ConTExT: Temporal Knowledge Graph Question Answering via Contractive Dual-Space Subgraph Search

Anonymous ACL submission

Abstract

Temporal Knowledge Graph Question Answering (TKGQA) commonly relies on retrieval-aggregation-reasoning pipelines, which suffer from two critical challenges: *error propagation*, where early retrieval mistakes accumulate along reasoning paths, and *temporal conflicts* caused by post-hoc temporal filtering, leaving temporally inconsistent paths in the search space and undermining multi-hop reasoning reliability. To address these limitations, we propose **ConTExT** (Contractive Temporal Exploration), a contractive subgraph search framework over structural and temporal spaces for TKGQA. It dynamically anchors question-relevant entities, performs selective subgraph exploration guided by path semantics and sub-question consistency, and applies Temporal Feasible Region Propagation (TFRP), which actively propagates temporal constraints during search and maintains a temporal feasible region. Experiments on MultiTQ demonstrate that ConTExT consistently outperforms prior state-of-the-art methods, with particularly strong gains on multi-hop questions, where it achieves an approximately 27.1% improvement. Our code is available at `https://anonymous.4open.science/r/ConTExT-D8B0/`.

1 Introduction

Temporal Knowledge Graph Question Answering (TKGQA) aims to answer natural language questions by reasoning over time-aware facts in Temporal Knowledge Graphs (TKGs). Compared with static knowledge graph question answering, it requires identifying relevant entities and relations while satisfying temporal operators such as *before*, *after*, *first*, and *last*, making it particularly challenging under implicit event references, multi-hop dependencies, and multiple temporal constraints.

Recent studies have advanced TKGQA through multi-granularity temporal modeling, LLM-enhanced retrieval, abstract temporal reasoning, and recursive question decomposition (Chen et al., 2023; Qian et al., 2024; Chen et al., 2024b; Gao et al., 2024; Gong et al., 2025). Despite these advances, existing methods often decouple structural evidence construction from temporal constraint propagation, allowing early retrieval errors and temporally inconsistent paths to persist throughout multi-hop reasoning.

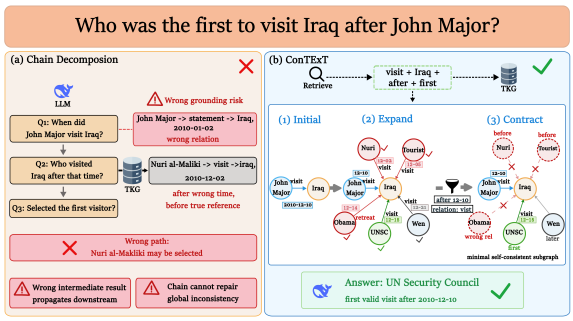

Figure 1: Comparison between chain-style decomposition and ConTExT. Chain-style reasoning commits to intermediate answers sequentially and may propagate early mistakes, while ConTExT maintains a compact reasoning subgraph and refines it under structural and temporal constraints.

Addressing *error propagation* remains a highly challenging issue for current reasoning pipelines. Existing methods construct question-specific subgraphs (Sun et al., 2018), iteratively retrieve evidence (Sun et al., 2019), or recursively decompose complex questions (Gong et al., 2025). However, because traditional subgraph expansion largely relies on local structures, early retrieval mistakes can accumulate and amplify along reasoning paths (He et al., 2021), as shown in Figure 1(a). Without continuous semantic screening and active structural compression, irrelevant neighbors can easily saturate the candidate space, severely undermining the reliability of multi-hop reasoning.

Furthermore, another critical deficiency stems from *temporal conflicts* caused by post-hoc temporal filtering. Existing TKGQA methods typically treat time in a passive manner, either as embedding-

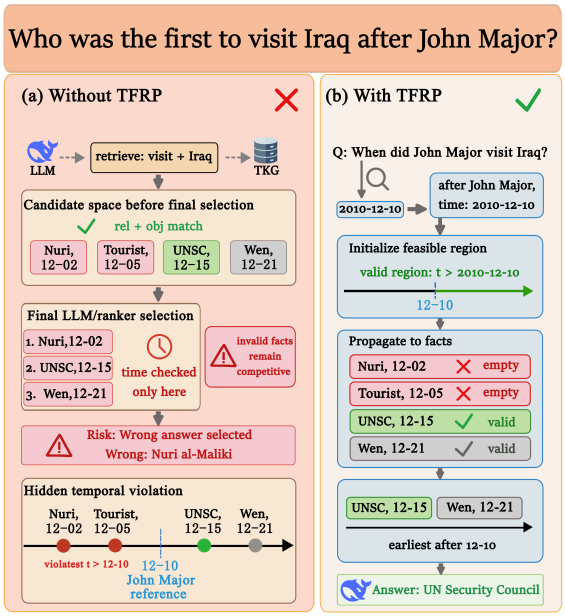

Figure 2: Without TFRP vs. with TFRP. Without TFRP, temporal validation is delayed until final ranking, allowing time-conflicting paths to remain during reasoning. With TFRP, feasible time regions are updated during search, so temporally invalid evidence can be pruned earlier.

level temporal features (Saxena et al., 2021) or as an isolated final-stage filtering signal (Qian et al., 2024). Although these strategies improve temporal sensitivity (Mavromatis et al., 2022), they fail to resolve temporal conflicts early in the reasoning process, leaving temporally inconsistent paths scattered across the search space (Chen et al., 2022). This passivity leads to a rapid growth of invalid structural combinations that appear semantically plausible but are temporally infeasible, as demonstrated in Figure 2(a).

