Temporal Knowledge Question Answering via Abstract Reasoning Induction

Ziyang Chen[1*] **Dongfang Li**[2*] **Xiang Zhao**[1†] **Baotian Hu**[2†] **Min Zhang**[2]
[1] Laboratory for Big Data and Decision, National University of Defense Technology, China
[2] Harbin Institute of Technology (Shenzhen), Shenzhen, China
{chenziyangnudt,xiangzhao}@nudt.edu.cn
{lidongfang,hubaotian,zhangmin2021}@hit.edu.cn

Abstract

In this study, we address the challenge of enhancing temporal knowledge reasoning in Large Language Models (LLMs). LLMs often struggle with this task, leading to the generation of inaccurate or misleading responses. This issue mainly arises from their limited ability to handle evolving factual knowledge and complex temporal logic. To overcome these limitations, we propose Abstract Reasoning Induction (ARI) framework, which divides temporal reasoning into two distinct phases: Knowledge-agnostic and Knowledge-based. This framework offers factual knowledge support to LLMs while minimizing the incorporation of extraneous noisy data. Concurrently, informed by the principles of constructivism, ARI provides LLMs the capability to engage in proactive, self-directed learning from both correct and incorrect historical reasoning samples. By teaching LLMs to actively construct knowledge and methods, it can significantly boosting their temporal reasoning abilities. Our approach achieves significant improvements, with relative gains of 29.7% and 9.27% on two temporal QA datasets, underscoring its efficacy in advancing temporal reasoning in LLMs. The code can be found at https://github.com/czy1999/ARI-QA.

1 Introduction

> *"Knowledge is not simply transmitted from teacher to student, but actively constructed in the mind of the learner."*
>
> — Jean Piaget

In practical scenarios, factual knowledge frequently undergoes evolution over time (Roddick and Spiliopoulou, 2002; Hoffart et al., 2011; Liang et al., 2023b, 2022). For instance, the host city of the Winter Olympic Games in 2018 was South Korea, while in 2022 it was Beijing. Despite their proficiency in various linguistic tasks, LLMs often exhibit limitations in efficiently processing and understanding tasks that involve temporal information. (Huang and Chang, 2023; Zhao et al., 2023a; Liang et al., 2023a).

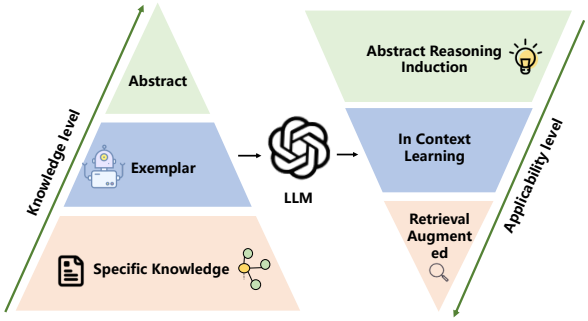

Figure 1: LLMs, when integrated with various levels of information, exhibit varying scopes of applicability; the more abstract and refined the knowledge, the broader its potential application.

Specifically, when tasks require complex temporal reasoning, LLMs tend to mislead the process and provide an inaccurate result. For instance, *"Which country's government leader visited China for the last time in 2015?"*, to answer this question, we need to (1) get which countries visited China in 2015 ; (2) filter out the country with the earliest visiting date. In step 1, LLMs easily meet hallucinations due to the incomplete training data and the uncertainty of parameterised knowledge. In step 2, LLMs may lead to the error because of the inaccuracy of the time filtering. Within such temporal reasoning tasks, any misjudgment in the temporal knowledge or errors during the temporal reasoning will culminate in erroneous conclusions. The problem might stem from the temporal unawareness of LLMs, impeding their ability to track and interpret events over time, particularly in situations requiring subtle and time-sensitive understanding.

Based on intuitive and empirical analysis, the cause accounting for the problem can be identi-

[*] Equal Contribution.
[†] Corresponding authors.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4872–4889

fied from two aspects: *lack of temporal knowledge* and *lack of complex temporal reasoning*. And the definition is given as follows:

Lack of Temporal Knowledge. LLMs acquire vast knowledge through pre-training on extensive datasets. However, the fixed nature of their parameters after training solidifies their knowledge base, which leads to LLMs' failure in understanding unseen and evolving knowledge.

Lack of Complex Temporal Reasoning. Owing to the inherent nature of large models that generate outputs based on maximum probability, they are limited in directly conducting complex reasoning. Facing the interconnected multi-step temporal reasoning, LLMs might accumulate errors during the process of probabilistic generation.

Despite the neglect of essences, current studies relatively approach above challenges. To augment the LLMs' capacity for understanding unseen and evolving information, researchers incorporate external knowledge to supply contextually relevant information, known as Retrieval Augmented Generation (RAG) (Zhao et al., 2023b; Baek et al., 2023; Sun et al., 2023). Although these methods enhance the richness of LLMs' responses, the retrieval accuracy and input length limitations might result in irrelevant noises and incomplete reasoning clues, degrading overall performance (Wang et al., 2023; Lu et al., 2023). Furthermore, although tailored examples serve as prompts to guide LLMs (Dong et al., 2023; Min et al., 2022), they are often inadequate for diverse practical tasks and require substantial efforts in time and human to acquire high-quality examples. In conclusion, above approaches fail to provide necessary guidance for ongoing temporal reasoning processes and are susceptible to incorporating extraneous noise, as shown in Figure 2.

To overcome the limitation, it is crucial to recognize that LLMs are inherently limited by reliance on passively absorbing training instances. Constructivism (Savery and Duffy, 1995; Kirschner et al., 2006), deeply embedded in philosophical and psychological schools of thought, contends that knowledge and learning emerge not from mere exposure to external information but through active construction. It asserts that learners synthesize new knowledge by building upon their existing understanding and experiences (Lake and Baroni, 2023). In this view, learning is an active and ongoing process wherein individuals continuously modify and refine their cognitive frameworks.

Inspired by the principles of constructivism, we try to steer LLMs towards an active and self-initiated learning approach, and propose an Abstract Reasoning Induction (ARI) framework. This will equip LLMs with the capacity for abstract synthesis and personalized knowledge application, enhancing relevance and utility in various contexts.

In details, to handle the lack of temporal knowledge, we transfer the data generation to an active process, consisting of two stages: *Knowledge-agnostic* and *Knowledge-based*. In knowledge-agnostic part, LLMs only need to choose potential steps. It is only in the knowledge-based part that the corresponding action is executed on the specific knowledge base to obtain the answer. This procedure offers factual knowledge support to LLMs while minimizing the incorporation of extraneous noisy data. On the other hand, to complete LLMs' complex temporal reasoning ability, ARI actively engages in proactive and self-directed learning from both correct and incorrect historical reasoning samples. This approach enables LLM to summarize and generalize methodologies (i.e. knowledge-agnostic step-by-step instructions) for different types of questions. When similar questions are encountered again, these abstract methods will guide the LLM to perform more efficient multi-step reasoning. By teaching LLMs to actively construct knowledge and methods, it can significantly boosting their temporal reasoning abilities without the need for further training.

In summary, our contribution is three-fold:

- Grounded in the principles of constructivism, we offer fresh perspectives for enhancing the reasoning capabilities and task adaptability of LLMs.
- We present ARI, a novel temporal reasoning framework that divides the process into two phases: *Knowledge-agnostic* and *Knowledge-based*. ARI enables LLMs to learn and construct proactively from historical reasoning samples, fostering a perpetual refinement of LLMs' reasoning abilities.
- The experimental results demonstrate that, compared to the leading TKGQA models, our approach achieves relative improvements of 29.7% and 9.27% respectively on two temporal QA datasets.

