\mathbf{H}(x)$, where nearby vectors get the same hash index with higher probability. We achieve this by exploiting a random projection as follows:

$$\mathbf{H}(x) = \arg\max\left([xR; -xR]\right), \tag{4}$$

where R is a random matrix of size $(d, b/2)$ and b is the number of groups in the memory. $[u; v]$ denotes the concatenation of two vectors. This implementation is a well known LSH scheme (Andoni et al., 2015). In particular, the embedding $\mathbf{U}^{\text{emb}}$ of raw texts is used to construct the hash index for these two cache types, which cam enhance the structural consistency between them,

3.3 Think-in-Memory: Self-organized Agents

In this section, we design role-playing LLM agents to implement a self-organizing metacognition space, which achieve the actions of thought generator, retriever, and organizer.

Thought Generator. The main challenge is to generate high-quality sentences matching relation triples. Here we provide two kinds of solutions to generate thoughts: (1) pre-trained model for open information extraction, such as OpenIE (Angeli et al., 2015); (2) In-context learning with few-shot prompts based on LLM. In this work, we utilize a LLM agent to serve as a thought generator. Given a query $\mathbf{Q}$ and a corresponding response $\mathbf{R}$, we ask a frozen LLM to work as a thought generator, which derives the thoughts $\mathbf{T}$ for the $(\mathbf{Q}, \mathbf{R})$ following the specialized prompt:

$$\mathbf{T}_{\text{gen}} = \textbf{Agent}\left((\mathbf{Q}, \mathbf{R}), \text{Role}_{\text{gen}}, \text{Prompt}_{\text{gen}}\right). \tag{5}$$

LLM backbone $\mathbf{f}_\theta$ first provides a response $\mathbf{R}$ for $\mathbf{Q}$ according to Eq. 3, then generates the thoughts upon the Q-R pair. Thus, this thought generator stage is termed as Post-Think. Finally, $(\mathbf{Q}, \mathbf{R})$ and the generated $\mathbf{T}_{\text{gen}}$ are stored into $\mathbf{M}_b$ and $\mathbf{M}_s$ according to Eq. 4, respectively.

Thought Organizer. With the above-discussed generator, the long-term memory capability of LLMs can be well enhanced via self-organization. Motivated by the human brain, there needs some organization actions for dynamic evolution of the metacognition space, which can make the memory mechanism more natural and applicable. Three basic self-organizing actions are formulated, *i.e.*, $\text{Action}_{\text{org}} = \{\text{Insert}, \text{Forget}, \text{Merge}\}$. Insert action is performed by Eq. 4. Assuming that new thought is inserted into the group $\mathbf{G}$, thought organizer performs the self-organizing actions to organize the thoughts of $\mathbf{G}$:

$$\hat{\mathbf{G}} = \textbf{Agent}\left(\mathbf{G}, \text{Role}_{\text{org}}, \text{Action}_{\text{org}}, \text{Prompt}_{\text{org}}\right), \tag{6}$$

where $\hat{\mathbf{G}}$ denotes the newly evolved thought group. Intuitively, Forget action is to remove unnecessary thoughts such as contradictory thoughts. Merge action is to combine similar thoughts together, such as thoughts with the same head entity.

Thought Retriever. Built on the cache module, we implement a thought retriever $\mathbf{F}(\cdot)$ based on Retrieval-Augmented Generation (RAG), which operates a two-stage retrieval task to search the most relevant thoughts, *i.e.*, LSH-based retrieval followed by similarity-based retrieval. **Stage-1:** LSH-based Retrieval. For a new query $\mathbf{Q}$, we first obtain its embedding vector x based on LLM backbone $\mathbf{f}_\theta$. Then LSH function (*i.e.*, Eq. 4) can provide the hash index of Q, which indicates the its nearest thought group in $\mathbf{M}_s$ according to the property of LSH. **Stage-2:** Similarity-based Retrieval. Within the nearest group, we calculate the pairwise similarity between the query and each piece of thought in the group. Then top-k thoughts are recalled as the relevant history for accurately answering the query. It should be noticed that pairwise similarity is only calculated within a group rather than the whole memory space, which can achieve more efficient retrieval than previous memory mechanisms. Besides, the two-stage retrieval is only performed within the metacognition space. If without relevant thoughts, similarity-based retrieval will be executed in the memory bank.