To circumvent these limitations, we propose ConTExT, which formulates TKGQA as a unified dual-space search over structural and temporal spaces. The framework jointly optimizes the candidate space through two interactive components. 1) First, it dynamically anchors question-relevant entities and performs selective subgraph exploration. Distinct from naive neighborhood expansion, ConTExT tightly couples graph expansion with active searching: it combines path semantics with sub-question consistency to introduce query-aligned evidence, while continuously compressing the structural reasoning space. Figure 1(b) provides a concrete example illustrating this dynamic process. 2) Second, TFRP mitigates path conflicts by actively propagating temporal constraints during exploration and maintaining a dynamic temporal feasible region. By integrating time constraints as online search signals rather than passive post-filters , as shown in Figure 2(b), TFRP substantially suppresses the propagation of temporally incompatible paths, ensuring bounded computational complexity and rigorous convergence.

Our contributions are summarized as follows:

- We propose ConTExT, a unified dual-space search framework for TKGQA that jointly optimizes structural paths and temporal spaces.
- We introduce a semantic-driven structural contraction mechanism that tightly couples expansion with active searching, complemented by TFRP which suppresses temporally incompatible paths.
- Extensive experiments on MultiTQ demonstrate that ConTExT achieves an approximate 27.1% improvement on challenging multi-hop questions.

2 Preliminaries

A temporal knowledge graph (TKG) is denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, where $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{F}$ denote the sets of entities, relations, timestamps, and temporal facts, respectively. Each temporal fact is represented as a quadruple $f = (e_s, r, e_o, t) \in \mathcal{F}$, where e_s and e_o are the subject and object entities, r is the relation, and t specifies the timestamp at which the fact holds.

TKGQA aims to answer a natural language question q based on the temporal facts in $\mathcal{G}$. The answer can be either an entity or a temporal expression, depending on the expected answer type expressed in the question. Notably, TKGQA requires a model to satisfy not only entity–relation matching but also temporal constraints such as *before*, *after*, *first*, and *last*.

We use C_q to denote the structural and temporal constraints derived from q, and $A(\mathcal{G}', q)$ to denote the candidate answers supported by a subgraph $\mathcal{G}' \subseteq \mathcal{G}$.

3 Method

3.1 Overview

The operational workflow of ConTExT follows an iterative search-and-contraction paradigm over joint structural and temporal spaces to approximate a compact, self-consistent reasoning subgraph. As illustrated in Figure 3, the pipeline executes through three tightly coupled phases:

Question Parsing and Initialization: Given a query q and a graph $\mathcal{G}$, the model extracts structural and temporal constraints to initialize a localized reasoning subgraph (Figure 3, Left).

Dual-Space Joint Optimization: Starting from the initial state, the framework iteratively coordinates structural and temporal operations until convergence (Figure 3, Center):Structural Contraction: ConTExT expands the graph frontier by retrieving query-aligned evidence while continuously pruning irrelevant topology.Temporal Feasible Region Propagation (TFRP): Simultaneously, TFRP actively propagates temporal constraints across candidate facts, dynamically shrinking their feasible time regions to eliminate temporally incompatible paths.

Constrained Answer Selection: ConTExT evaluates and ranks all remaining graph-supported candidates under the accumulated joint constraints to output the final answer (Figure 3, Right).

The complete procedure is formally summarized in Algorithm 1.

Algorithm 1: ConTExT

Input: Question q, temporal knowledge graph $\mathcal{G}$
Output: Answer a^*
`// LLM-based query parsing and state initialization`
$\mathcal{O}_q \leftarrow \mathrm{LLMParse}(q)$
$(Q_q, C_q, t_q^{ref}) \leftarrow \mathrm{Normalize}(\mathcal{O}_q, \mathcal{G})$
$S_0 = (G_0, \Theta_0, C_q, \mathcal{B}_0) \leftarrow \mathrm{InitState}(Q_q, t_q^{ref}, \mathcal{G})$
`// Iterative subgraph evolution`
$i \leftarrow 0$
while $\neg\mathrm{Converged}(S_i)$ **do**
  $q_i \leftarrow \mathrm{GenSubQ}(q, S_i)$
  $\mathcal{C}_i \leftarrow \mathcal{N}(\mathcal{B}_i)$
  $G_i^+ \leftarrow G_i \cup \mathrm{Top}_K\big(\mathcal{C}_i; \mathbf{w}_{exp}^\top \mathbf{x}_{exp}(f, q_i, S_i)\big)$
  foreach $f \in G_i^+$ **do**
    $\Theta_{i+1}(f) \leftarrow \Theta_i(f) \cap \{t \in \mathcal{T} \mid \phi(t, C_q^{tmp}) = 1\}$
  $\bar{G}_{i+1} \leftarrow (\Pi_{time} \circ \Pi_{path} \circ \Pi_{rel})(G_i^+, \Theta_{i+1}, C_q)$
  $G_{i+1} \leftarrow \Pi_{role}(\bar{G}_{i+1}, Q_q)$
  $\mathcal{B}_{i+1} \leftarrow \mathrm{Frontier}(G_{i+1}, G_i)$
  $S_{i+1} \leftarrow (G_{i+1}, \Theta_{i+1}, C_q, \mathcal{B}_{i+1}); \quad i \leftarrow i+1$
`// Constrained answer selection`
$G_* \leftarrow \mathrm{FinalSubgraph}(S_i)$
$(A_q, \tilde{s}, \mathcal{F}_{top}) \leftarrow \mathrm{ScoreCandidates}(G_*, Q_q)$
$a' \leftarrow \mathrm{LLMSelect}(q, A_q, \mathcal{F}_{top})$
$a^* \leftarrow \mathrm{Guard}(a', A_q, \tilde{s}, Q_q)$
return a^*

