SAR: A Structure-Aligned Reasoning Framework for Temporal Knowledge Graph Question Answering

Qianyi Hu[123*], **Jiaxue Liu**[123*], **Xinhui Tu**[123†], **Shoujin Wang**[4]

[1]Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Central China Normal University, Wuhan, China
[2]School of Computer, Central China Normal University, Wuhan, China
[3]National Language Resources Monitor and Research Center for Network Media, Central China Normal University, Wuhan, China
[4]University of Technology Sydney, Sydney, Australia
huqianyi@mails.ccnu.edu.cn, ljx_1001@mails.ccnu.edu.cn, tuxinhui@ccnu.edu.cn, shoujin.wang@uts.edu.au

Abstract

Large language models (LLMs) augmented with retrieval have shown impressive performance in open-domain question answering, yet struggle significantly with temporal knowledge graph question answering (TKGQA). The core issue lies in structural misalignment: performing searches on structured, temporally sensitive knowledge graphs using plain-text queries often retrieves semantically similar yet structurally incorrect facts, resulting in critical inaccuracies. To address this, we introduce SAR, a Structure-Aligned Reasoning framework. SAR leverages an iterative agent-based architecture composed of three core modules: Reasoning and Answer Generation, Structure-Aligned Evidence Retrieval, and Iterative Answer Verification. The retrieval module is particularly essential; it employs structured query decomposition, embedding-based semantic matching, and chronological reranking to retrieve temporally consistent and schema-aligned knowledge facts from temporal knowledge graphs (TKGs). These precisely retrieved facts guide the LLM-based reasoning agent through iterative reasoning cycles, significantly reducing hallucinations and ensuring accuracy. A final verification stage ensures that proposed answers strictly adhere to requirements, reinforcing accuracy and coherence. Extensive experiments conducted on two benchmark datasets, MultiTQ and CronQuestions, demonstrate the effectiveness of SAR. Specifically, utilizing GPT-4.1, SAR achieves a Hits@1 score of 78.2% on MultiTQ, significantly surpassing existing methods, and similarly establishes new performance benchmarks on CronQuestions. Our findings highlight the crucial importance of structural alignment in temporal reasoning, particularly for complex queries involving multiple temporal constraints and multi-hop reasoning.

Extended version — https://github.com/Huqianyi/SAR

Introduction

TKGQA is the task of answering questions using facts stored in TKGs, where each fact is a time-stamped quadruple $(\textit{head entity}, \textit{relation}, \textit{tail entity}, \textit{time})$ (Jia et al. 2018).

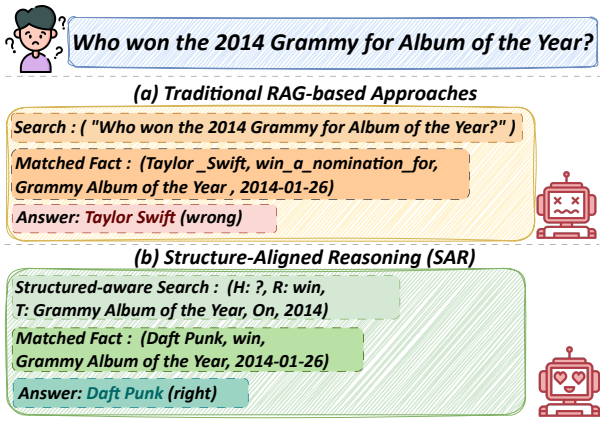

Figure 1: This figure compares SAR to traditional RAG-based approaches, showing how explicitly aligning queries with the head-relation-tail-time schema of temporal knowledge graphs prevents incorrect associations and accurately identifies factual answer.

In contrast to standard knowledge graph question answering (KGQA), TKGQA must handle temporal constraints in questions, such as "Who became CEO of Company X after 2015?" Answering such questions requires not only finding relevant entities and relations but also reasoning about specific time points or intervals (Saxena, Chakrabarti, and Talukdar 2021; Shang et al. 2022). This introduces unique challenges, as the question answering (QA) system needs to discern when certain facts hold true, filtering or comparing knowledge based on timestamps (Ding et al. 2024; Wang et al. 2024; Su et al. 2024).

Recent TKGQA approaches integrate LLMs with retrieval from TKGs, significantly advancing the state of the art by enhancing temporal reasoning with external knowledge (Zhu et al. 2023; Qian et al. 2024; Hu et al. 2025). Nevertheless, existing frameworks often treat knowledge graph (KG) facts as unstructured text, causing structural misalignment between queries and the TKG schema (Ding et al.

*These authors contributed equally.
†Corresponding authors.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2024; Wang et al. 2024; Yue 2025). Empirically, we observe that employing free-text searches within structured TKGs tends to yield facts that are semantically relevant but structurally inconsistent, especially when queries involve multiple entities or stringent temporal constraints (Chen, Liao, and Zhao 2023; Chen et al. 2024a). Consequently, even sophisticated LLM agents can retrieve temporally incorrect facts or misinterpret relations, leading to hallucinations or violations of query conditions (Arslan et al. 2024; Zhao et al. 2024b). This structural misalignment occurs both at the retrieval phase—where unstructured queries fail to respect the subject–predicate–object structure of the TKG—and during the reasoning phase, where the LLM's chain-of-thought process may inadequately enforce strict temporal logic (Gao et al. 2024; Qian et al. 2024; Luo et al. 2024). Addressing these structural misalignments is critical for robust and accurate TKGQA, especially for complex queries requiring multi-hop or implicit temporal reasoning (Li et al. 2024b; Sanmartin 2024).