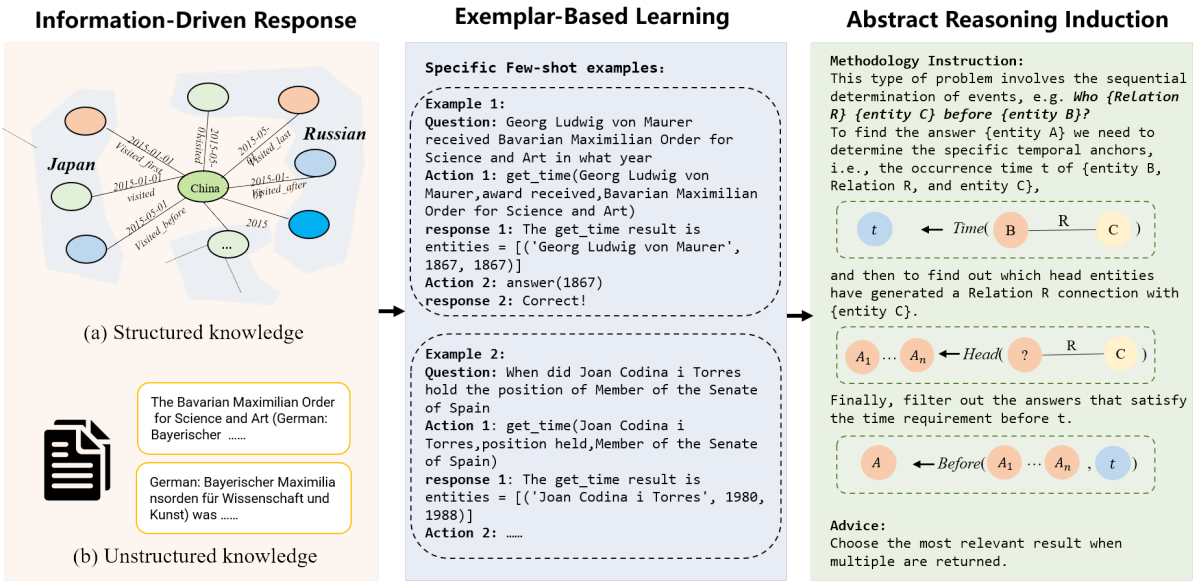

Figure 2: Three levels of information utilisation. *Information-Driven Response*, which extracts pertinent knowledge to form the basis of answers; *Exemplar-Based Learning*, offering cases of reasoning for the language model to assimilate and guide current inferences; and *Abstract Reasoning Induction*, providing step-wise abstract methodological guidance to the present question, distinct from concrete knowledge, thereby steering the language model's inference process.

2 Related Work

2.1 TKGQA Models

Traditional temporal knowledge graph question answering (TKGQA) methodologies fall into two categories. The first, exemplified by TEQUILA (Jia et al., 2018), deconstructs the initial question into sub-questions and temporal constraints, employing standard KGQA models for resolution, followed by a comparative analysis to select the most fitting answer. The second approach, such as CronKGQA (Saxena et al., 2021a), seeks to leverage TKG embeddings for semantic similarity assessments in answer determination, featuring a learnable reasoning process independent of handcrafted rules. Despite CronKGQA's proficiency with simpler inquiries, its performance falters with complex questions necessitating specific temporal inference. TempoQR (Mavromatis et al., 2021) addresses this by incorporating temporal scope data and employing the EaE method (Févry et al., 2020) to enrich question representation semantically.

However, traditional approaches rely on handcrafted rules or learnable representations, struggling with sophisticated temporal reasoning (Chen et al., 2022). In contrast, our model, leveraging the power of LLMs, excels in these challenging scenarios, showcasing superior adaptability and reasoning capabilities.

2.2 LLM Reasoning with External Information

Addressing hallucinations in generative models presents a compelling challenge, with one promising solution being the augmentation of LLMs with external knowledge (Mialon et al., 2023). Integration with an external knowledge base has become a prevalent strategy in question-answering and conversational tasks (Peng et al., 2023). There are mainly two approaches: explicit and implicit knowledge injection (Yang et al., 2023a). Explicit injection involves directly supplying LLMs with pertinent knowledge via prompts. For instance, KAPING (Baek et al., 2023) retrieves facts relevant to a query from a knowledge graph and appends these to the query as a prompt for the LLM, while CoK (Li et al., 2023a) first evaluates answer credibility and, if necessary, uses the LLM to decompose the question and generate various SPARQL queries to extract information from external knowledge bases. ChatKBQA (Luo et al., 2023) finetunes LLMs on KG structure to generate logical queries, which can be executed on KGs to obtain answers. Symbol-LLM (Xu et al., 2023) propose a dual-stage fine-tuning framework to integrates symbolic knowledge into LLMs, enhancing their reasoning capabilities. ToG (Sun et al., 2023) treats the LLM as an agent to interactively explore related entities and relations on KGs and perform reasoning based

on the retrieved knowledge. Implicit injection, on the other hand, subtly steers the LLM by incorporating knowledge semantic embeddings during reasoning or in the decoding process. KID (Liu et al., 2022) represents a novel decoding algorithm for generative LMs that dynamically infuses external knowledge at each step of LM decoding, and KPE (Zhao et al., 2023b) introduces a trainable parameter-sharing adapter to a parameter-freezing PLM for knowledge integration. Additionally, the incorporation of knowledge from other modalities such as visual information (Li et al., 2023c,b) has also become a method of knowledge introduction.

While the integration of knowledge into LLMs can mitigate issues of hallucinations, it is not without challenges. Explicit knowledge injection often struggles to acquire high-quality, relevant information, and is constrained by the finite-length contexts. Implicit injection typically necessitates fine-tuning of parameters, which can be prohibitively costly. We address these limitations by dividing temporal knowledge reasoning into two distinct components: knowledge-related and knowledge-agnostic. This approach achieves a clear separation between knowledge and reasoning, thereby circumventing the aforementioned constraints.

2.3 LLM Reasoning with Memories

Memory plays a pivotal role in human intelligence (Atkinson and Shiffrin, 1968). Given that LLMs inherently lack long-term memory and their short-term memory is constrained by the scope of their context window, numerous studies have embarked on the journey to equip LLMs with memory capabilities (Pan et al., 2023; Zhong et al., 2023). Instead of the conventional approach where accumulated conversations are retrieved directly, MemoChat (Lu et al., 2023) innovatively constructs and updates a structured, instant memo that categorizes past dialogues. Conversations are then fetched based on their specific topics and summaries. Reflexion (Shinn et al., 2023) exploits a working memory to store experiences for a dedicated task to improve the performance of the agent through several trials. However, the histories stored in working memory cannot benefit the episode for different task goals. MemPrompt (Madaan et al., 2022) designs a persistent memory to store human feedback to remind the chatbot of the conversational information and improve it continuously. RLEM (Zhang et al., 2023) adopts a persistent environment-grounded experience memory to store the experiences and assist in future decision-making even for different task goal. Thought Propagation (Yu et al., 2023) emphasizes the ability to explore and apply insights from analogous solutions. By delving into and utilizing solutions from problems related to the given issue, it improves performance and accuracy across various tasks.

However, current memory-enhanced methods are limited to passively received historical information, overlooking the active construction of abstract knowledge based on previous experience. Starting from constructivism, we apply the proposed method to provide large models with an active and continuous learning process, offering knowledge that is abstract and generalized.