3.2 Question Parsing and Initialization

Given an input question q, ConTExT parses it into a structured query to extract the foundational constraints:

$$\mathcal{Q}_q = (E_a, E_r, \mathcal{R}, o, y, g), \tag{1}$$

where E_a and E_r denote anchor entities and reference events; $\mathcal{R}$ denotes candidate relations; and o, y, and g represent the temporal operator, answer type, and temporal granularity, respectively. $\mathcal{Q}_q$ induces joint structural constraints $\mathcal{C}_q^{str}$ and temporal constraints $\mathcal{C}_q^{tmp}$. If no explicit timestamp is provided, the reference time t_{ref} is implicitly estimated via E_r over $\mathcal{G}$.

Instead of performing unrestricted k-hop neighbor expansion, ConTExT utilizes $\mathcal{Q}_q$ to construct a focused initial subgraph $\mathcal{G}_0$. Each temporal fact $f \in \mathcal{G}$ is evaluated by an initialization scoring function:

$$s_{init}(f) = \mathbf{w}_{init}^\top \mathbf{x}_{init}(f, \mathcal{Q}_q), \tag{2}$$

where $\mathbf{x}_{init}$ computes the joint structural and temporal compatibility between fact f and query $\mathcal{Q}_q$. The initial subgraph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ is then materialized by retrieving the top facts based on an initialization budget B:

$$\mathcal{E}_0 = \mathrm{Top}_B(\mathcal{F}; s_{init}), \tag{3}$$

where $\mathcal{V}_0$ comprises all entities involved in $\mathcal{E}_0$. This localized subgraph defines the initial reasoning state $\mathcal{S}_0 = (\mathcal{G}_0, \Theta_0, \mathcal{C}_q)$, where $\Theta_0(f) = \{t_f\}$ initializes the active temporal region for each fact, serving as the starting frontier for the subsequent dual-space joint optimization, as shown in Figure 3(left).

3.3 Iterative Subgraph Evolution

The initial subgraph may contain incomplete evidence or locally plausible but globally inconsistent facts. ConTExT therefore evolves the reasoning space through a coupled expansion–propagation–contraction process. This design allows the model to introduce new evidence when necessary, while continuously suppressing facts that violate semantic, structural, or temporal constraints.

Sub-question guided expansion. At iteration i, ConTExT derives a local sub-question q_i from the original question q and the current state S_i. Instead of treating q_i as an intermediate answer commitment, ConTExT uses it as a soft search guide to identify which relation, entity region, or reference event should be explored. For each candidate fact adjacent to the current frontier $\mathcal{B}_i$, the expansion score is computed as

$$s_{exp}(f) = \mathbf{w}_{exp}^\top \mathbf{x}_{exp}(f, q_i, S_i), \tag{4}$$

where $\mathbf{x}_{exp}$ integrates semantic relevance, relation matching, entity compatibility, and temporal com-

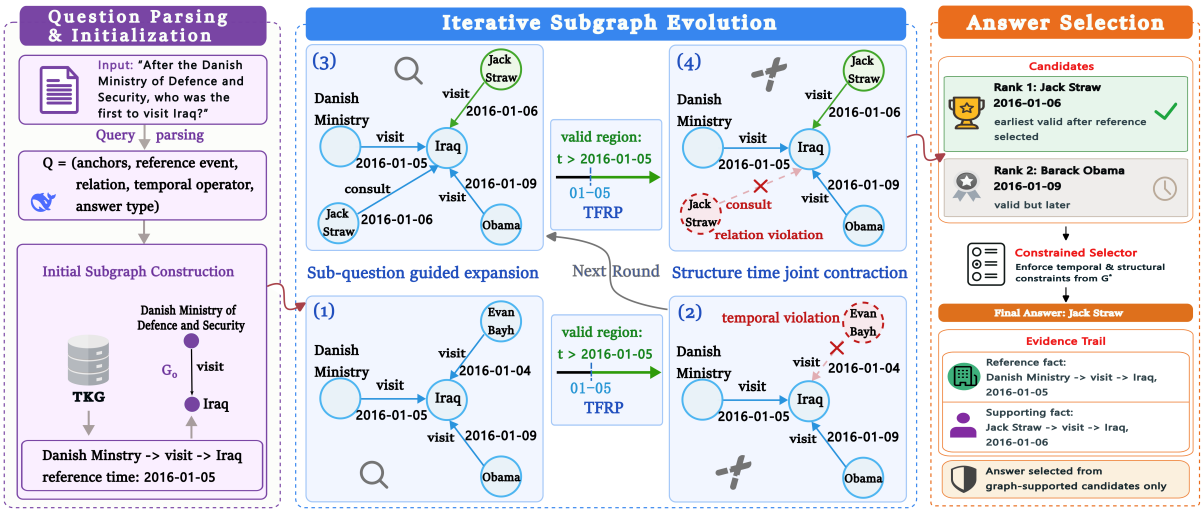

Figure 3: Overview of the ConTExT framework. ConTExT parses the input question into structured constraints, constructs an initial reasoning subgraph, iteratively evolves the subgraph through expansion, temporal propagation, and contraction, and finally resolves graph-supported candidates under structural and temporal constraints.

patibility. The expanded subgraph is

$$G_i^+ = G_i \cup \mathrm{Top}_K(\mathcal{C}_i; s_{exp}), \quad \mathcal{C}_i = \mathcal{N}(\mathcal{B}_i). \quad (5)$$

This step keeps expansion selective: new facts are introduced only when they are consistent with the current search intent and the evolving reasoning state.