To tackle the structural misalignment prevalent in TKGQA, we propose SAR, a novel structure-aligned reasoning framework explicitly designed to integrate LLM reasoning processes with structured TKGs. Central to SAR is the Structure-Aligned Evidence Retrieval module, which significantly enhances the retrieval process by maintaining strict schema alignment and temporal accuracy. As illustrated in Figure 1, traditional RAG-based approaches can mistakenly retrieve structurally incorrect facts, resulting in erroneous answers—such as retrieving nomination information when the question asks for the actual winner of the 2014 Grammy for Album of the Year. In contrast, SAR employs structured query decomposition coupled with embedding-based retrieval techniques, as demonstrated by accurately identifying Daft Punk as the correct award recipient. This methodology enables precise semantic matching between query components and candidate facts, effectively reducing retrieval of irrelevant or contextually mismatched information.

SAR further employs an iterative loop involving a reasoning agent and a validation module. The agent systematically formulates responses by decomposing queries into structured sub-queries, ensuring that retrieved evidence adheres closely to the TKG schema. Subsequently, the validation module evaluates responses for accuracy, completeness, and temporal consistency, initiating further refinement iterations when necessary. This feedback-driven iterative refinement ensures that all reasoning steps are consistently supported by accurate, temporally coherent facts, significantly improving the overall reliability and structural integrity of TKGQA outcomes. In summary, our contributions are as follows:

- We introduce SAR, an LLM-driven TKGQA framework that maintains alignment with the KG's structure at every step. By decomposing queries into subject–relation–object–time components and retrieving facts in a schema-consistent way, SAR tightly couples LLM reasoning with the TKG's inherent structure.
- SAR consistently outperforms existing methods by a substantial margin on two challenging TKGQA benchmarks. These results underscore the importance of structural alignment in temporal reasoning tasks and demonstrate that SAR effectively alleviates the key challenges faced by previous TKGQA systems.

Related Work

TKGQA research follows two main paradigms: embedding-based and semantic parsing approaches (Liu, Feng, and Huang 2025; Su et al. 2024). Recent LLM integration has significantly advanced TKGQA frameworks (Chen et al. 2024c; Gao et al. 2024; Hu et al. 2025). We discuss representative systems in each category below.

Embedding-based Approaches. Embedding methods encode entities, relations, and timestamps into vector spaces to capture temporal information implicitly. EXAQT (Jia et al. 2021) employed a two-stage process with BERT-enhanced search and time-aware GNNs for complex temporal queries. CronKGQA (Saxena, Chakrabarti, and Talukdar 2021) utilized pre-trained temporal embeddings and transformer-based encoders, achieving significant accuracy improvements on CronQuestions. TempoQR (Mavromatis et al. 2022) incorporated question-specific temporal contexts directly into embeddings, while MultiQA (Chen, Liao, and Zhao 2023) handled mixed temporal granularities by aggregating information at different scales (day, month, year). However, embedding methods struggle with explicit logical constraints and interpretability due to latent vector computations (Su et al. 2024; Ding et al. 2024).

Semantic Parsing and Query-Based Approaches. Semantic parsing methods translate natural language questions into structured queries executable against KGs, emphasizing interpretability and precise logical reasoning. TEQUILA (Jia et al. 2018) decomposed questions into sub-queries with distinct temporal constraints, establishing a modular parsing approach. SYGMA (Neelam et al. 2021) presented modular reasoning adaptable across knowledge bases. Recent neural-guided frameworks include SF-TQA (Ding et al. 2022), which integrated symbolic temporal logic into query generation, and Prog-TQA (Chen et al. 2024c), which synthesized logical queries via iterative refinement for improved multi-hop temporal reasoning. These methods can be brittle due to parsing errors or rigid grammars.

LLM-Augmented and Agent-Based Approaches. Recent LLM-based approaches have integrated retrieval processes with agent paradigms (Wang et al. 2024; Zhao et al. 2024a). TimeR4 (Qian et al. 2024) implemented a Retrieve-Rewrite-Retrieve-Rerank mechanism, reformulating implicit temporal queries into explicit formats and reranking retrieved facts with LLM guidance to mitigate temporal hallucinations. TempAgent (Hu et al. 2025) applied the ReAct paradigm with explicit temporal constraints, achieving substantial accuracy gains by filtering irrelevant knowledge. The ARI framework (Chen et al. 2024b) divided temporal reasoning into knowledge-agnostic and knowledge-based phases, training LLMs through self-reflection on reasoning examples.