Once the top-k relevant thoughts $\mathbf{T}_{\text{ret}} = \{\mathbf{T}_0, \mathbf{T}_1, \cdots, \mathbf{T}_k\}$ are returned, we ask a frozen LLM to work as a retrieval-augmented generator, which integrates the thoughts $\mathbf{T}_{\text{ret}}$ with the original $\mathbf{Q}$ following the specialized prompt:

$$\hat{\mathbf{T}}_{\text{ret}} = \textbf{Agent}\left(\mathbf{Q}, \mathbf{T}_{\text{ret}}, \text{Role}_{\text{ret}}, \text{Prompt}_{\text{ret}}\right), \tag{7}$$

where $\hat{\mathbf{T}}_{\text{ret}}$ denotes the final retrieval result. Thus, thought retriever can recall relevant history contexts and integrate them according to the requirements of the long-term interactions.

3.4 Long-term Environment Simulation

In this section, we implement a role-playing framework to simulate the environment for long-term medical consultations, which targets to evaluate the effectiveness of TiM for the medical scenario.

Patient Simulator. One LLM with in-context instruction prompt is utilized to imitate the behavior

Table 2: Comparison Results on Three Datasets. Top-5 thoughts are recalled from the memory cache.

Dataset	LLM	Topic	Memory	Retrieval Accuracy	Response Correctness	Contextual Coherence
GVD	ChatGLM	Open-EN	Silicon	0.809	0.438	0.680
			Ours	**0.820**	**0.450**	**0.735**
		Open-CN	Silicon	0.840	0.418	0.428
			Ours	**0.850**	**0.605**	**0.665**
Kdconv	ChatGLM	Film-CN	✗	-	0.657	0.923
			Ours	**0.920**	**0.827**	**0.943**
		Music-CN	✗	-	0.666	0.910
			Ours	**0.970**	**0.826**	**0.926**
		Travel-CN	✗	-	0.735	0.906
			Ours	**0.940**	**0.766**	**0.912**
	Baichuan2	Film-CN	✗	-	0.360	0.413
			Ours	**0.913**	**0.743**	**0.870**
		Music-CN	✗	-	0.253	0.283
			Ours	**0.900**	**0.710**	**0.780**
		Travel-CN	✗	-	0.207	0.280
			Ours	**0.833**	**0.757**	**0.807**
RMD	ChatGLM	Medical-CN	✗	-	0.806	0.893
			Ours	**0.900**	**0.843**	**0.943**
	Baichuan2	Medical-CN	✗	-	0.506	0.538
			Ours	**0.873**	**0.538**	**0.663**

of patients for clinical consultations. The patient LLM could provide accurate medical information such as descriptions. The data source of patients' medical information is from the realistic medical records, as shown in Appendix. Similar to real-world patients, the patient LLM is to provide medical information in a lazy mode. These requirements are achieved via a specially designed prompt:

$$\mathbf{R}_{\text{pat}} = \mathbf{Agent}\left(\mathbf{Q}_{\text{doc}}, \text{Role}_{\text{pat}}, \text{Prompt}_{\text{pat}}\right), \tag{8}$$

where $\mathbf{Q}_{\text{doc}}$ denotes the query from the doctor and $\mathbf{R}_{\text{pat}}$ is the response of the patient.

Doctor Simulator. Another LLM is utilized to act as a doctor for clinical consultations. At the beginning, the doctor LLM should ask questions about key medical information based on the patient's basic situation. Then, the doctor LLM needs to provide accurate diagnosis and treatment results according to the historical medical information of the patient. The doctor LLM is also achieved via a specially designed prompt:

$$\mathbf{Q}_{\text{doc}}/\text{END} = \mathbf{Agent}\left(\mathbf{R}_{\text{pat}}, \mathbf{T}_{\text{med}}, \text{Role}_{\text{doc}}, \text{Prompt}_{\text{doc}}\right), \tag{9}$$

where $\mathbf{T}_{\text{med}}$ denotes the retrieved relevant history information from the memory. END denotes the conversation end with final diagnosis results.

Retrieval-Augmented Module (RAM). RAM performs a connection between doctor and patient. The data base is a pool of patients' medical records. (1) Given a query from the doctor, the patient simulator firstly understand the intent of the doctor. Based on the intent, RAM aims to retrieve relevant medical information as the candidate responses for the patient. Then, the retrieved results are integrated into the original query as an augmented query $\mathbf{Q}_{\text{doc}}$ in Eq. 8. (2) For each patient, when the doctor gives the final diagnosis and treatment results, RAM can achieve automatic evaluation by retrieving the ground-truth from the pool.