Structure time joint contraction. After expansion and temporal propagation, ConTExT contracts the subgraph to preserve a compact and self-consistent reasoning space:

$$G_{i+1} = (\Pi_{role} \circ \Pi_{time} \circ \Pi_{path} \circ \Pi_{rel})(G_i^+), \quad (6)$$

where Π_{rel} removes relation-object mismatches, Π_{path} removes facts disconnected from valid reasoning paths, Π_{time} removes facts with empty feasible time regions, and Π_{role} enforces answer-side role consistency. The frontier is updated as

$$\mathcal{B}_{i+1} = \mathrm{Frontier}(G_{i+1}, G_i). \quad (7)$$

The iteration terminates when the maximum step is reached, no expandable facts remain, the subgraph becomes empty, or the subgraph no longer changes. Through repeated expansion and contraction, ConTExT gradually approaches a compact reasoning subgraph that is both structurally connected and temporally valid.

TFRP. For each fact f, ConTExT maintains a feasible time region $\Theta_i(f)$, which records the timestamps still compatible with the temporal constraints. Given $c \in C_q^{tmp}$, TFRP updates this region as

$$\Theta_{i+1}(f) = \Theta_i(f) \cap \{t \in \mathcal{T} \mid \phi(t, c) = 1\}, \quad (8)$$

where $\phi(t, c)$ checks whether timestamp t satisfies the temporal constraint c. This update makes temporal validity part of the evolving reasoning state. When $\Theta_{i+1}(f) = \emptyset$, the fact becomes temporally infeasible and is removed during contraction.

3.4 Answer Selection

After convergence, ConTExT selects the final answer from the stabilized subgraph G_*. For each candidate a, its supporting facts are

$$\mathcal{F}_*(a) = \{f \in G_* \mid \rho(f, Q_q) = a\}, \quad (9)$$

where $\rho(\cdot)$ extracts an answer according to the parsed answer type and role constraint. The candidate set is $A_q = \{a \mid \mathcal{F}_*(a) \neq \emptyset\}$. Each candidate is scored by

$$\tilde{s}(a) = |\mathcal{F}_*^{op}(a)| + \eta b(a, o, t_q^{ref}) + \Delta(a; \mathbf{x}_a), \quad (10)$$

where the three terms measure operator-compatible evidence, temporal boundary preference, and lightweight reranking features. If the top candidate is confident, ConTExT returns $\arg\max_{a \in A_q} \tilde{s}(a)$; otherwise, an LLM is used only to select from A_q, and invalid outputs are replaced by the highest-ranked graph-supported candidate.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate ConTExT on MultiTQ (Chen et al., 2023), a large-scale TKGQA bench-

mark built from ICEWS05-15. MultiTQ contains questions with multiple temporal granularities and covers both single-constraint and multi-constraint reasoning. The test split includes entity answers and time answers, which allows us to evaluate whether a model can jointly identify relevant facts, compare temporal boundaries, and return the required answer type. Table 1 reports the category distribution used in our experiments.

Table 1: Statistics of question categories in the MultiTQ benchmark.

	Category	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**	**57,979**	**54,584**

Baselines. Following previous TKGQA evaluations (Qian et al., 2024; Gong et al., 2025), we compare ConTExT with three groups of methods. The first group contains pre-trained language models and general LLM baselines, including BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2020), LLaMA2 (Touvron et al., 2023), and ChatGPT. The second group contains embedding-based TKGQA methods, including EmbedKGQA (Saxena et al., 2020), CronKGQA (Saxena et al., 2021), and MultiQA (Chen et al., 2023). The third group contains recent temporal reasoning methods, including ARI (Chen et al., 2024b), TimeR4 (Qian et al., 2024), and RTQA (Gong et al., 2025). We report Hits@1 and Hits@10, and further break down the results by question type and answer type.

Implementation Details. We implement ConTExT in Python and conduct all experiments on the official MultiTQ split. For each question, ConTExT builds a bounded local subgraph from the linked anchors. The initialization stage uses one-hop facts and keeps at most 800 facts. During subgraph evolution, ConTExT considers up to 600 frontier-neighbor facts and keeps the top 30 candidates for the final evidence pack. The final prompt is limited to the top 10 candidates with compact supporting evidence. In line with deterministic LLM-based TKGQA settings (Qian et al., 2024; Gong et al., 2025), we set the temperature to 0. The main experiments use deepseek-v4-flash as the LLM judge through an OpenAI-compatible API, with a maximum length of 1024 tokens for final answer selection. The default hyperparameter values are defined as Operator-aware ranking strength: 1.00, Multiple-boundary score margin: 0.250, and Peer-relation promotion gap: 0.20. The LLM is only allowed to select from the generated candidate evidence pack rather than freely generate an answer. All reported results are measured by Hits@1 and Hits@10.