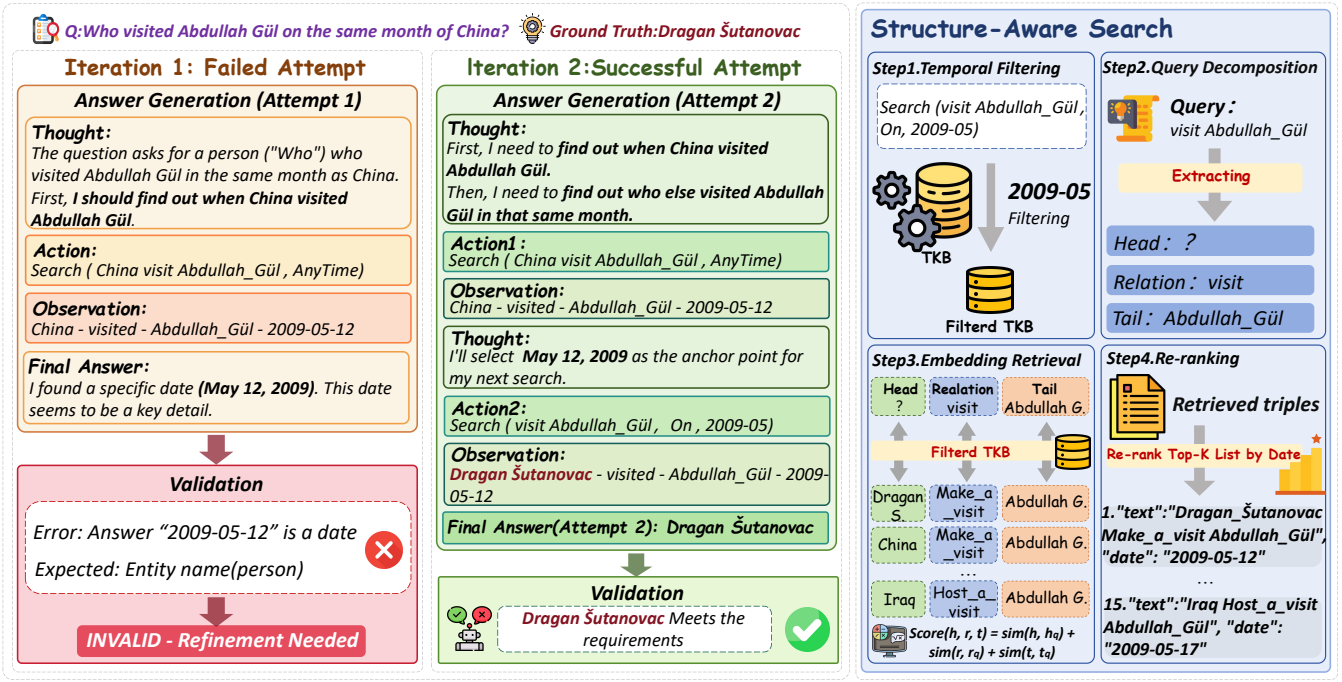

Figure 2: Workflow of the proposed SAR framework.

However, these approaches frequently retrieve KG facts as unstructured text, causing structural misalignments and erroneous reasoning (Ding et al. 2024; Du, Li, and Li 2024). In contrast, SAR explicitly preserves KG quadruple structures throughout retrieval and reasoning, ensuring precise schema alignment and robust temporal logic enforcement for superior performance on complex temporal reasoning tasks.

Preliminaries

TKGQA. A TKG is formally defined as a collection of timestamped facts represented by quadruples, denoted as $\mathcal{G} = (h, r, t, \tau) \mid h, t \in \mathcal{E}, r \in \mathcal{R}, \tau \in \mathcal{T}$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T}$ is the set of timestamps. Each quadruple (h, r, t, τ) encodes the fact that a head entity h is connected to a tail entity t through a relation r at a specific timestamp τ. Given a query q, the objective of TKGQA is to derive an accurate answer a by retrieving and reasoning over the relevant temporal facts within $\mathcal{G}$.

Methodology

The core insight motivating our approach is that effective TKGQA hinges on maintaining structural alignment between the reasoning process and the underlying knowledge graph schema, while ensuring that each reasoning step is firmly grounded in verifiable temporal facts. To operationalize this principle, we propose **SAR**, an agent-based reasoning framework explicitly designed to align LLM reasoning processes with structured TKGs. In this section, we first provide a high-level overview of SAR's architecture, followed by detailed descriptions of its core components: the **Reasoning and Answer Generation** module, the **Structure-Aligned Evidence Retrieval** module, and the **Iterative Answer Verification** module. The comprehensive workflow of SAR is formally presented in Algorithm 1 and visually depicted in Figure 2.

Overview

At a high level, SAR models TKGQA as an interactive process involving two key components: a reasoning agent and a validation module. The reasoning agent performs initial reasoning over the input question, queries the TKG to retrieve relevant evidence, and formulates an initial answer. The validation module then evaluates this answer for accuracy and completeness, triggering additional iterations of reasoning and retrieval if the initial response is insufficient. This iterative interaction ensures the LLM's free-form generation remains consistently grounded in factual, temporally annotated KG quadruples, thereby preserving structural alignment between the reasoning process and the underlying knowledge schema.

The reasoning agent operates within an iterative loop, limited to a maximum of N attempts. In each iteration, the agent produces an answer candidate through a structured reasoning process enhanced by evidence retrieved from the KG. Following each generation step, the validation module—implemented as an LLM prompt acting as a critic—evaluates the candidate's correctness. If the candidate satisfies verification criteria, specifically matching the expected answer type, fulfilling all temporal constraints, and being thoroughly supported by retrieved facts, the iteration

terminates. Otherwise, the agent refines its approach based on the provided feedback and attempts another iteration. This iterative refinement mechanism is formally detailed in lines 1–10 of Algorithm 1. By integrating feedback-driven iterations, SAR effectively rectifies intermediate reasoning errors, such as overlooked facts or misinterpreted temporal constraints, before finalizing the output. Empirically, we observe that most queries converge after a single refinement iteration, demonstrating the precision and effectiveness of our verification approach.