4 Experiment

4.1 Multi-turn Dialogue

Datasets. Three datasets are used to demonstrate the effectiveness of TiM. **KdConv**: KdConv is a Chinese multi-domain knowledge-driven conversation benchmark (Zhou et al., 2020) grounding the topics to knowledge graphs, which involves 4.5K conversations and 86K utterances from three domains (film, music, and travel). The average turn number is 19. **Generated Virtual Dataset (GVD)**: GVD is a long-term conversation dataset (Zhong et al., 2023) involving 15 virtual users (ChatGPT) over 10 days. Conversations are synthesized using pre-defined topics, including both English and Chinese languages. For the test set, (Zhong et al., 2023) manually constructed 194 query questions (97 in English and 97 in Chinese) to evaluate whether the LLM could accurately recall the memory and pro-

Table 3: Performance for the Simulated Medical Consultation.

Accuracy	Memory	HuatuoGPT II-13B	Baichuan2 chat-13B	Chatglm2 6B	Chatglm3 6B	GPT3.5 Turbo	GPT4 -
Diagnosis	No	18.18	20.45	15.91	2.27	18.18	0.0
	Raw	25.00	20.45	22.72	20.45	22.73	50.00
	TiM	27.27	22.27	22.72	20.45	22.73	50.00
Treatment	No	2.27	2.27	0	2.27	4.55	0.0
	Raw	4.54	2.27	0	9.10	6.82	6.82
	TiM	6.82	4.54	2.27	9.10	6.82	6.82

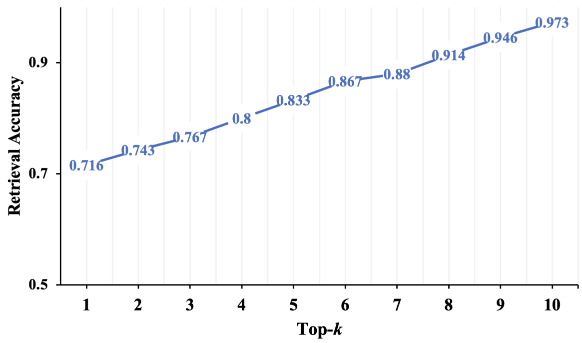

Figure 2: Top-k Retrieval accuracy.

Table 4: Comparisons of Retrieval Time. Baseline directly calculates pairwise similarity.

Method	Retrieval Time (ms)
Baseline	0.6287
Ours (TiM)	0.5305

duce the appropriate answers. **Real-world Medical Dataset (RMD)**: To evaluate the effectiveness of the proposed memory mechanism in the real-world scenarios, we manually collect and construct a dataset containing 1,800 conversations for medical healthcare consumer. For the test set, 80 conversations are used to evaluate whether the LLM could provide the accurate diagnosis.

Evaluation Settings. One baseline is to answer questions without using any memory mechanism. Another baseline is SiliconFriend (Zhong et al., 2023), a classical memory mechanism, which can store the raw text into the memory and support reading operation. To be fair, during evaluation, the prediction results of all LLMs are firstly shuffled, ensuring the human evaluator does not know which LLM the results come from. Then the final evaluation results are obtained by the human evaluation. Following (Zhong et al., 2023), three metrics are adopted to evaluate the performance of the proposed method. **Retrieval Accuracy**: Retrieval accuracy evaluates whether the relevant memory is successfully recalled (labels: {0: no; 1: yes}). **Response Correctness**: Response correctness evaluates if correctly answering the probing question (labels: {0: wrong; 0.5: partial; 1 : correct}). **Contextual Coherence**: Contextual coherence evaluates whether the response is naturally and coherently generated, *e.g.*, connecting the dialogue context and retrieved memory (labels: {0: not coherent; 0.5: partially coherent; 1: coherent}).

Comparison Results. The main results on three datasets are summarized in Table 2. The following insights are observed: (1) Our method significantly outperforms the considered baselines on these datasets. Compared with SiliconFriend (Zhong et al., 2023), our method exhibits superior performance for all metric, especially for the contextual coherence, indicating the effectiveness of TiM mechanism. (2) TiM delivers better results on both languages and various topics. The performance improvement on Chinese is larger than English, which may be attributed to the abilities of the LLMs. We evaluate 2 different LLMs with TiM over different topics (film, music, and travel) and our method can obtain best results across all topics. Our method can achieve high retrieval accuracy to recall the relevant thoughts. (3) TiM can help to improve the contextual coherence of the response. RMD dataset contains the realistic conversations between the doctors and patients. Our method can improve the overall response performance for the real-world medical conversations. In detail, using TiM, both ChatGLM and Baichuan2 can improve their capability for long-term conversations, *i.e.*, significant improvements on the response correctness and the contextual coherence.