3 Method

Algorithm 1 Abstract Reasoning Induction

Require: Temporal knowledge graph $\mathcal{K}$, question q, historical memory H_q, abstract methodology instruction set M_C

Ensure: Answer to the question q

$M_C \leftarrow LLM(H_q)$ (4)
Initialize subject entity e_h from q
Find 1-hop subgraph G_{e_h} of e_h in $\mathcal{K}$ (1)
Enumerate initial candidate actions P_0 from G_{e_h} (2)
while $LLM(M_{C^*}, P'_{t_i}) \neq answer(a)$ **do**
  Filter candidate actions P_{t_i} to get P'_{t_i} (3)
  $C_t^* \leftarrow$ findKmeansCluster(q) (5)
  $a_i^* = LLM(M_{C^*}, q, P'_{t_i})$ (6)
  Execute selected action a_i^* and update current environment
  Regenerate candidate actions for the next step (2)
  if $t_i \geq t_{max}$ **then**
    Break
  end if
end while
Execute final action a_i^* to abtain the answer
Add current process to H_q
return Answer derived from the reasoning process

3.1 Task Definition

Given a Temporal Knowledge Graph (TKG) $\mathcal{K}$ and a natural language question q, TKGQA aims to

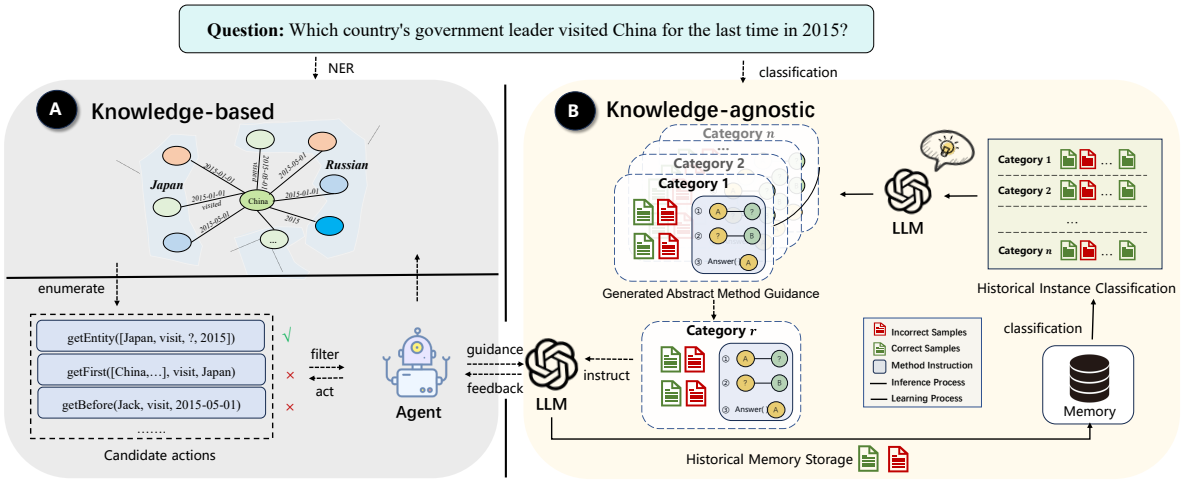

Figure 3: Model architecture of ARI. Our framework divides temporal reasoning into two distinct phases: *Knowledge-agnostic* and *Knowledge-based*. This division aims to reduce instances of hallucinations and improve LLM's capacity for integrating abstract methodologies derived from historical experience. See the detailed instructions in Appendix A.4.

extract an entity $s/o \in \mathcal{E}$ or a timestamp $\tau \in \mathcal{T}$ that correctly answers the question q. For instance, for the question *'Which country's government leader visited China for the last time in 2015?'* Based on the event information contained in the TKG, we can get the answer to the question is the KG entity *Vietnam*.

3.2 ARI Framework

The constructivist perspective posits that knowledge does not merely encapsulate universal laws but must be contextually reconstructed for specific situations. This view emphasizes that understanding is a construct developed by the learner, uniquely shaped by their experiential background and dependent on their learning trajectory in a particular context (Kirschner et al., 2006; Savery and Duffy, 1995). In line with this philosophy, we introduce ARI framework. We overview the framework in Figure 3. Diverging from previous research that directly feeds knowledge into LLMs , we divide temporal knowledge reasoning into two parts: *knowledge-agnostic* and *knowledge-based*.

The knowledge-based module extracts relevant TKG subgraphs based on the given question, generating all feasible fine-grained actions through traversal (cf. § 3.3). This module encompasses all operations and interactions with knowledge, freeing the LLM from the vast, noisy specific information to focus on reasoning. This design approach allows us to build intricate knowledge queries by combining fine-grained atomic operations, while effortlessly adapting to various knowledge bases (see the detailed operations in Appendix A.3). In the knowledge-agnostic module, the LLM performs high-level strategic decisions and candidate action selection. We have innovated mechanisms that allow LLMs to internalize lessons from past decisions, forming generalized abstract methodologies (solid line process in Figure 3). This foundation empowers the LLM to proactively develop abstract methodological guidelines for various question types, enhancing its ability to reason on new questions efficiently (cf. §3.4).

For the inference process, ARI first categorizes the question and selects the most suitable abstract methodological guidance. According to these guidelines, the LLM interacts multiple turns with the knowledge-agnostic module to reason and gradually solve the problem, as shown in the dashed line process in Figure 3. We also present a specific reasoning example in Figure 5 of Appendix A.4.

3.3 Knowledge-based Interaction

In the knowledge-based interaction part, we frame complex temporal knowledge reasoning challenges as multi-step inferentce tasks (Gu and Su, 2022; Gu et al., 2023). At the beginning of each step, we employ an filtering mechanism to engage with the TKG and the current question. This interaction produces a set of feasible candidate actions for each step. The LLM then selects the most suitable action from these candidates. Following this selection, the model interacts with the TKG, updating the initial

state for the next step in a recursive process.

Candidate Action Enumeration. Specifically, given a complex temporal question q and a TKG $\mathcal{K} := (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ denote entities, relations, and timestamps, respectively. Starting from the subject entity e_h of q, we first find the 1-hop subgraph of e_h in $\mathcal{K}$. Let N_{e_h} be the set of nodes in the 1-hop (undirected) neighborhood of e_h in the TKG, and R_{e_h} be the corresponding edge.

$$G_{e_h} = \{(e, r) | e \in N_{e_h}, r \in R_{e_h}\}, \quad (1)$$

where G_{e_h} is the corresponding 1-hop subgraph of e_h. For each edge $r \in G_{e_h}$, our agent will strictly follow finely-grained atomic operations in Appendix A.3, traversing and replacing the relations and entities present in the current G_{e_h} to construct the set of candidate actions P_0, where the subscript 0 denotes the candidate actions for the initial step,

$$P_0 = \{Enum(action, e, r) | e, r \in G_{e_h}\}. \quad (2)$$

Candidate Action Filtration. However, due to the continual occurrence and updating of temporal events, even the scale of a 1-hop subgraph can be vast. This results in an excessively large set of generated candidate actions, which can significantly impede the judgment of LLMs (cf. § section 4.2). Consequently, we propose a filtration process for candidate actions, retaining only those that are correct, feasible, and semantically relevant.

Specifically, for each action a within set P_0, we execute the corresponding function on the TKG. If the function returns a non-empty value, the action is considered correct and feasible; otherwise, it is discarded. Among all remaining actions, we retain the top-K actions based on the calculation of their semantic similarity to the question q,

$$P_0^{'} = \{a | \text{exec}(a) \neq \emptyset \wedge a \in \text{Top-}K(P_0, q)\}. \quad (3)$$

Based on the LLM's decision, the agent executes the corresponding action, thereby updating the current environment. Subsequently, it regenerates the next set of candidates P_1 based on the newly identified subject entities, repeating this process until a termination command is received.