Reasoning and Answer Generation

We employ a ReAct-inspired LLM agent (Yao et al. 2022) to systematically generate answers through structured and iterative reasoning steps (Algorithm 1, lines 11–23). The agent maintains an internal dialogue memory composed of thought–action–observation triples. At each reasoning step, it formulates a reasoning trace (*thought*) guided by both the input question and the current state of its internal memory. Subsequently, the agent selects an appropriate *action*: either performing a SEARCH—by formulating a sub-query with explicit temporal constraints to retrieve relevant facts from the temporal knowledge graph—or initiating a FINISH operation to conclude reasoning and produce the final answer. When a SEARCH action is executed, the resulting retrieved facts (*observation*) are integrated into the agent's memory, thereby enriching the context for subsequent reasoning iterations. This iterative ReAct reasoning cycle continues, progressively refining the agent's internal understanding and aligning closely with factual evidence, until a FINISH action is triggered.

By decomposing the reasoning process into incremental thought steps and structurally aligned sub-queries, the LLM agent remains consistently attuned to the schema of the underlying knowledge graph and firmly grounds each inference step in verifiable facts. This structured, iterative strategy naturally addresses complex multi-hop questions, allowing the agent to progressively focus on simpler subproblems, retrieve pertinent evidence, and iteratively update its internal memory. As a result, our ReAct-based approach significantly mitigates hallucinations and enhances factual correctness throughout the answer generation process.

Structure-Aligned Evidence Retrieval

The SEARCH function retrieves semantically relevant and temporally consistent facts from TKG $\mathcal{G}$ for the reasoning agent. Unlike traditional keyword-based retrieval methods, our approach explicitly preserves the structured representations of both the query and the retrieved facts, rather than flattening them into unstructured textual forms. By maintaining this structural alignment with the knowledge graph schema, our method effectively avoids irrelevant matches and preserves explicit relational and temporal links between entities.

The evidence retrieval process (Algorithm 1, lines 24–34) involves four main steps. To provide clarity, we illustrate these steps with a running example query: *"Who visited Abdullah Gül on the same month of China?"* (Figure 2)

Algorithm 1: The Overall Pipeline

Input: Query q, KG $\mathcal{G}$
Parameter: Max verification iteration count N
Output: Final answer a to q

```
1: a ← ε
2: i ← 0
3: while i ≤ N do
4:     a ← REASONING(q)
5:     if LLM(P_verify, a) is Valid then
6:         break
7:     end if
8:     i ← i +1
9: end while
10: return a
11: function REASONING(q)
12:     Memory ← [P_reasoning, q]
13:     loop
14:         (Thought, Action) ← LLM(Memory)
15:         if Action is SEARCH then
16:             (sub_query, type, time) ← PARSE(Action)
17:             Fact ← SEARCH(sub_query, type, time)
18:             Memory.append(Thought, Action, Fact)
19:         else if Action is FINISH then
20:             return answer
21:         end if
22:     end loop
23: end function
24: function SEARCH(sub_query, type, time)
25:     if type is AnyTime then
26:         G' ← G
27:     else
28:         G' ← FILTER(G, type, time)
29:     end if
30:     (h, r, t) ← LLM(P_decomp, sub_query)
31:     F ← TOP-K(G', h, r, t)
32:     F' ← RE-ORDER(F)
33:     return F'
34: end function
```

Temporal Filtering. When the Answer Generation agent specifies temporal constraints, we first narrow the knowledge graph to a relevant subset. Formally, given a temporal constraint type $type \in \{\text{ANYTIME}, \text{BEFORE}, \text{AFTER}, \text{IN}\}$ and an associated timestamp $time$, we derive a filtered subgraph $\mathcal{G}' = \text{FILTER}(\mathcal{G}, type, time)$ containing only facts whose timestamps satisfy the specified temporal condition. For instance, if $type = \text{BEFORE}$ and $time = 2010$, the subgraph $\mathcal{G}'$ includes only quadruples (h, r, t, τ) where $\tau < 2010$. In cases where no explicit temporal filter is specified (i.e., $type = \text{ANYTIME}$), the original knowledge graph is preserved: $\mathcal{G}' = \mathcal{G}$. In our running example, *"Who visited Abdullah Gül on the same month of China?"*, the agent imposes no temporal constraints (i.e., $type = \text{ANYTIME}$), and thus the full knowledge graph $\mathcal{G}$ remains accessible for search.

Structured Query Decomposition. The sub-query passed to the search module typically originates as a natural-

language snippet. We transform this sub-query into a structured triple representation (h, r, t) by leveraging an LLM-based parsing step. Specifically, the triple is defined as: h, the head entity; r, the relation predicate; and t, the tail entity, each interpreted directly from the sub-query. For example, the sub-query *"China visit Abdullah Gül date"* would be parsed into the structured triple $(\textit{China}, \textit{visit}, \textit{Abdullah Gül})$, indicating that the agent is retrieving facts matching the pattern *"China visited Abdullah Gül (at some timestamp)."* This structured representation directly mirrors the quadruple form $(head, relation, tail, timestamp)$ employed in the temporal knowledge graph, thereby facilitating precise and schema-aligned matching in subsequent retrieval steps.