4.2 Main Results

Overall Effectiveness. The main comparison on MultiTQ demonstrates that ConTExT achieves the strongest overall performance among all compared methods, as reported in Table 2. Compared with RTQA, the strongest baseline in Table 2, ConTExT improves overall Hits@1 by 0.6 percentage points (77.1% vs. 76.5%) and overall Hits@10 by 10.4 percentage points (87.2% vs. 76.8%). This indicates that dual-space contractive search can more effectively control both structural evidence and temporal validity than retrieval-augmented or decomposition-based reasoning pipelines.

Strong Gains on Multi-hop Reasoning. The most notable improvement appears on Multiple questions, where ConTExT increases Hits@1 from 42.4% to 53.9% compared with RTQA. This gain suggests that ConTExT is particularly effective when a question requires multiple reasoning steps or involves more than one temporal constraint. A likely reason is that ConTExT does not commit to a fixed chain of intermediate answers. Instead, it maintains an evolving reasoning subgraph and continuously removes evidence that violates structural or temporal constraints, thereby reducing the effect of early reasoning errors.

Improved Candidate Quality. The Hits@10 results further show that ConTExT improves not only the final top-ranked prediction but also the quality of the candidate set. It achieves 74.8% Hits@10 on Multiple questions and 86.6% Hits@10 on entity answers, indicating that correct answers are more frequently preserved after iterative subgraph evolution. This result supports the effectiveness of semantic-driven expansion, structural contraction, and TFRP, which jointly suppress irrelevant or temporally infeasible paths before answer selection.

Remaining Gap on Time Answers. Although ConTExT performs strongly on overall and entity-based reasoning, TimeR4 and RTQA remain more competitive on time-answer Hits@1. This suggests that direct timestamp prediction may still benefit

Table 2: Performance comparison on the MultiTQ test set. The best and second-best results are marked in bold and underlined, respectively.

Model	Hits@1 (%)					Hits@10 (%)				
	Overall	Multiple	Single	Entity	Time	Overall	Multiple	Single	Entity	Time
BERT	8.3	6.1	9.2	10.1	4.0	44.1	39.2	46.1	53.1	22.2
DistilBERT	8.3	7.4	8.7	10.2	3.7	48.2	42.6	50.5	59.1	21.6
ALBERT	10.8	8.6	11.6	13.9	3.2	48.4	41.5	51.2	58.9	22.8
LLaMA2	18.5	10.1	22.0	23.9	5.5	–	–	–	–	–
ChatGPT	10.2	7.7	14.7	13.7	2.0	–	–	–	–	–
EmbedKGQA	20.6	13.4	23.5	29.0	0.1	45.9	43.9	46.7	64.8	0.1
CronKGQA	27.9	13.4	33.7	32.8	15.6	60.8	45.3	67.1	69.6	39.2
MultiQA	29.3	15.9	34.7	34.9	15.7	63.5	51.9	68.2	73.3	39.6
ARI	38.0	21.0	68.0	39.4	34.4	–	–	–	–	–
TimeR4	72.8	33.5	88.7	63.9	**94.5**	–	–	–	–	–
RTQA	76.5	42.4	**90.2**	69.2	94.2	76.8	42.7	90.7	69.7	**94.2**
ConTExT	**77.1**	**53.9**	86.2	**74.5**	82.6	**87.2**	**74.8**	**92.2**	**86.6**	88.5

from specialized temporal calibration or generation-oriented mechanisms. Nevertheless, ConTExT shows clearer advantages in structurally complex cases, especially where entity reasoning, multi-hop evidence, and temporal constraints must be jointly considered.

4.3 Ablation Study

Component Contributions. The ablation results in Table 3 examine whether the performance of ConTExT comes from its dual-space search design rather than from a single scoring heuristic. Each variant removes one core component while keeping the same evaluation protocol and comparable search budget. Here, "w/o" denotes removing the corresponding module from the full framework.

Table 3: Ablation results on MultiTQ Hits@1.

Model	Overall	Multiple	Single	Entity	Time
ConTExT	**77.84**	**54.69**	**86.94**	**75.26**	**83.89**
w/o expansion	77.66	54.38	86.82	75.02	83.85
w/o contraction	76.75	53.52	86.03	74.36	83.51
w/o TFRP	51.36	19.52	63.74	39.03	80.03

Expansion Safely Boosts Topology Recall. Expansion safely boosts topology recall: removing expansion leads to a slight but consistent decline, with Overall Hits@1 decreasing from 77.84% to 77.66%, Multiple questions from 54.69% to 54.38%, and Entity answers from 75.26% to 75.02%. The small drop indicates that the initial subgraph already covers many useful facts, while expansion still recovers complementary topology around the anchored evidence. Its gains are therefore concentrated on questions that require additional connected facts beyond the initial neighborhood. This supports selective expansion as a controlled recall mechanism rather than unrestricted neighborhood growth.