4.2 Medical Consultation

Medical consultations generally require understanding the patient's entire health conditions to make accurate diagnoses and treatment decisions, while conventional methods easily suffer from the

Table 5: Examples of Case Study. The red color denotes the mistake. 1-th column is the history context before i-th dialog. 3th column shows varying i-th dialog generations of different memory methods. P is patient and D is doctor.

History Context	Memory	i-th Dialog
P: I last came in due to a urinary tract infection. **D:** Have you felt any improvement since your last visit? **P:** I'm not sure. I've had a urine analysis done recently. **D:** *The results of the urine analysis?* **P:** The white blood cell count in the urinary sediment is $3623.20/\mu l$. **D: That's a relatively high value.** Are you experiencing frequent urination? **P:** Yes, it's been quite noticeable recently.	No	**Forget and Repeat Question** **D:** *Have you had a urine analysis done recently?* **P:** Yes, you just asked me.
	Raw Text	**Inconsistent Reasoning Result** **D:** Okay, combined with **reasonable** white blood cell count and frequent urination, everything is fine.
(patient, have, urine analysis) (white blood cell count, is, $3623.20/\mu l$) ($3623.20/\mu l$, is, a relatively high value) (patient, have, frequent urination)	TiM	**D:** Okay, combined with relatively higher white blood cell count and frequent urination, you may catch an **acute urinary tract infection.**

forgetting of the history context. With the proposed metacognition-augmented memory, TiM allows the LLM to remember and retrieve specific past details about a patient's medical history, test results, treatments, allergies, and other crucial information that could influence the current consultation.

Evaluation Settings. Based on the simulated framework in Section A.2, three memory schemes are achieved as the baselines: (1) Without any memory mechanisms (No); (2) Raw Dialog Context as the memory (RaW); (3) The proposed method (TiM). Both diagnosis and treatment accuracy results are adopted as the evaluation protocols.

Simulation Results. Table 3 reports the comparison results on the simulated medical environment for the interactions between doctor and patient. As shown in Table 3, our method can perform better than baseline memory methods for all LLMs. In detail, both diagnosis and treatment accuracy would be increased by TiM (*e.g.*, HuatuoGPT, Baichuan2, and Chatglm2). For GPT3.5 and GPT4 with strong capabilities, the probability of reasoning inconsistency is relatively low, thus there is no obvious performance gap between TiM and raw text.

4.3 Ablation Studies

Retrieval Time. We report the comparison results of retrieval time. The baseline is to calculate pairwise similarity between the question and the whole memory, which is utilized as the default retrieval way for most previous mechanisms. For both baseline and our method, the memory length is as 140 and the memory context is fixed. Table 2 shows the time cost for making a single retrieval. It is observed that our method can reduce about 0.1 ms retrieval time compared with baseline method.

Top-k Retrieval. The retrieval accuracy with different k values are summarized in Figure 2. Our method achieves gradually improved retrieval accuracy along with increasing k. Meanwhile, top-1 retrieval accuracy is higher than 0.7 and top-10 can achieve 0.973 retrieval accuracy. Besides, as shown in Table 2, the overall model performance is also improved with increasing value of k. For example, when $k = 5$, our method can significantly improve the performance of existing LLMs for long-term conversations. As shown in Table 3, our method can still outperform the comparison baselines (Raw and No) for medical scenarios.

Case Study Table 5 exhibits a patient case, where doctor LLMs are augmented by different memory mechanisms, respectively. As indicated by the first row, the doctor LLM may forget the previous information with repeatedly asking the similar query about *"urine analysis"*. When storing the raw text as the memory, the doctor LLM would conduct multiple reason steps over the same medical information of $3623.20/\mu l$, but results in the inconsistent reasoning paths, *i.e.*, *higher VS. reasonable* white blood cell count. TiM can store the minimum unit thoughts in the memory, which can avoid excessive reasoning and generation. Therefore, our method can finish the task of medical consultation and provide correct diagnosis results, *i.e.*, *acute urinary tract infection*.