Embedding-Based Retrieval. Given a structured query triple (h, r, t) and a temporally filtered subgraph $\mathcal{G}'$, candidate facts are retrieved through semantic matching rather than exact keyword matching. Entities and relations within the knowledge graph are embedded into a continuous vector space using SBERT (Reimers and Gurevych 2019), enabling similarity-based comparisons between query components and candidate facts. For each fact (h', r', t', τ') in $\mathcal{G}'$, we compute a relevance score as the aggregated similarity across subject, predicate, and object components:

$$\mathrm{S}\big((h, r, t) \mid (h', r', t')\big) = S(h, h') + S(r, r') + S(t, t'), \quad (1)$$

where $S(x, y)$ denotes the cosine similarity or dot product between embedding vectors of x and y. This scoring strategy ensures candidate facts are highly ranked only if they exhibit strong alignment across all query components. We select the Top-K candidate facts (e.g., $K = 15$, tuned via validation data for an optimal recall-precision balance) to form the candidate set F.

In our running example, given the query triple $(\textit{China}, \textit{visit}, \textit{Abdullah Gül})$, the retrieval module searches for facts whose subject aligns closely with China, predicate with visit, and object with "Abdullah Gül". Relevant facts such as $(\textit{China}, \textit{visit}, \textit{Abdullah Gül}, 2009)$ and $(\textit{China}, \textit{visit}, \textit{Abdullah Gül}, 2012)$ will achieve high scores due to strong component-wise matches. Conversely, facts sharing partial overlap but differing significantly in relation or context receive lower scores, thereby reducing spurious retrievals. This embedding-based, schema-aligned retrieval effectively filters out irrelevant or contextually mismatched facts, ensuring accurate and reliable evidence retrieval.

Chronological Re-ranking. The initial candidate set $\mathcal{F}$, composed of the Top-K retrieved facts, typically lacks inherent temporal ordering. When the query or context indicates a temporal sequence—such as explicitly requesting the "first","next", or chronologically ordered events—we impose chronological order on the retrieved facts. Specifically, we derive an ordered list $\mathcal{F}' = \textsc{Reorder}(\mathcal{F})$ wherein each fact's timestamp precedes or equals the subsequent one. Formally, for any two consecutive facts f_i and f_{i+1} in $\mathcal{F}'$ with timestamps τ_i and τ_{i+1} respectively, it holds that $\tau_i \leq \tau_{i+1}$.

In our running example, if the retrieved set $\mathcal{F}$ includes the facts $(\textit{China}, \textit{visit}, \textit{Abdullah Gül}, 2012)$ and $(\textit{China}, \textit{visit}, \textit{Abdullah Gül}, 2009)$, chronological re-ranking arranges them in ascending temporal order as $[2009, 2012]$. This approach ensures that the evidence provided to the agent follows a coherent temporal sequence, aligning naturally with queries expecting temporally ordered responses. Moreover, presenting facts in chronological order enhances the logical consistency of the information, enabling the reasoning agent to construct coherent narratives and more effectively perform temporal reasoning during answer generation.

Through the steps outlined above, the SEARCH module supplies the reasoning agent with a concise, structurally aligned, and temporally coherent set of knowledge facts. Because our retrieval mechanism explicitly respects temporal constraints and maintains alignment with the knowledge graph schema, the reasoning agent is freed from the burden of filtering extraneous or structurally misaligned information. Consequently, the agent can concentrate on synthesizing accurate and consistent answers from a focused evidence set that inherently matches the query's structural and temporal requirements.

Iterative Answer Verification

Once the reasoning agent generates a candidate answer a, SAR employs an LLM-based validation step to rigorously verify its correctness. Specifically, we implement this validation via an LLM using a specialized prompt for structured evaluation. The LLM first predicts the expected answer type, confirms that the response directly and concretely addresses the question, verifies precise type alignment, and ensures all criteria are satisfied.

The verifier returns a verdict: VALID if all criteria are met, or INVALID if any criteria fail. A VALID verdict concludes validation. An INVALID verdict triggers a refinement loop, incrementing an attempt counter and prompting iterative response refinement based on explicit feedback. This iterative process continues until a valid response is confirmed or a predefined maximum of N attempts is reached, ensuring enhanced structural accuracy, temporal precision, and overall reliability of SAR-generated answers.

Experiment

Experiment Settings

Datasets. We evaluate SAR on two established TKGQA benchmarks to comprehensively assess its effectiveness and generalizability. For our primary evaluation, we utilize the **MultiTQ** dataset (Chen, Liao, and Zhao 2023), which encompasses diverse temporal reasoning challenges at multiple granularities. Specifically, we construct a focused test set consisting of 560 questions, randomly sampled to represent the original MultiTQ data. These questions are evenly divided into two categories: those with a single temporal constraint (*Single*) and those requiring reasoning over multiple temporal conditions (*Multiple*), enabling a rigorous examination of SAR's ability to handle complex temporal logic. Additionally, we perform evaluations on the widely adopted **CronQuestions** benchmark (Saxena, Chakrabarti, and Talukdar 2021) to further demonstrate our method's robustness across different TKGQA scenarios.