Contraction Purges Inconsistent Candidates to Fine-Tune Reasoning Precision. Contraction purges inconsistent candidates to fine-tune reasoning precision: removing contraction causes a smaller but consistent decline, with Overall Hits@1 decreasing from 77.84% to 76.75%, Multiple questions from 54.69% to 53.52%, and Time answers from 83.89% to 83.51%. This suggests that expansion and TFRP recover and temporally regulate most necessary evidence, while contraction sharpens the final reasoning space by filtering relation-object mismatches, disconnected paths, temporally infeasible facts, and role-inconsistent candidates. Its effect is most visible when several semantically plausible facts compete but only part of them remains compatible with the evolved subgraph.

TFRP Actively Suppresses Temporal Conflicts. Removing TFRP causes the most substantial degradation. Overall Hits@1 drops from 77.84% to 51.36%, while Multiple-question performance falls sharply from 54.69% to 19.52%. Entity-answer performance also decreases from 75.26% to 39.03%, showing that temporal conflicts can mislead entity selection, not only timestamp prediction. This confirms that temporal information should actively regulate the search process. Without TFRP, relation-compatible but temporally invalid facts remain in the intermediate reasoning space, which increases the number of misleading candidates and weakens multi-hop reasoning stability.

4.4 LLM Analysis

We further evaluate ConTExT with different LLMs as the final constrained selector. This experiment

does not aim to compare the parametric knowledge of these LLMs. Instead, all models receive the same structured candidate evidence pack and must select from the provided candidates. As shown in Table 4, gpt-4o obtains the best overall Hits@1, while deepseek-v4-flash, gemini-2.0-flash, and gpt-4o-mini remain close.

Table 4: ConTExT results with different LLM backbones on MultiTQ Hits@1.

Model	Overall	Multiple	Single	Entity	Time
deepseek-v4-flash	78.53	53.98	87.73	75.25	86.30
gemini-2.0-flash	78.07	53.86	87.13	74.96	85.39
gpt-4o	**78.83**	**54.48**	**87.96**	**75.48**	**86.74**
gpt-4o-mini	77.67	52.70	87.04	73.89	86.56

The overall range across the four LLMs is only 1.16 points, indicating that ConTExT is not strongly tied to a single LLM backend. This stability comes from the separation between symbolic evidence control and language-model selection. Query parsing, subgraph evolution, and candidate ranking provide most of the search constraints, while the LLM mainly resolves ambiguous comparisons inside a compact evidence pack. The small gap between gpt-4o and gpt-4o-mini also suggests that once the reasoning space is well restricted, a smaller LLM can remain competitive. This is important for practical deployment, where inference cost and latency are often as important as final accuracy.

4.5 Parameter Sensitivity

To examine parameter sensitivity, we vary one coefficient at a time while keeping the remaining settings fixed, as shown in Figure 4. This one-factor design isolates the contribution of each constraint signal in ConTExT.

Operator-aware ranking strength (OAR). OAR adjusts the influence of temporal-operator compatibility during candidate ranking. Raising OAR from 0.25 to 1.00 improves overall Hits@1 from 70.65% to 77.11%, indicating that the final answer is sensitive to whether candidates satisfy operators such as *before*, *after*, *first*, *last*, and equality. The curve also shows that further increasing this factor does not bring additional gains, suggesting that temporal ordering should guide the ranking process without dominating all structural evidence.

Multiple-boundary score margin (MBM). MBM strengthens the separation among candidates in questions that involve several temporal boundaries. Its improvement is more moderate than OAR but remains stable across the tested range, which is consistent with the role of MBM: it mainly helps cases where an answer must satisfy more than one boundary condition rather than simple single-operator queries. The slight decline beyond the default setting implies that overly strict boundary margins may discard facts that are still compatible with the intended reasoning path.

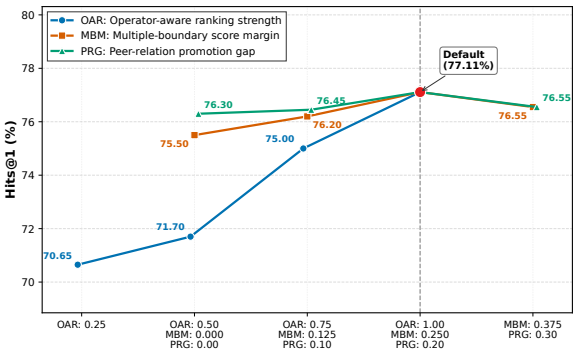

Figure 4: Parameter sensitivity on overall Hits@1.

Peer-relation promotion gap (PRG). PRG refines the ranking of semantically close relation candidates after temporal feasibility has been established. A larger gap helps prevent neighboring relations with similar surface forms from sharing nearly identical scores, thereby making the top candidate more stable. However, the curve is relatively flat compared with OAR, showing that PRG is a secondary calibration factor rather than the main source of improvement.

4.6 Case Study

We analyze a MultiTQ test question labeled as Multiple: "After Tony Blair, which country was the first to express an interest in cooperating with Iraq?" This case requires two coupled reasoning steps. The model must first resolve the reference event *(Tony Blair, express intent to cooperate, Iraq, 2006-05-24)*, and then search for the earliest country-level subject satisfying the same cooperation-intent relation with Iraq after this boundary. As illustrated in Figure 5, a flat retriever can collect many relation-compatible facts, including *(France, express intent to cooperate, Iraq, 2005-03-27)*, *(Kuwait, express intent to cooperate, Iraq, 2005-06-09)*, *(Japan, express intent to cooperate, Iraq, 2005-12-06)*, and *(Iran, express intent to cooperate, Iraq, 2006-03-17)*. Although these facts match the target relation and object, they precede the reference event and should not support the answer.