3.4 Knowledge-agnostic Reasoning

The knowledge-agnostic module enables LLMs to distill and apply abstract methodologies from historical reasoning examples, enabling adaptation to diverse questions. This approach fosters a general methodology, applicable across various domains and to a wide range of knowledge-independent inquiries, enhancing LLMs' versatility.

Historical Memory Storage and Learning . In the LLM's reasoning process, we meticulously document the current state at each step t_i, encompassing the current temporal question q, the set of candidate actions P_t, and the LLM's decision a_t. The aggregate of all stepwise states for a given question forms the historical decision set H_q,

$$H_q = \{(q, t_i, P_{t_i}, a_{t_i}) \mid i \in T\}, \quad (4)$$

where T is the set of all steps in the process.

Temporal reasoning is often multi-step and complex, yet the types of reasoning involved tend to be consistent, with similar questions requiring similar inference steps. Therefore, once the LLM conducts reasoning and accumulates a series of historical inferential steps, we employ unsupervised clustering K-means (MacQueen, 1967) to categorize these historical steps into distinct clusters C_H.

After the LLM engages in reasoning and compiles historical reasoning steps, these are subjected to unsupervised clustering to form distinct clusters. Each cluster contains a mix of both accurate and erroneous reasoning processes. We enable the LLM to actively learn from specific historical instances within each cluster and distill abstract methodologies independent of domain-specific knowledge.

LLM Decision with Abstract Reasoning. When addressing new inference challenges, we initiate the process by identifying the historical reasoning cluster most closely aligned with the new question. We then extract its abstract methodologies to guide the LLM in its reasoning for the current question.

Specifically, for a given question q, we calculate the similarity score $S(C_i, q)$ with each historical reasoning cluster C_i. We then retrieve the abstract method instruction M_{C^*} from the cluster that yields the maximum S. Let $\{C_1, C_2, \ldots, C_n\}$ be the set of historical reasoning clusters. For a given question q, we calculate the similarity score $S(C_i, q)$ for each cluster C_i, and select the cluster C^* with the highest similarity score to the query q,

$$C^* = \underset{C_i}{\text{argmax}}\, S(C_i, q). \quad (5)$$

The abstract method instruction M_{C^*} is then

selected from the cluster C^* such that:

$$a_i^* = LLM(M_{C^*}, q, P_i), \tag{6}$$

where M_{C^*} is the abstract method instruction of C^*, and a_i^* is the final output of LLM. The reasoning sequence concludes when the LLM outputs a termination action or when the length of reasoning steps exceeds the predetermined maximum threshold.

4 Experiment

4.1 Implementation Details

We use `gpt-3.5-turbo-0613` as our LLM (More experiments and analyses of other LLMs can be found in Section A.7). We configure the LLM to access and investigate a corpus of 200 historical reasoning samples, with the maximum length of reasoning path set to 5, and the number of historical path categories fixed at 10. Due to the vast size of the test set, which comprises more than 50,000 question-answer pairs, we employ a stratified sampling approach for evaluation, extracting a subset of 200 questions from the test set for each iteration. In our evaluation, we compare several baseline methods, including the traditional TKGQA models and LLM-based models (see Appendix A.1 and A.2 for more details).

4.2 Overall Results

Table 1 presents the comparative results of ARI against other baselines on MULTITQ.

LLM only Performance. ChatGPT's performance on two datasets reveal a significant shortcoming in its application to temporal knowledge reasoning, even when all the knowledge required for the questions is within the scope of its training data prior to 2021. This deficiency is particularly pronounced when compared to traditional TKGQA methods, suggesting that the parameterised knowledge acquired by LLMs is not seamlessly transferred to temporal reasoning tasks. A stark contrast in performance is observed between the MULTITQ and CRONQUESTIONS datasets, the latter benefiting from ChatGPT's extensive training on `WikiData`(Vrandecic and Krötzsch, 2014). This discrepancy points to a significant challenge: MULTITQ's reliance on the `ICEWS`(Boschee et al., 2015), which encompasses more niche and frequent events, poses a difficulty for LLMs due to their limited exposure to such data during training. This contrast elucidates the inherent limitations of LLMs in processing temporal knowledge.

Reasoning with External Knowledge. Incorporating additional knowledge graph data into LLMs, as seen with KG-RAG, considerably enhances their performance in knowledge-intensive QA tasks. KG-RAG outperforms ChatGPT by 81% and 96% across two datasets, respectively. Nevertheless, it still falls short of leading TKGQA models. This gap can be attributed to two main factors. First, the vast and complex nature of temporal information presents a challenge. A single question may involve thousands of related events, which cannot be accurately incorporated through prompts alone, leading to insufficient background information for reasoning. Second, the retrieved external knowledge often contains redundant or irrelevant information, which can further mislead the model's inference process.

Reasoning with Exemplar Guidance. The CoT $_{KB}$ approach, despite providing step-by-step reasoning guidance and knowledge-based interaction, does not reach the efficacy levels of ARI. This discrepancy stems from the exemplar-based learning method's dependence on specific instances, which can introduce extraneous knowledge and detract LLMs from focusing on reasoning methodologies, leading to inaccurate conclusions. Additionally, its static examples fail to provide the customized guidance necessary for diverse reasoning questions. Similarly, our investigation also extends to the augmented knowledge-agnostic module in ReAct, designed to mitigate the generation of infeasible actions. Despite this enhancement, a performance gap persists when compared to ARI, underscoring similar limitations identified in the CoT $_{KB}$ approach. Despite ReAct $_{KB}$'s interaction with the environment, it still lacks customized abstract guidance for the current reasoning question.

ARI significantly outperforms current state-of-the-art TKGQA models, achieving a relative improvement of 29.7% on the MULTITQ dataset and a 9.27% increase in performance on the CRONQUESTIONS dataset. These substantial gains can be attributed to the knowledge adaptability and the abstract methodology instruction mechanism, which empower LLMs to make advanced decisions. By leveraging abstract methodologies, LLMs can select optimal temporal reasoning steps without engaging with the specifics of the underlying knowledge.

Model	MULTITQ					CRONQUESTIONS				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Simple	Complex	Entity	Time		Simple	Complex	Entity	Time
BERT	0.083	0.092	0.061	0.101	0.040	0.243	0.249	0.239	0.277	0.179
ALBERT	0.108	0.116	0.086	0.139	0.032	0.248	0.255	0.235	0.279	0.177
EmbedKGQA	0.206	0.235	0.134	0.290	0.001	0.288	0.290	0.286	0.411	0.057
CronKGQA	0.279	0.134	0.134	0.328	0.156	0.647	0.987	0.392	0.699	0.549
MultiQA	0.293	0.347	0.159	0.349	0.157	-	-	-	-	-
ChatGPT	0.102	0.147	0.077	0.137	0.002	0.249	0.250	0.247	0.246	0.253
KG-RAG	0.185	0.200	0.160	0.230	0.07	0.490	0.460	0.518	0.470	0.520
CoT $_{KB}$	0.240	0.440	0.120	0.220	0.320	0.640	0.690	0.610	0.620	0.660
ReAct $_{KB}$	0.310	0.635	0.136	0.313	0.300	0.685	0.835	0.525	0.650	0.755
ARI	**0.380****	**0.680****	**0.210****	**0.394****	**0.344****	**0.707****	0.860	**0.570****	**0.660***	**0.800***

Table 1: Performance of baselines and our methods on the MULTITQ and CRONQUESTIONS. $^{*}(p \leq 0.05)$ and $^{**}(p \leq 0.005)$ indicate paired t-test of ARI versus the best baseline.