Baselines. We evaluate SAR against several strong baselines. **MultiQA** (Chen, Liao, and Zhao 2023) is the offi-

Method	Base LLM	MultiTQ					CronQuestions				
		Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
			Single	Multiple	Entity	Time		Simple	Complex	Entity	Time
MultiQA	–	28.9	36.1	11.2	32.9	20.3	–	–	–	–	–
ARI	GPT-3.5	38.0	68.0	21.0	39.4	34.4	70.7	86.0	57.0	66.0	80.0
Naive RAG	Llama3	32.3	40.1	12.4	18.0	62.7	53.3	62.1	20.0	53.7	52.6
	GPT-3.5	32.0	40.4	11.2	20.1	57.6	49.2	58.9	20.0	52.4	47.4
	GPT-4.1	40.9	49.6	19.3	28.2	68.4	68.3	73.7	48.0	67.1	71.1
ReAct RAG	Llama3	33.2	40.6	14.9	24.8	51.4	72.5	78.9	48.0	68.3	81.6
	GPT-3.5	33.0	41.1	13.0	20.2	61.0	69.2	75.8	44.0	67.1	73.7
	GPT-4.1	43.4	54.1	12.4	19.3	**91.5**	83.3	88.4	64.0	80.5	89.5
TempAgent	Llama3	54.3	69.7	16.2	48.3	67.2	80.0	85.3	60.0	76.8	89.5
	GPT-3.5	53.9	68.4	16.8	47.8	66.1	76.7	82.1	56.0	73.2	84.2
	GPT-4.1	60.7	73.4	28.6	54.0	75.1	87.5	93.7	64.0	85.4	92.1
SAR (Ours)	Llama3	65.0	81.5	31.5	56.9	87.0	83.3	87.4	68.0	81.7	86.8
	GPT-3.5	57.5	72.9	18.0	49.6	73.5	80.0	88.9	64.0	78.0	84.2
	GPT-4.1	**78.2**	**90.0**	**50.0**	**72.1**	**91.5**	**91.7**	**95.8**	**76.0**	**90.2**	**94.7**

Table 1: Performance comparison (Hits@1, %) of SAR against baselines on MultiTQ and CronQuestions datasets. The **best** results are highlighted in bold, and the second-best results are underlined.

cial embedding-based baseline provided with MultiTQ. **ARI** (Chen et al. 2024b) integrates temporal reasoning strategies into LLMs specifically for TKGQA tasks. To assess the effectiveness of interactive reasoning, we implement two retrieval-augmented generation (RAG) variants: a standard **Naive RAG** pipeline (Chen et al. 2024a), which directly generates answers from retrieved KG text, and a **ReAct RAG** agent (Yao et al. 2022) that utilizes iterative reasoning steps. Lastly, we include **TempAgent** (Hu et al. 2025), an agent-based TKGQA framework, as a competitive reference. All methods are evaluated under identical conditions on our test sets.

Evaluation Metric. Following established practices in QA evaluation (Sun et al. 2024; Li et al. 2024a), we adopt Hits@1 as our primary metric, where a prediction is considered correct if it exactly matches any ground-truth answer (Qian et al. 2024; Li et al. 2024a; Sun et al. 2024; Jiang et al. 2023; Liu et al. 2024).

Backbone LLMs. To isolate our framework's contribution from the LLM's capabilities, we evaluate SAR across multiple backbones, including OpenAI's `GPT-3.5-Turbo` and `GPT-4.1`, as well as the open-source, 70B-parameter `Llama3-Instruct`. All agent-based models are evaluated using these LLM backbones to ensure a comprehensive comparison.

Overall Performance

Table 1 presents a performance comparison between SAR and baseline methods on the MultiTQ and CronQuestions datasets. Using GPT-4.1 as the backbone, SAR achieves superior results, reaching Hits@1 scores of 78.2% on MultiTQ and 91.7% on CronQuestions. This performance significantly exceeds that of the strongest baseline, TempAgent (also utilizing GPT-4.1), which attains 60.7% on MultiTQ and 87.5% on CronQuestions. The improvement of more than 17 percentage points on MultiTQ clearly demonstrates the effectiveness of our structure-aligned reasoning approach for accurate answer extraction from TKGs.

SAR consistently demonstrates significant performance gains across various backbone LLMs. For instance, with the open-source Llama-3 (70B) model, SAR achieves 65.0% Hits@1 on MultiTQ, surpassing other frameworks like TempAgent (54.3%) using the same backbone. This indicates that SAR's structured reasoning approach effectively compensates for the limitations of less powerful LLMs by providing more precise structural guidance.

Performance by Question Complexity and Answer Type

To gain deeper insights into the results, we further analyze performance by breaking it down according to question complexity and answer type on each dataset. The detailed results of this analysis are shown in Table 1.

Single vs. Multiple Time Constraints (MultiTQ). Questions with a single temporal constraint are generally easier for all models than those with multiple time conditions. Our SAR (GPT-4.1) achieves an impressive 90.0% Hits@1 on single-constraint questions, compared to the strongest GPT-4.1 baseline, TempAgent (73.4%), and the embedding baseline (36.1%). More importantly, for questions with multiple temporal constraints – the most challenging category – SAR significantly outperforms other methods, achieving 50.0% accuracy where the best baseline (TempAgent with GPT-4.1) reaches only 28.6%. This gap of over 21 points illustrates SAR's strength in complex temporal reasoning. The structure alignment in our approach helps navigate multiple time-dependent facts more effectively than baselines that either retrieve text naively or do not enforce graph-structured reasoning.

Simple vs. Complex Questions (CronQuestions). A similar pattern is observed on CronQuestions. For the "sim-

Model	Overall	Question Type		Answer Type	
		Single	Multiple	Entity	Time
SAR	78.2	90.0	50.0	72.1	91.5
w/o Decompose	71.1	84.5	37.9	61.1	92.7
w/o Validation	77.0	89.0	47.2	70.8	90.4

Table 2: Ablation study evaluating the impact of structured query decomposition and iterative validation on SAR's performance (Hits@1, %) on MultiTQ.