5 Conclusion

In this work, we propose a TiM framework with a novel self-organizing metacognition space. TiM leverages role-playing LLM agents with predefined stationary actions for thought generator, retriever, and organizer, which can imitate human-level metacognition to manage history context. Additionally, TiM can process ultra-long history context in a plug-and-play paradigm to benefit downstream interactions. Experiments demonstrate that our method achieves remarkable improvements on memory-augmented long-term dialogues.

6 Limitations

TiM incorporates external memory components to enhance LLMs' capacity to handle long-term dependencies in a dialog system, providing a mechanism to store and retrieve information effectively across extended contexts. However, such memory-augmented LLMs also have certain limitations about interpretability. Understanding why and how the LLMs use the memory is important for debugging, improving, and trusting the dialog system.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. 2015. Practical and optimal lsh for angular distance. *Advances in Neural Information Processing Systems*, 28.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. *arXiv preprint arXiv:2401.05856*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pages 2206–2240. PMLR.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

John Dunlosky and Janet Metcalfe. 2008. *Metacognition*. Sage Publications.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*.

John H Flavell. 1979. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10):906.

Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. 2023. A practical survey on faster and lighter transformers. *ACM Computing Surveys*, 55(14s):1–40.

Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. 2023. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10877.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, volume 1, page 2.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR.

Xinnian Liang, Bing Wang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2023. Unleashing infinite-length input capacity for large-scale language models with self-controlled memory system. *arXiv preprint arXiv:2304.13343*.

Qi Liu, Dani Yogatama, and Phil Blunsom. 2022. Relational memory-augmented language models. *Transactions of the Association for Computational Linguistics*, 10:555–572.

Lian Meng and Minlie Huang. 2018. Dialogue intent classification with long short-term memory networks. In *Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6*, pages 42–50. Springer.

OpenAI. 2022. Chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Jason Phang, Yao Zhao, and Peter J Liu. 2022. Investigating efficiently extending transformers for long input summarization. *arXiv preprint arXiv:2208.04347*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2024. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022. Long time no see! open-domain conversation with long-term persona memory. *arXiv preprint arXiv:2203.05797*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. *arXiv preprint arXiv:2301.07069*.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Ali Babar, and Xiao-Yang Liu. 2023b. Enhancing financial sentiment analysis via retrieval augmented large language models. *arXiv preprint arXiv:2310.04027*.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023c. Huatuogpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*.

Hao Zhou, Chujie Zheng, Kaili Huang, Minlie Huang, and Xiaoyan Zhu. 2020. KdConv: A Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation. In *ACL*.

A Appendix

A.1 Parameter-efficient Tuning

We adopt a computation-efficient fine-tuning approach called Low-Rank Adaptation (LoRA) (Hu et al., 2021) for the scenarios with limited computational resources. LoRA (Hu et al., 2021) optimizes pairs of rank-decomposition matrices while keeping the original weights frozen, which can effectively reduce the number of trainable parameters. Specifically, considering a linear layer defined as $y = Wx$, LoRA fine-tunes it according to $y = Wx + BAx$, where $W \in \mathbf{R}^{d\times k}$, $B \in \mathbf{R}^{d\times r}$, $A \in \mathbf{R}^{r\times k}$, and $r \ll \min(d; k)$. Essentially, this fine-tuning stage can adapt LLMs to multi-turn conversations, providing appropriately and effectively response to users. For all experiments, we set LoRA rank r as 16 and train the LLM models for 10 epochs.

A.2 Evaluation Simulation

In this section, we implement a multi-agent framework to simulate the long-term medical consultations, which targets to evaluate the effectiveness of TiM for the medical scenario.

Patient Simulator. One LLM with in-context instruction prompt is utilized to imitate the behavior of patients for clinical consultations. The patient LLM could provide accurate medical information such as descriptions. To be in line with real-world patients, the patient LLM is to provide medical information in a lazy mode. These requirements are achieved via a specially designed prompt:

$$\mathbf{R}_{\mathrm{pat}} = \mathbf{Agent}\left(\mathbf{Q}_{\mathrm{doc}}, \mathrm{Role}_{\mathrm{pat}}, \mathrm{Prompt}_{\mathrm{pat}}\right), \tag{10}$$

where $\mathbf{Q}_{\mathrm{doc}}$ denotes the query from the doctor and $\mathbf{R}_{\mathrm{pat}}$ is the response of the patient.