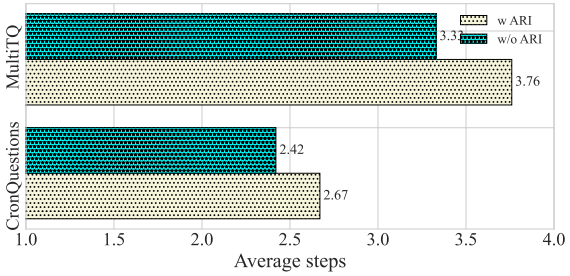

Figure 4: Comparison of average reasoning steps of ARI on MULTITQ.

Model	Accuracy (%)	
	MULTITQ	CRONQUESTIONS
ARI	38.0	70.7
w/o Abstract Guidance	30.5	67.1
w/o History Cluster	34.5	68.9
w/o Action Filter	33.1	66.5
w/o Incorrect Examples	36.5	69.2

Table 2: Ablation results of ARI.

Comparison of Reasoning Efficiency. To validate the effectiveness of abstract instruction, we conduct an evaluation of reasoning efficiency. On the test set, with all other components of the model remaining constant, we remove the abstract instruction and record the average number of steps taken for reasoning. Compared with the ARI, we observe that under the guidance of abstract methodologies, LLMs not only improve in reasoning accuracy but also reduce their average number of reasoning steps by 11.4% on MULTITQ and 9.3% on CRONQUESTIONS. This underscores that the guidance provided by abstract methodologies can significantly enhance the efficiency of LLMs in temporal reasoning tasks.

4.3 Ablation Study

To evaluate the efficacy of the individual components of the model, we conducted ablation studies.

Initially, we remove the abstract guidance component, requiring the LLM to rely on its own understanding of the questions without the aid of historical information. This result in significant performance drops on both datasets, with a 19.7% decrease on MULTITQ and a 3.7% decrease on CRONQUESTIONS. This suggests that distilled abstract guidance plays a substantial role in supporting the model's reasoning capabilities.

To further assess the impact of abstract guidance, we eliminate the clustering module, thus deriving a universal abstract guidance from all historical reasoning processes without categorization based on question type. The model performance dropped by 9.2% on MULTITQ and 2.5% on CRONQUESTIONS, indicating that a singular abstract methodology is insufficient for guiding various types of questions and that targeted abstract methodological guidance is more effective. In Appendix A.5, we illustrate the impact of varying cluster quantity on the model's final reasoning performance.

To verify the role of incorrect samples in ARI, we remove incorrect examples from the process of generating abstract methods, providing only correct examples as guidance. As evident from the results, the removal of incorrect examples leads to a decrease in the quality of abstract guidance, resulting in a performance drop. By encountering and learning these incorrect examples, LLMs become more adept at avoiding similar pitfalls in subsequent reasoning tasks. This also aligns with our intuition and has been validated in prior studies (Wang and Li, 2023; Yang et al., 2023b; An et al., 2023).

Lastly, we remove the action selection module, allowing the LLM to choose from all generated actions without filtering. This led to a decrease in performance on both datasets by 12.8% and 5.9%, underscoring that unfiltered actions result in an excessive number of options, including irrelevant ones, which hinders the LLM's reasoning and complicates the decision-making process.

5 Conclusion and Limitation

This study, anchored in the principles of constructivism, critically examines the shortcomings of LLMs in addressing complex temporal reasoning challenges and proposes an innovative approach to augment their reasoning capabilities. Through the integration of a knowledge adaptability framework and abstract methodological guidance, we have shown that LLMs can attain more precise and efficient reasoning in complex temporal scenarios, effectively overcoming their constraints in processing and interpreting time-sensitive knowledge.

Limitations. While ARI demonstrates impressive results, it also presents several limitations for ongoing refinement. Firstly, The efficacy of generating abstract guidance heavily relies on the capabilities of LLMs. Smaller-scale LLMs may struggle to produce high-quality abstract guidance, thus potentially restricting their application. Secondly, the ARI framework depends on multi-step reasoning to arrive at final answers, a process moderately influenced by the LLM's reasoning efficiency, which extends the duration of inference. Finally, our method is primarily concentrated on complex temporal reasoning, with its effectiveness in other reasoning domains remaining to be examined. Future research should aim to refine these methods to make them more adaptable to various models and problem domains, enhance the balance between reasoning efficiency and depth, and expand their scope to include a broader range of reasoning tasks.

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful and constructive comments, which greatly contributed to improving the quality of the paper. This work was partially supported by National Key R&D Program of China (No. 2022YFB3103600), NSFC (Nos. U23A20296, 62272469,62376067), The Science and Technology Innovation Program of Hunan Province (No. 2023RC1007), and Guangdong Basic and Applied Basic Research Foundation (2023A1515110078).

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes LLM better reasoner. *CoRR*, abs/2310.20689.

Richard C. Atkinson and Richard M. Shiffrin. 1968. Human memory: A proposed system and its control processes. In *The psychology of learning and motivation*.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *CoRR*, abs/2306.04136.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS Coded Event Data.

Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023. Multi-granularity temporal question answering over knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowl. Based Syst.*, 251:109134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *ArXiv*, abs/2301.00234.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4937–4951. Association for Computational Linguistics.

Yu Gu, Xiang Deng, and Yu Su. 2023. Don't generate, discriminate: A proposal for grounding language models to real-world environments. In *Proceedings of the 61st Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 4928–4949. Association for Computational Linguistics.

Yu Gu and Yu Su. 2022. Arcaneqa: Dynamic program induction and contextualized encoding for knowledge base question answering. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1718–1731. International Committee on Computational Linguistics.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 229–232. ACM.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1049–1065. Association for Computational Linguistics.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. TEQUILA: temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1807–1810. ACM.

Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41:75 – 86.

Brenden M Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, pages 1–7.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq R. Joty, Soujanya Poria, and Lidong Bing. 2023a. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources.

Yunxin Li, Baotian Hu, Xinyu Chen, Lin Ma, and Min Zhang. 2023b. Lmeye: An interactive perception network for large language models. *arXiv preprint arXiv:2305.03701*.

Yunxin Li, Baotian Hu, Wei Wang, Xiaochun Cao, and Min Zhang. 2023c. Towards vision enhancing llms: Empowering multimodal knowledge storage and sharing in llms. *arXiv preprint arXiv:2311.15759*.

Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu. 2023a. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–12.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. 2023b. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1559–1568.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767*.

Ruibo Liu, Guoqing Zheng, Shashank Gupta, Radhika Gaonkar, Chongyang Gao, Soroush Vosoughi, Milad Shokouhi, and Ahmed Hassan Awadallah. 2022. Knowledge infused decoding. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *CoRR*, abs/2308.08239.

Haoran Luo, Haihong E, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, and Wei Lin. 2023. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *CoRR*, abs/2310.08975.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2833–2861. Association for Computational Linguistics.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Soji Adeshina, Phillip R. Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2021. Tempoqr: Temporal question reasoning over knowledge graphs. *CoRR*, abs/2112.05785.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *CoRR*, abs/2302.07842.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *ArXiv*, abs/2202.12837.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *CoRR*, abs/2308.03188.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *CoRR*, abs/2302.12813.

John F. Roddick and Myra Spiliopoulou. 2002. A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. Knowl. Data Eng.*, 14(4):750–767.