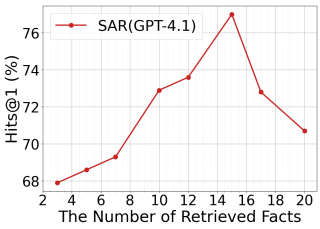

Figure 3: SAR's performance (Hits@1) on MultiTQ with varying retrieval sizes (K).

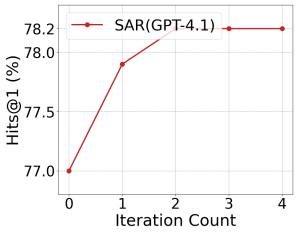

Figure 4: SAR's performance (Hits@1) on MultiTQ with varying verification iteration counts (N).

ple" temporal questions, all methods perform well; SAR hits 95.8% and strong baselines also exceed 85%. However, for complex questions requiring multi-hop reasoning or multiple temporal comparisons, our model excels. SAR with GPT-4.1 achieves 76.0% on complex questions, substantially higher than TempAgent (64.0%) or ReAct (64.0%) under the same GPT-4.1 backbone. In fact, SAR with GPT-4.1 outperforms ARI (57.0% on complex questions) by a large margin. This demonstrates that SAR's design – decomposing queries and verifying answers – is particularly effective for complex temporal queries that stymie other approaches.

Entity Answers vs. Time Answers. We also categorize questions by the type of answer expected: an entity versus a timestamp. Our model shows strong performance on both categories. In MultiTQ, SAR with GPT-4.1 achieves 72.1% on entity answers and 91.5% on time answers, whereas the best baseline (TempAgent with GPT-4.1) managed 54.0% and 75.1% respectively. The improvement on temporal (Time) answers is especially pronounced; we attribute this to SAR's ability to precisely align and filter by timestamps during retrieval. On CronQuestions, SAR maintains over 90% accuracy on both, slightly higher for time answers. Overall, these breakdowns confirm that SAR is not trading off one capability for another – it improves performance for all question types and answer types, with particular benefit on the more challenging temporal reasoning cases.

Ablation Studies

We conduct ablation studies on the MultiTQ test set to quantify the contribution of SAR's core components, with results shown in Table 2.

Impact of Structured Query Decomposition. Removing the structured query decomposition module (w/o decompose) causes the most significant performance degradation, with Hits@1 dropping by 7.1 points overall (78.2% to 71.1%) and over 12 points on complex queries. This confirms our central hypothesis: enforcing structural alignment by parsing queries into structured triples is the primary driver of SAR's accuracy, as it prevents the LLM from being confused by irrelevant facts.

Impact of Iterative Answer Verification. Disabling the iterative verification module (w/o verification) results in a smaller but notable accuracy drop. This component's primary role is to ensure the solidity and reliability of the final answer. It acts as a crucial final check to correct edge-case errors. While the overall accuracy gain is modest, this verification step is essential for increasing the trustworthiness of the output with minimal computational overhead.

Hyper-parameter Studies

Effect of Number of Retrieved Facts (K). The parameter K controls the number of top-ranked candidate quadruples retrieved for each query. As illustrated in Figure 3, our experiments indicate that the performance is sensitive to the choice of K. Selecting too small a value of K risks omitting critical facts necessary for accurate reasoning, while excessively large values introduce noise, potentially confusing the LLM. Empirically, increasing K from 5 to 15 notably improves the overall Hits@1, from approximately 74% to 78%. However, further increasing K to 30 provides no additional advantage and slightly degrades performance, especially on complex questions involving multiple constraints. Therefore, we set $K = 15$ by default, effectively balancing retrieval recall and precision.

Effect of Verification Iterations (N). The verification step in our framework enables the agent to iteratively refine answers when initial predictions are deemed unsatisfactory. By default, we permit at most two verification iterations, allowing the model to revise its initial response twice. Empirically, these two iterations effectively resolve the majority of initial inaccuracies. Additional verification iterations yield diminishing returns; for instance, the third and fourth iterations contribute virtually no performance improvement, indicating that nearly all gains are realized within the first two iterations. Figure 4 illustrates this diminishing returns phenomenon, supporting our choice to set the iteration count at $N = 2$ to balance effectiveness and computational efficiency.

Conclusion

In this work, we address structural misalignment in TKGQA by introducing SAR, a novel framework that maintains schema alignment through structured query decomposition, structure-aligned retrieval, and iterative answer verification. Experiments on two widely-used benchmarks demonstrate substantial accuracy improvements, particularly for complex temporal queries, highlighting the importance of preserving structural consistency with the underlying knowledge graph. Future research will focus on enhancing SAR's efficiency and expanding its applicability to more intricate temporal scenarios, such as event durations and recurring events.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62472192). Authors are grateful to the anonymous reviewers for helpful comments.

References

Arslan, M.; Ghanem, H.; Munawar, S.; and Cruz, C. 2024. A Survey on RAG with LLMs. *Procedia computer science*, 246: 3781–3790.

Chen, J.; Lin, H.; Han, X.; and Sun, L. 2024a. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17754–17762.