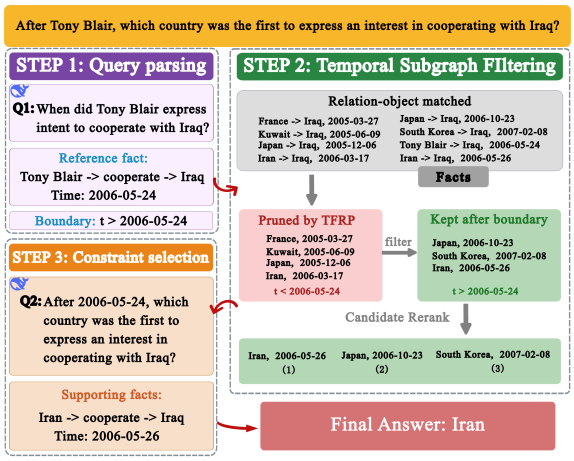

Figure 5: Case study of ConTExT on a multiple-constraint MultiTQ question. ConTExT first grounds the reference event, propagates the resulting temporal boundary, removes relation-compatible but temporally invalid facts, and selects the earliest valid answer fact from the evolved subgraph.

ConTExT handles this case by explicitly separating reference grounding, temporal propagation, and answer selection. The reference fact fixes the feasible region to timestamps later than 2006-05-24. TFRP removes all candidate facts whose timestamps fall outside this region, while contraction keeps only facts that remain connected to the parsed relation-object pattern and answer role. In the remaining subgraph, *(Iran, express intent to cooperate, Iraq, 2006-05-26)* is the earliest valid candidate, whereas later valid facts such as *(Japan, express intent to cooperate, Iraq, 2006-10-23)* and *(South Korea, express intent to cooperate, Iraq, 2007-02-08)* satisfy the boundary but lose under the *first* operator. The final answer is therefore Iran. This example explains why ConTExT is especially effective on Multiple questions: the method does not merely retrieve facts that mention the right relation, but constructs a compact reasoning subgraph in which the reference event, temporal boundary, answer role, and ordering operator must be jointly satisfied.

5 Conclusion

In this paper, we introduce ConTExT, a framework for temporal knowledge graph question answering that reformulates temporal reasoning as a constrained search process over a compact reasoning subgraph. Rather than retrieving independent facts and relying on free-form generation, ConTExT progressively constructs a graph-supported reasoning space through query parsing, focused subgraph construction, sub-question guided expansion, structure time joint contraction, and TFRP. The final answer is resolved from graph-supported candidates under explicit structural and temporal constraints. Experimental results on the evaluated TKGQA benchmark show that ConTExT effectively improves complex temporal reasoning, particularly for questions involving temporal ordering, multi-hop evidence, and multiple temporal constraints.

Limitations

Although ConTExT shows promising performance, several limitations remain. First, while the framework is effective for multi-hop questions, its advantage is less evident on single-hop questions. This suggests that contractive subgraph search may introduce unnecessary filtering when the answer can be obtained from a relatively direct evidence path. Second, the current temporal modeling mainly assumes normalized discrete timestamps and explicit comparison operators. More complex cases involving vague temporal expressions, interval-valued facts, or duration-based reasoning require more expressive temporal constraint modeling. Future work will focus on adaptive search control for different question complexities, more robust reference-time estimation, and interval-aware temporal reasoning to improve the generalizability of ConTExT across broader TKGQA scenarios.

References

Zhuo Chen, Zhao Zhang, Zixuan Li, Fei Wang, Yutao Zeng, Xiaolong Jin, and Yongjun Xu. 2024a. 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)*, pages 14579–14594.

Ziyang Chen, Dongfang Li, Xiang Zhao, Baotian Hu, and Min Zhang. 2024b. Temporal knowledge question answering via abstract reasoning induction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4872–4889.

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)*, pages 11378–11392.

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

Sitao Cheng, Ziyuan Zhuang, Yong Xu, Fangkai Yang, Chaoyun Zhang, Xiaoting Qin, Xiang Huang, Ling Chen, Qingwei Lin, Dongmei Zhang, et al. 2024. Call me when necessary: LLMs can efficiently and faithfully reason over structured environments. *arXiv preprint arXiv:2403.08593*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yifu Gao, Linbo Qiao, Zhigang Kan, Zhihua Wen, Yongquan He, and Dongsheng Li. 2024. Two-stage generative question answering on temporal knowledge graph using large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6719–6734.

Zhaoyan Gong, Juan Li, Zhiqiang Liu, Lei Liang, Huajun Chen, and Wen Zhang. 2025. RTQA: Recursive thinking for complex temporal knowledge graph question answering with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 553–561. Association for Computing Machinery.

Qianyi Hu, Xinhui Tu, Cong Guo, and Shunping Zhang. 2025. Time-aware ReAct agent for temporal knowledge graph question answering. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6028–6039.

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. *arXiv preprint arXiv:1909.11942*.

Yonghao Liu, Di Liang, Fang Fang, Sirui Wang, Wei Wu, and Rui Jiang. 2023. Time-aware multiway adaptive fusion network for temporal knowledge graph question answering. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Adesoji Adeshina, Phillip R. Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2022. TempoQR: Temporal question reasoning over knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5825–5833.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. 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*, pages 6942–6952.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. 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*, pages 6663–6676.