John R. Savery and Thomas M. Duffy. 1995. Problem based learning: An instructional model and its constructivist framework. *Educational Technology archive*, 35:31–38.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. 2021a. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6663–6676. Association for Computational Linguistics.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. 2021b. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6663–6676. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4498–4507. Association for Computational Linguistics.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *CoRR*, abs/2303.11366.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *CoRR*, abs/2307.07697.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Danqing Wang and Lei Li. 2023. Learning from mistakes via cooperative study assistant for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10667–10685. Association for Computational Linguistics.

Qingyue Wang, Liang Ding, Yanan Cao, Zhiliang Tian, Shi Wang, Dacheng Tao, and Li Guo. 2023. Recursively summarizing enables long-term dialogue memory in large language models. *CoRR*, abs/2308.15022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu. 2023. Symbol-llm: Towards foundational symbol-centric interface for large language models. *CoRR*, abs/2311.09278.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2023a. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. *CoRR*, abs/2306.11489.

Zeyuan Yang, Peng Li, and Yang Liu. 2023b. Failures pave the way: Enhancing large language models through tuning-free rule accumulation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1751–1777. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Junchi Yu, Ran He, and Rex Ying. 2023. Thought propagation: An analogical approach to complex reasoning with large language models. *CoRR*, abs/2310.03965.

Danyang Zhang, Lu Chen, Situo Zhang, Hongshen Xu, Zihan Zhao, and Kai Yu. 2023. Large language model is semi-parametric reinforcement learning agent. *arXiv preprint arXiv:2306.07929*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023a. A survey of large language models. *CoRR*, abs/2303.18223.

Ziwang Zhao, Linmei Hu, Hanyu Zhao, Yingxia Shao, and Yequan Wang. 2023b. Knowledgeable parameter efficient tuning network for commonsense question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9051–9063. Association for Computational Linguistics.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *CoRR*, abs/2305.10250.

A Appendix

A.1 More Details about Baseline Methods

In our evaluation, we compared several baseline methods.

- **Pre-trained LMs**: To evaluate BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2020), we generate their LM-based question embedding and concatenate it with the entity and time embeddings, followed by a learnable projection. The resulted embedding is scored against all entities and timestamps via dot-product.
- **EmbedKGQA** (Saxena et al., 2020) is designed with static KGs. To deal with multiple temporal granularities, timestamps are ignored during pre-training and random time embeddings are used.
- **CronKGQA** (Saxena et al., 2021a) is designed for single temporal granularity. To deal with multiple granularities, time embeddings at the year/month granularity are drawn at random from corresponding day embeddings.
- **MultiQA** (Chen et al., 2023) is designed for multi-granularity temporal granularity with a transformer-based time aggregation module.
- **ChatGPT** [*]. We use ChatGPT to provide direct answers to the questions.
- **KG-RAG**. To validate the performance of the LLM in the presence of relevant background knowledge, we extracted relevant quaternions (up to 20) from the TKG based on the entity and time information appearing in the question, and put them in the prompt for ChatGPT to answer as a retrieval-enhanced way of comparison.
- **ReAct $_{KB}$**:To address the applicability of ReAct (Yao et al., 2023) to our task, we designed a variant of ReAct by integrating our knowledge-agnostic module, which generates all feasible actions (using the same atomic action templates as ARI). LLMs were then prompted to select one action from the available options. Parameter settings remained consistent with our ARI approach, including the use of the same Named Entity Linking (NEL) method and reasoning length.
- **CoT $_{KB}$**:We introduce the CoT $_{KB}$ method, which integrates a knowledge-based module into the Chain-of-Thought (CoT) (Wei et al., 2022) framework. This allows the LLM to interact with KG under the guidance of examples, thereby obtaining the final answer more effectively. We manually constructing 9 specific instance examples with detailed reasoning steps to guide the LLM, while keeping other settings unchanged.

[*]https://chat.openai.com/

		Train	Dev	Test
Single	Equal	135,890	18,983	17,311
	Before/After	75,340	11,655	11,073
	First/Last	72,252	11,097	10,480
Multiple	Equal Multi	16,893	3,213	3,207
	After First	43,305	6,499	6,266
	Before Last	43,107	6,532	6,247
Total		386,787	587,979	54,584

Table 3: Statistics of question categories in MULTITQ.

		Train	Dev	Test
Simple	Simple Entity	90,651	7,745	7,812
	Simple Time	61,471	5,197	5,046
Complex	Time Join	55,453	3,878	3,832
	First/Last	118,556	11,198	11,159
	Before/After	23,869	1,928	2,151
Total		350,000	30,000	30,000

Table 4: Statistics of question categories in CRONQUESTIONS.

A.2 Datasets Statistics

CRONQUESTIONS (Saxena et al., 2021b) is a dataset for temporal knowledge graph question answering. The entities and times present in the questions are annotated. CRONQUESTIONS has four question types, including both simple and complex temporal questions.

MULTITQ (Chen et al., 2023) is a complex temporal question answering dataset with multi-granularity temporal information. Compared to existing datasets, MULTITQ features in a few advantages, including large scale, ample relations and multiple temporal granularity, which hence better reflects real-world scenarios.

We summarize the number of questions in MULTITQ across different types in Table 3 and Table 4. In Table 5, we present sample questions from MULTITQ as per question type, time granularity and answer type.

Property	Sample Question
By question type	
Equal	*Which country provided humanitarian aid to Sudan in 2007?*
Before/After	*Who commended the Military of Mali before the Armed Rebel of Mali did?*
First/Last	*When did the Militant of Taliban first commend the Government of Pakistan?*
Equal Multi	*In 2012, who last did Barack Obama appeal for?*
Before Last	*Who was threatened by Benjamin Netanyahu last before Middle East?*
After First	*Who first wanted to negotiate with Evo Morales after the Citizen of Brazil did?*
By time granularity	
Year	*Who first made Abu Sayyaf suffer from conventional military forces In 2015?*
Month	*In Dec, 2008, who would wish to negotiate with the Senate of Romania?*
Day	*In Jul 21st, 2011, who criticized the Media of Ecuador?*
By answer type	
Entity	*Which country visited Japan in 2013?*
Time	*When did China express intent to meet with the Government of Pakistan?*

Table 5: Representative examples from MULTITQ.

A.3 Action Templates

Our approach is designed to be both generalizable and scalable. The templates in ARI are not highly manually defined high-level functions, but rather finely-grained atomic operations (e.g., extracting time, entity extraction, etc.). These atomic operations can be flexibly combined to generalize a wide range of complex actions, demonstrating their versatility and extensibility.

Our action templates in ARI strictly follow the definition of functions in Table 6. We employ several specialized functions to facilitate precise information retrieval. The *getTime* function retrieves the timing of specific events, based on given entities and relation. For temporal positioning, *getBefore*, *getAfter*, and *getBetween* identify entities or events relative to specified time frames. In terms of entity queries, *getTailEntity* and *getHeadEntity* ascertain linked entities based on existing relation, with an optional time constraint. For queries targeting specific time instances, *getFirst* and *getLast* pinpoint entities with the earliest and latest occurrences, respectively. Responses are then articulated using the *answer* function, providing a streamlined method for answering queries within the TKG.

Our method exhibits low coupling with templates, making it adaptable to new data and domains. The extensibility of atomic templates is straightforward, allowing for easy incorporation of additional templates as needed. For instance, if we were to extend our approach to handle spatio-temporal data questions, adding a spatial atomic operation would be a straightforward task without the need for significant modifications.