Chen, Z.; Li, D.; Zhao, X.; Hu, B.; and Zhang, M. 2024b. Temporal Knowledge Question Answering via Abstract Reasoning Induction. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4872–4889. Bangkok, Thailand: Association for Computational Linguistics.

Chen, Z.; Liao, J.; and Zhao, X. 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)*, 11378–11392.

Chen, Z.; Zhang, Z.; Li, Z.; Wang, F.; Zeng, Y.; Jin, X.; and Xu, Y. 2024c. Self-Improvement Programming for Temporal Knowledge Graph Question Answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 14579–14594.

Ding, W.; Chen, H.; Li, H.; and Qu, Y. 2022. Semantic Framework based Query Generation for Temporal Question Answering over Knowledge Graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1867–1877.

Ding, Y.; Fan, W.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; and Li, Q. 2024. A Survey on RAG Meets LLMs: Towards Retrieval-Augmented Large Language Models. *CoRR*, abs/2405.06211.

Du, C.; Li, X.; and Li, Z. 2024. Semantic-Enhanced Reasoning Question Answering over Temporal Knowledge Graphs. *Journal of Intelligent Information Systems*, 1–23.

Gao, Y.; Qiao, L.; Kan, Z.; Wen, Z.; He, Y.; and Li, D. 2024. Two-stage Generative Question Answering on Temporal Knowledge Graph Using Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 6719–6734. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.

Hu, Q.; Tu, X.; Guo, C.; and Zhang, S. 2025. Time-aware ReAct Agent for Temporal Knowledge Graph Question Answering. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Findings of the Association for Computational Linguistics: NAACL 2025*, 6013–6024. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-195-7.

Jia, Z.; Abujabal, A.; Saha Roy, R.; Strötgen, J.; and Weikum, G. 2018. Tequila: Temporal Question Answering over Knowledge Bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 1807–1810.

Jia, Z.; Pramanik, S.; Saha Roy, R.; and Weikum, G. 2021. Complex Temporal Question Answering on Knowledge Graphs. In *Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management*, CIKM '21, 792–802. ACM.

Jiang, J.; Zhou, K.; Dong, Z.; Ye, K.; Zhao, W. X.; and Wen, J.-R. 2023. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 9237–9251.

Li, X.; Zhao, R.; Chia, Y. K.; Ding, B.; Joty, S.; Poria, S.; and Bing, L. 2024a. Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Li, Z.; Chen, X.; Yu, H.; Lin, H.; Lu, Y.; Tang, Q.; Huang, F.; Han, X.; Sun, L.; and Li, Y. 2024b. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *arXiv preprint arXiv:2410.08815*.

Liu, J.; Tian, X.; Tong, H.; Xie, C.; Ruan, T.; Cong, L.; Wu, B.; and Wang, H. 2024. Enhancing Chinese abbreviation prediction with LLM generation and contrastive evaluation. *Information Processing & Management*, 61(4): 103768.

Liu, Q.; Feng, S.; and Huang, M. 2025. TEQA: Temporal knowledge graph enhanced question answering. *Knowledge-Based Systems*, 113916.

Luo, L.; Li, Y.-F.; Haf, R.; and Pan, S. 2024. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Mavromatis, C.; Subramanyam, P. L.; Ioannidis, V. N.; Adeshina, A.; Howard, P. R.; Grinberg, T.; Hakim, N.; and Karypis, G. 2022. TempoQR: Temporal Question Reasoning over Knowledge Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 5825–5833.

Neelam, S.; Sharma, U.; Karanam, H.; Ikbal, S.; Kapanipathi, P.; Abdelaziz, I.; Mihindukulasooriya, N.; Lee, Y.-S.; Srivastava, S.; Pendus, C.; Dana, S.; Garg, D.; Fokoue, A.; Bhargav, G. P. S.; Khandelwal, D.; Ravishankar, S.; Gurajada, S.; Chang, M.; Uceda-Sosa, R.; Roukos, S.; Gray, A.; Riegel, G. L.; Luus, F.; and Subramaniam, L. V. 2021. SYGMA: System for Generalizable Modular Question Answering OverKnowledge Bases. arXiv:2109.13430.

Qian, X.; Zhang, Y.; Zhao, Y.; Zhou, B.; Sui, X.; Zhang, L.; and Song, K. 2024. TimeR4: Time-aware retrieval-augmented large language models for temporal knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6942–6952.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sanmartin, D. 2024. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035*.

Saxena, A.; Chakrabarti, S.; and Talukdar, P. 2021. Question Answering over Temporal Knowledge Graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Shang, C.; Wang, G.; Qi, P.; and Huang, J. 2022. Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8017–8026.

Su, M.; Li, Z.; Chen, Z.; Bai, L.; Jin, X.; and Guo, J. 2024. Temporal knowledge graph question answering: A survey. *arXiv preprint arXiv:2406.14191*.

Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.; Shum, H.-Y.; and Guo, J. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024. A Survey on Large Language Model Based Autonomous Agents. *Frontiers of Computer Science*, 18(6): 1–26.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

Yue, M. 2025. A survey of large language model agents for question answering. *arXiv preprint arXiv:2503.19213*.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024a. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhao, S.; Yang, Y.; Wang, Z.; He, Z.; Qiu, L. K.; and Qiu, L. 2024b. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Zhu, Y.; Wang, X.; Chen, J.; Qiao, S.; Ou, Y.; Yao, Y.; Deng, S.; Chen, H.; and Zhang, N. 2023. LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities. *World Wide Web*.