Apoorv Saxena, Aditay Tripathi, and Partha 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*, pages 4498–4507.

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 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)*, pages 8017–8026. Association for Computational Linguistics.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2380–2390.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tingyi Zhang, Jiaan Wang, Zhixu Li, Jianfeng Qu, An Liu, Zhigang Chen, and Hongping Zhi. 2024. MusTQ: A temporal knowledge graph question answering dataset for multi-step temporal reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11688–11699.

A Related Work

Temporal Knowledge Graph Question Answering (TKGQA) requires models to answer natural language questions over temporally annotated facts. Compared with conventional KGQA, TKGQA must jointly consider entity–relation semantics, temporal operators, event ordering, and multi-hop dependencies. Existing studies are closely related to three aspects of our problem setting: controlling error propagation in multi-hop evidence construction, modeling temporal constraints during TKGQA, and using LLMs for complex temporal reasoning.

A.1 Error Propagation in Multi-hop Evidence Construction

Multi-hop KGQA methods usually construct an intermediate reasoning space before predicting the final answer. Early studies build question-specific subgraphs and perform neural reasoning over retrieved evidence. GRAFT-Net constructs a heterogeneous subgraph from knowledge bases and text, while PullNet further retrieves evidence iteratively to expand the question-specific subgraph (Sun et al., 2018, 2019). Later work introduces intermediate supervision to improve multi-hop reasoning stability and reduce spurious paths (He et al., 2021). These methods show that explicit evidence construction is important for multi-hop question answering, but their expansion process is mainly driven by local graph neighborhoods or learned retrieval scores. When irrelevant nodes are introduced early, later reasoning may still proceed over a noisy and enlarged search space.

Recent LLM-enhanced graph reasoning methods further explore planning, path discovery, and interaction with structured environments (Luo et al., 2023; Cheng et al., 2024). Such methods improve semantic flexibility and interpretability, but they often rely on step-wise decisions or generated reasoning paths. This makes the reasoning process vulnerable to accumulated errors when an early path, entity, or intermediate answer is incorrect. In TKGQA, this issue becomes more severe because an intermediate state must be not only structurally plausible but also temporally valid. RTQA explicitly addresses this problem by recursively decomposing complex temporal questions and aggregating multiple answer sources to mitigate error propagation (Gong et al., 2025). However, decomposition-based reasoning still depends on the correctness of intermediate sub-questions and answers, and it does not fully replace the need for a globally constrained reasoning space.

A.2 Temporal Constraint Modeling and Delayed Validation

A central challenge in TKGQA is how to incorporate time into question answering. Early TKGQA methods formulate the task as temporal knowledge graph completion or representation matching. CronKGQA projects question representations into temporal KG embedding spaces, while TempoQR augments question embeddings with contextual, entity-aware, and time-aware modules (Saxena et al., 2021; Mavromatis et al., 2022). Other studies improve time sensitivity through temporal fusion, temporal-aware scoring, or subgraph-based reasoning (Shang et al., 2022; Liu et al., 2023; Chen et al., 2022). These methods demonstrate that temporal information is essential for TKGQA, especially when questions involve implicit time scopes or time-dependent entity states.

More recent benchmarks and models further emphasize temporal granularity and complex temporal constraints. MultiTQ introduces multi-granularity temporal question answering, where questions may involve different time units and temporal operators (Chen et al., 2023). MusTQ further focuses on multi-step temporal reasoning over temporal knowledge graphs (Zhang et al., 2024). TimeR4 incorporates time-aware retrieval and reranking to reduce temporal hallucination in LLM-based TKGQA (Qian et al., 2024). Despite these advances, temporal information is often used as an embedding feature, a retrieval signal, or a final-stage ranking criterion. Such designs can filter obviously invalid answers, but temporally conflicting facts may remain in the intermediate reasoning space until late-stage selection. Therefore, complex TKGQA still requires more explicit temporal-state control during the reasoning process, rather than treating time only as a post-hoc validation signal.

A.3 LLM-enhanced Complex Temporal Reasoning

Large language models have recently been introduced into TKGQA to improve question interpretation, temporal reasoning, and evidence utilization. ARI divides temporal reasoning into knowledge-agnostic and knowledge-based stages, enabling LLMs to use abstract reasoning guidance and factual support (Chen et al., 2024b). Two-stage gener-

ative methods combine LLM-generated representations with graph information to improve temporal question answering (Gao et al., 2024). Prog-TQA represents temporal constraints through executable programs, while TempAgent uses time-aware tools to handle temporal constraints in an agent-style reasoning process (Chen et al., 2024a; Hu et al., 2025). These methods show the benefit of LLMs in parsing complex questions and interpreting temporal semantics.

However, LLM-centered temporal reasoning also introduces new challenges. If the model freely generates intermediate reasoning steps, the process may become difficult to verify against graph evidence. If it repeatedly calls an LLM for decomposition, retrieval, and selection, the framework may incur high inference cost and remain sensitive to early mistakes. TimeR4 and RTQA are representative attempts to strengthen LLM-based TKGQA through time-aware retrieval and recursive decomposition, respectively (Qian et al., 2024; Gong et al., 2025). Nevertheless, the two limitations discussed above remain central: the reasoning process needs stronger control over evidence scope to reduce error propagation, and temporal constraints need to participate more actively in intermediate reasoning rather than being delayed to the final answer stage.