A.4 Details about the Instruction Format

In Figure 5, we illustrate an example of reasoning using the ARI model. Table 8 shows some exemplars of ARI. During each step of the process, the LLM receives guidance from abstract methods and selects the optimal action from available paths, continuing until it deems an answer has been sufficiently formulated or the maximum reasoning length is reached. Figure 8 presents the complete set of instructions used in our experiments, comprising components such as task definition, functional interpretations of potential actions, the current temporal question under consideration, historical reasoning steps, available candidate actions for the current round, feedback from the previous round's action, and requirements for output formatting.

A.5 Impact of Cluster Quantity

In Figure 7, we show the reduced dimensional clustering diagram for the 10 categories of questions in the experiment. To verify the effect of different number of clusters on the results, we present the impact of the number of historical reasoning process clusters on the results. As shown in Figure 6. We

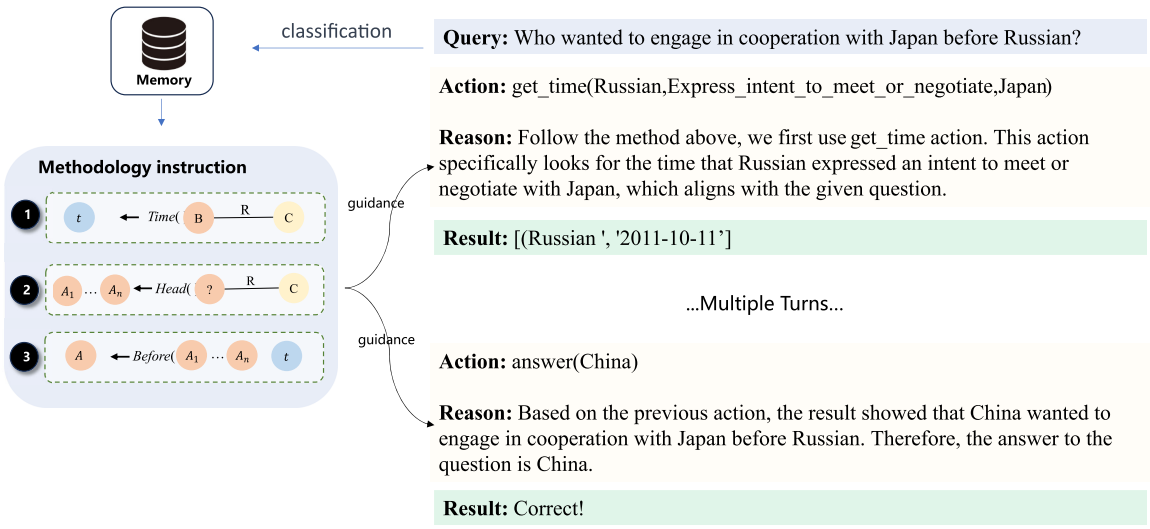

Figure 5: A demonstration sample of ARI reasoning on MULTITQ.

Action template	Comments
`getTailEntity(head,rel,time)`	Identify the tail/object entity based on the head/subject entity and relation
`getHeadEntity(tail,rel,time)`	Identify the head/suject entity based on the tail/object entity and relation
`getTime(head,rel,tail)`	Retrieve the time of a specific event based on the head entity, relation and tail entity
`getBetween(entities,Time1,Time2`	Identify entities/events that occurred between two specific times
`getBefore(entities,time)`	Identify entities/events that occurred before a given time
`getAfte(entities,time)r`	Identify entities/events that occurred after a given time
`getFirst(entities,time)`	Pinpoint entities with the earliest occurrence
`getLast(entities,time)`	Pinpoint entities with the latest occurrence
`answer(entities/time)`	To provide your answer, use the `answer` function

Table 6: Action templates in ARI.We employ these specialized functions to facilitate precise information retrieval.

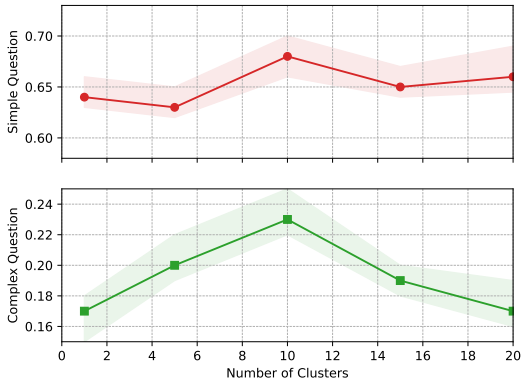

Figure 6: Accuracy v.s Number of Clusters of ARI on MULTITQ.

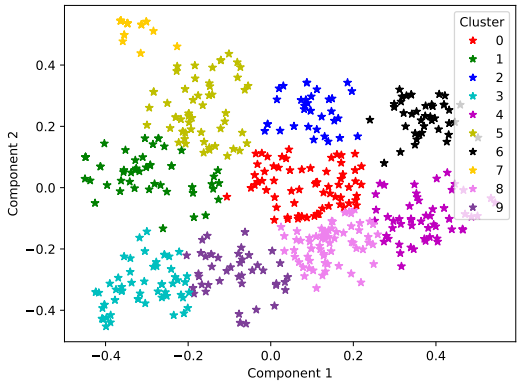

Figure 7: Clustering results for historical inference questions.

observe an initial increase followed by a decline in performance for both simple and complex problems. This pattern can be attributed to the fact that when the number of clusters is too low, the LLM is unable to distill concise and effective abstract methods from the noisy and abundant historical paths. Conversely, when the number of clusters is too high relative to a fixed number of historical samples, each category contains too few samples to provide the LLM with sufficient information to refine abstract methods. Thus, we observe a trend of improvement that eventually reverses as the number of clusters increases.

A.6 Error Analysis

For error analysis, we randomly sample 100 error instances from the test set and summarized the following three types of typical errors: (1) Retrieving irrelevant entities (in MULTITQ), meaning the model obtained wrong entities from the KG; Although our entity linking model can achieve a high prediction accuracy, wrong entities still exist in some questions. (2)Low-quality abstract methodological guidance. Within the dataset, there exist complex problems for which the historical reasoning processes consistently led to incorrect conclusions. This lack of sufficient correct reasoning histories hampers the LLM's ability to synthesize and refine effective abstract methodologies. Consequently, the low-quality abstract methods derived by the LLM prove inadequate in guiding subsequent reasoning processes, leading to a cascade of errors. (3) Uncertainty outputs of LLMs. Despite the constraint that LLMs can only select from candidate actions or provide final answers, there are instances where they do not strictly adhere to the given instruction. This non-compliance leads to the failure of our predefined graph query methods, consequently impeding effective reasoning.

This demonstrates that more efforts are needed to strengthen the model's reasoning capabilities, particularly in enhancing the reasoning capabilities of LLMs and diversifying their reasoning processes. It is crucial to provide a richer array of effective historical information for the generation of abstract methods. This approach is vital to prevent the LLMs from falling into a repetitive cycle of errors.

A.7 Generalizability of Method on Other LLMs

Model	LLM only	ARI
Llama-2 Chat 7B	0.040	0.105
GPT-4	0.125	0.411

Table 7: Accuracy of ARI with Other LLMs on MULTITQ

To assess the effectiveness of the ARI across various LLMs, we conducted experiments using the open-source model `Llama-2-7B-chat` and `GPT-4`. Our findings indicate that models with greater inherent capabilities yield better direct inference outcomes, presumably due to the acquisition of more extensive knowledge during training. Furthermore, the performance enhancements in ARI models built upon these base models are more pronounced. Notably, the ARI model based on `GPT-4` achieved a score of 0.411 on the MULTITQ benchmark, representing a relative improvement of 40.2% over the current state-of-the-art models. This significant advancement underscores our model's robust generalization capabilities.