Doctor Simulator. Another LLM is utilized to act as a doctor for clinical consultations. At the beginning, the doctor LLM should ask questions about key medical information based on the patient's basic situation. Then, the doctor LLM needs to provide accurate diagnosis and treatment results according to the historical medical information of the patient. The doctor LLM is also achieved via a specially designed prompt:

$$\mathbf{Q}_{\mathrm{doc}}/\mathrm{END} = \mathbf{Agent}\left(\mathbf{R}_{\mathrm{pat}}, \mathbf{T}_{\mathrm{med}}, \mathrm{Role}_{\mathrm{doc}}, \mathrm{Prompt}_{\mathrm{doc}}\right), \tag{11}$$

where $\mathbf{T}_{\mathrm{med}}$ denotes the retrieved relevant history information of the patients. END denotes the end of the conversations with final diagnosis results. Three memory schemes are achieved based on such simulation: (1) Without any memory mechanisms; (2) Raw Dialog as the memory; (3) Our TiM.

A.3 Insightful Discussion

Here we make a summary for previous memory mechanisms and our method in Table 1, including memory content, LLM-agnostic, and organization operations. There are several important observations from Table 1: (1) Previous memory mechanisms only save raw conversation text (Q-R pairs) as the memory, which requires repeated reasoning over the history. Our method maintains thoughts in the memory cache and can directly recall them without repeated reasoning. (2) Previous memory mechanisms only support simple read and write (insert) operations, while our method provides more manipulate way for the memory. (3) Some previous memory mechanisms store the tokens in the memory, which requires adjusting LLM architecture (LLM-aware) for applications. Our method is deigned as a LLM-agnostic module, which can be easily combined with other LLMs.

A.4 More Illustrations

Figure 3 illustrates the workflow of the proposed TiM, where post-think denotes the operation conducted by thought generator.

Figure 4 shows an real-world application, which equips LLM models with the proposed TiM.

Table 6 exhibits an example used for long-term environment simulation, which involves complete medical information of a virtual patient.

Figure 5, 7, 6 are three examples for the prompt templates used by role-play agents, respectively.

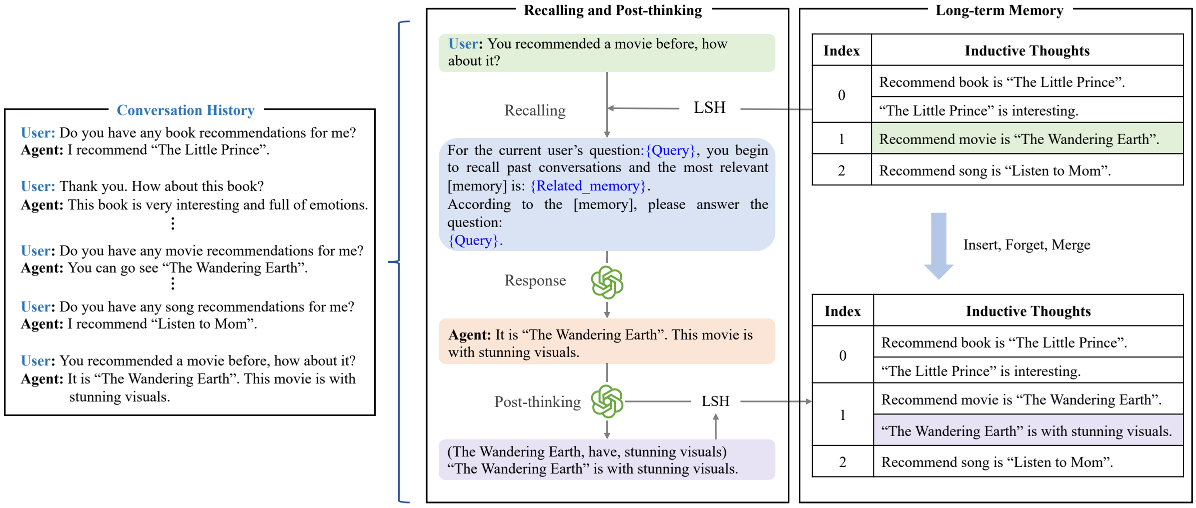

Figure 3: The overview of TiM framework. LLMs firstly recall history and give response for the question. Then new thoughts can be generated via the post-thinking step. These thoughts are saved as the memory to avoid repeated reasoning on the history.