Our work focuses on enhancing LLM reasoning with abstract guidance, minimizing irrelevant noise by isolating them from direct knowledge sources. Unlike mainstream LLM agents designed for interactive environments, we employ a knowledge-agnostic module for environmental interactions, allowing the LLM to make decisions based on abstract guidance alone. Additionally, the study is open to incorporating interactive decision-making elements from recent LLM advancements, aiming to enhance the project's effectiveness and broaden its applicability in future research.

Question 1: In which month did the City Mayor of Philippines first praise Ona?

```
Action 0    getTime(City_Mayor_(Philippines),Praise_or_endorse,Ona)
Response 0  entities = [('City_Mayor_(Philippines)', '2014-10-07')]
Action 1    getFirst({entities})
Response 1  entities = [('City_Mayor_(Philippines)', '2014-10-07')]
Action 2    answer(2014-10)
Response 2  Correct!
```

Question 2: Before Mali's militant, which country was the last to criticise France?

```
Action 0    getTime(Militant_(Mali),Criticize_or_denounce,France)
Response 0  entities = [('Militant_(Mali)', '2013-01-21')]
Action 1    getFirst({entities})
Response 1  entities = [("Government_(Cote_d'Ivoire)", ...]
Action 2    getLast({entities})
Response 2  entities = [('Royal_Administration_(Spain)', '2015-12-24')]
Action 3    answer(Royal_Administration_(Spain))
Response 3  Correct!
```

Question 3: Who was the last to give a criticism to Iran before Pervez Musharraf did?

```
Action 0    getTime(Pervez_Musharraf,Criticize_or_denounce,Iran)
Response 0  entities = [('Pervez_Musharraf', '2006-01-17')]
Action 1    getHeadEntity(Iran,Criticize_or_denounce,no time)
Response 1  entities = [('Shirin_Ebadi', '2005-01-05'), ...]
Action 2    getBefore({entities},2006-01-17)
Response 2  entities = [('Shirin_Ebadi', '2005-01-05'),...]
Action 3    getLast({entities})
Response 3  entities = [('Angela_Merkel', '2006-01-16')]
Action 4    answer(Angela_Merkel)
Response 4  Correct!
```

Question 4: At what time did Xi Jinping first make optimistic remarks on Japan?

```
Action 0    getTime(Xi_Jinping,Make_optimistic_comment,Japan)
Response 0  entities = [('Xi_Jinping', '2008-04-18'), ...]
Action 1    getFirst({entities})
Response 1  entities = [('Xi_Jinping', '2008-04-18')]
Action 2    answer(2008-04-18)
Response 2  Correct!
```

Question 5: Who wanted to cooperate with Japan in November, 2005?

```
Action 0    getHeadEntity(Japan,Express_intent_to_cooperate,2005-11)
Response 0  entities = [('Government_Official_(Russia)', ...]
Action 1    answer(South_Korea)
Response 1  Correct!
```

Table 8: Exemplars of ARI for MULTITQ

```
f'''Please use the tool provided below to interact with the knowledge graph. You
    will find a list of actions categorized into time-based queries, entity queries,
     and specific time queries. There may be more than one answer to the question,
    but you only need to answer one correct answer that satisfies the question.

To solve this question, you need to first identify the entities and relationships in
     the question, selecting the appropriate actions to retrieve the required
    information, and finally, providing the correct answer.

Time-based Queries:
Retrieve the time of a specific event based on the head/subject entity, relation and
     tail/object entity by using the $get_time(HEAD, RELATION, TAIL)$ function, .
Identify entities/events that occurred before a given time by using the $get_before(
    ENTITY_LIST, SPECIFIED_TIME)$ function.
Identify entities/events that occurred after a given time by using the $get_after(
    ENTITY_LIST, SPECIFIED_TIME)$ function.
Identify entities/events that occurred between two specific times by using the
    $get_between(ENTITY_LIST, START_TIME, END_TIME)$ function.

Entity Queries:
Identify the tail/object entity based on the head/subject entity and relation by
    using the $get_tail_entity(CURRENT_HEAD, RELATION, OPTIONAL_TIME_CONSTRAINT)$
    function.
Identify the head/suject entity based on the tail/object entity and relation by
    using the $get_head_entity(CURRENT_TAIL, RELATION, OPTIONAL_TIME_CONSTRAINT)$
    function.

Specific Time Queries:
Pinpoint entities with the earliest occurrence by using the $get_first(ENTITY_LIST)$
     function.
Identify entities with the latest occurrence by using the $get_last(ENTITY_LIST)$
    function.
To provide your answer, use the $answer(YOUR_ANSWER)$ function.

Note: Always enclose the selected action in $ and provide a reason for your choice
    if necessary.

Examples for your reference: {examples}
(end of examples)

Current Challenge:

Question: {question}

Methodology: {methodology}
(end of methodology)


Previous Actions: {history}
(end of previous actions)


Available Actions: {actions}

Choose your next action from the available actions above, ensuring its completeness.
     If you have found the answer, remember to use the answer function.

Organize your output by strictly following the format below:

Action:
<Choose your next action from the available actions above. Note: Always enclose the
    selected action in $. Replace {your specified time} with a specified time in the
     format YYYY or YYYY-MM or YYYY-MM-DD>

Reason:
<Explain the reason for choosing this action.>'''
```

Figure 8: Prompt for the action selection.

```
f'''Carefully analyze the following correct and incorrect examples. From these,
    extract and summarize the corresponding patterns and principles. Based on these
    examples, provide a comprehensive methodology that describes how to correctly
    tackle this type of problem, highlighting the key steps and common pitfalls to
    avoid.

Task Defination: <Task Defination>
(end of Task Defination)

Here is an example output:
Example 1:
Overall methodology Instruction:
This type of problem involves the sequential determination of events, e.g. Who {
    Relation R} {entity C} before {entity B}, to find the answer {entity A} we need
    to reason in three steps, firstly to determine the specific temporal anchors, i.
    e., the occurrence time t of {entity B, Relation, and entity C}, and then to
    find out which head entities have generated a Relation R connection with {entity
     C}. Then, we find out which head entities and {entity C} have been associated
    with Relation R, and finally filter out the answers that satisfy the time
    requirement before t. The specific steps are as follows. The steps are as
    follows

Step-by-step Guide:
1. Firstly, use get_time to find the time, $get_time(entity B, Relation R, entity C)
    $, to get the quaternion {entity B, Relation R, entity C, Time t};
2. use the get_head_entity method to get the head entity, $get_head_entity(entity C,
     Relation R, entity C)$, to be able to get a list of quaternions;
3. use the get_before method to filter the entities that satisfy the constraints,
    $get_before({entities},t)$, to be able to obtain a list of entities that satisfy
     the conditions
4. complete the reasoning process by answering the found answer $answer(entity A)$

(end of example output)


Here is the correct samples and incorrect samples for the current question type:
Correct samples:
{correct_examples}

Incorrect samples:
{incorrect_examples}
(end of samples)

Now start writing. Please design a methodology that describes how to correctly
    tackle this type of problem. The goal is to provide a comprehensive guide that
    highlights the key steps and common pitfalls to avoid when approaching this type
     of problem.organize your output by strictly following the output format as
    below:

Overall Instruction:
<Define this methodology in detail.  Provide a concise guide or inference. Note that
     the guidance you provide should be at a methodological level, for this type of
    question, not for a specific one. >

Step-by-step Guide:
<A step-by-step guide or procedure detailing how to approach and solve this kind of
    question. Note that the steps proposed should be specific and relevant to this
    type of question, tell which type of action should use in each step and the
    reason>'''
```

Figure 9: Prompt for the abstract methodology instruction generation.