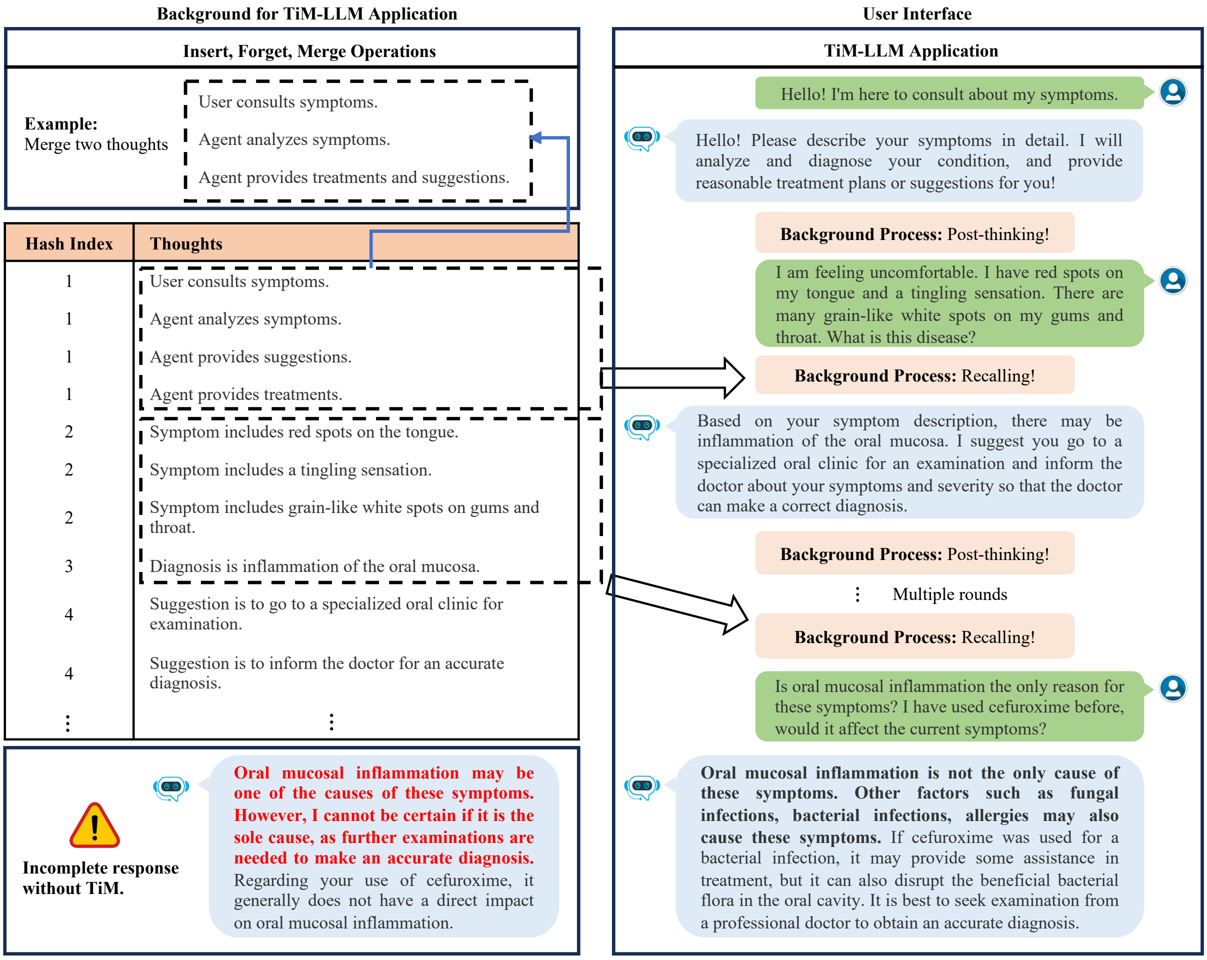

Figure 4: The application of TiM. The left is the background of TiM-LLM application and the right is user interface.

Table 6: An example of the medical record of a patient.

1-st Category	**2-nd Category**	**Report Result**	**Constraints**
Initial Query		After my mom peed, the toilet felt like there was blood in it.	
Patient's Intent		Prefer conservative treatment	
Expected Rounds		7 Rounds	
Age		Female	
Gender		84 Years Old	
Diagnosis	Core Diagnosis	Bladder Cancer	
Key Elements and Correct Order of Questions for Confirming the Diagnosis			
Symptom	Gross Hematuria	Consistent blood in urine, color is pink.	
Symptom	Urgency	None	
Symptom	Fever	None	
Symptom	Difficulty Urinating	None	
Test	Urinalysis	Urine color is brownish red, specific gravity 1.016, pH 6.0.	7 Days Ago
Test	Urinary NMP22	Positive	7 Days Ago
Examination	Urinary Ultrasound	Normal size and shape, clear capsule contour.	7 Days Ago
Examination	Bladder Enhanced MRI	Multiple occupying lesions on the left side.	7 Days Ago
Examination	CT Urography	Posterior wall of the bladder. Left kidney cyst.	7 Days Ago
Correct Order		Urinary Ultrasound > Enhanced MRI = CT Urography	
Treatment		TURBT	
Key Elements of Treatment Plan			
Surgical History	None		
Medication History	None		
General Condition		Sleep is fairly good, no unusual diet, no significant weight change.	
Medical History	Hypertension	Blood pressure controlled around 140/90	
Medical History	Diabetes Mellitus	Negative	
Medical History	Coronary Heart Disease	Negative	
Marital & Childbearing	Married with Child		
Menstrual History	Menopause		
Preoperative Test	Complete Blood Count	White blood cell count 4.69x10$\hat{9}$/L.	1 Day Ago
Preoperative Test	Liver Function	Total bilirubin 12.9 μmol/L, direct bilirubin 3.2 μmol/L.	1 Day Ago
Preoperative Test	Renal Function	Urea 5.60mmol/L, creatinine 48.0 μmol/L.	1 Day Ago
Preoperative Test	Fasting Blood Glucose	Fasting blood glucose 6.56mmol/L.	1 Day Ago
Preoperative Test	Coagulation Function	Prothrombin time 11.80 seconds.	1 Day Ago
Preoperative Test	B-type Natriuretic Peptide	B-type natriuretic peptide 70.0pg/mL;	1 Day Ago
Preoperative Test	Cardiac Infarction Markers	Troponin I 0.01ng/ml, Myoglobin 15.80ng/ml.	1 day ago
Preoperative Test	Hepatitis B	Hepatitis B surface antigen 0.45COI.	1 day ago
Preoperative Test	Hepatitis C	Hepatitis C antibody (C) 0.04S/CO,	1 day ago
Preoperative Test	HIV	HIV Ag/Ab 0.05COI	1 day ago
Preoperative Test	Syphilis	Confirmatory test for syphilis negative	1 day ago
Preoperative Test	Chest CT Scan	Scattered tiny nodules in both lungs.	1 day ago
Preoperative Test	Echocardiography	No obvious abnormalities.	1 day ago
Preoperative Test	Electrocardiogram	Sinus rhythm, low flat T waves.	1 day ago

Prompt for Forgetting Thoughts

Given the following thoughts, please remove the counterfactual thoughts or contradictory thoughts:

Example 1.
Input:
The capital of China is Beijing.
The capital of China is Shanghai.
The capital of the United States is Washington.
The capital of the United States is New York.
Output:
The capital of China is Beijing.
The capital of the United States is Washington.

Example 2.
Input:
Michael likes to play football.
Michael does not like to play football.
James likes to swim.
Mary likes to read books.
Output:
James likes to swim.
Mary likes to read books.

Input:
[A group of thoughts]
Output:

Figure 5: An example of prompts for forgetting merging thoughts.

Prompt for Merging Thoughts

Given the following thoughts, please merge the similar thoughts with the same entity:

Example 1.
Input:
John works as an actor.
John works as a director.
John works as a writer.
Mike works as a teacher.
Output:
John works as an actor, a director, and a writer.
Mike works as a teacher.

Example 2.
Input:
Michael likes to play football.
Michael likes to play basketball.
James likes to swim.
Mary likes to read books.
Output:
Michael likes to play football and basketball.
James likes to swim.
Mary likes to read books.

Input:
[A group of thoughts]
Output:

Figure 6: An example of prompts for merging thoughts.

Prompt for Generating Thoughts

Given the following question and response pairs, please extract the relation (subject, relation, object) with corresponding text:

Example 1.
Input:
Question: Do you have any company recommendations for me?
Response: I recommend Google.
Output:
(Company, Recommended, Google).
Recommended company is Google.

Example 2.
Input:
Question: Which City is the capital of China?
Response: Beijing.
Output:
(China, Capital, Beijing).
The capital of China is Beijing.

Input:
Question: Do you have any book recommendations for me?
Response: I recommend "The Little Prince".
Output:

Figure 7: An example of prompts for generating thoughts.