Let the Agent Search: Autonomous Exploration Beats Rigid Workflows in Temporal Question Answering

Anonymous ACL submission

Abstract

Temporal Knowledge Graph Question Answering (TKGQA) is challenging because it requires multi-hop reasoning under complex temporal constraints. Recent LLM-based approaches have improved semantic modeling for this task, but many still rely on fixed reasoning workflows or costly post-training, which can limit adaptability and make error recovery difficult. We show that *enabling an off-the-shelf Large Language Model (LLM) to determine its next action* is already effective in a zero-shot setting. Based on this insight, we propose **AT2QA**, an **A**utonomous and **T**raining-free **A**gent for **T**KG **Q**uestion **A**nswering. AT2QA empowers the LLM to iteratively interact with the TKG via a generic search tool, inherently enabling autonomous exploration and dynamic self-correction during reasoning. To further elicit the LLM's potential for complex temporal reasoning, we introduce a training-free experience mining mechanism that distills a compact few-shot demonstration library from successful self-generated trajectories. AT2QA also yields a transparent audit trail for every prediction. Experiments on three challenging benchmarks—MultiTQ, Timeline-CronQuestion, and Timeline-ICEWS-Actor—show that AT2QA achieves new state-of-the-art performance, surpassing the strongest baselines by 10.7, 4.9, and 11.2 absolute points, respectively. Our code is available at [Anonymous GitHub].

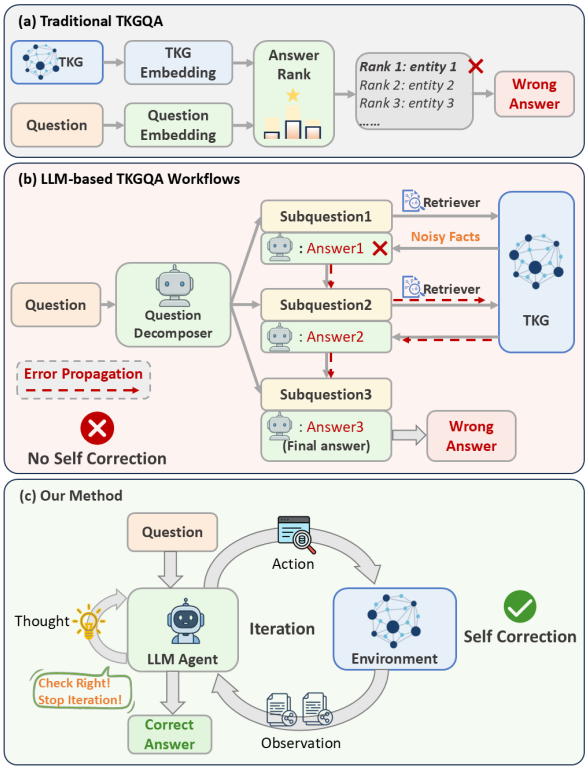

Figure 1: Comparison between **AT2QA** and existing methods. **(a) Embedding-based Methods** rely on vector representations, lacking semantic understanding. **(b) LLM-based Workflows** execute fixed pipelines, making them vulnerable to cascading errors. **(c) Our AT2QA** operates as an autonomous agent. It autonomously explores and self-corrects its reasoning trajectory by iteratively interacting with the TKG.

1 Introduction

While traditional Knowledge Graphs (KGs) have long served as a fundamental infrastructure for question answering, real-world facts are inherently dynamic and time-dependent. To prevent models from relying on outdated knowledge, Temporal Knowledge Graphs (TKGs) capture this evolution by extending static facts into quadruples denoted as $< subject, relation, object, timestamp >$ (Saxena et al., 2021). Consequently, Temporal Knowledge Graph Question Answering (TKGQA) is substantially more challenging than conventional static Knowledge Graph Question Answering (KGQA). This is because answering complex temporal questions often requires multi-hop reasoning over structural entities while simultaneously satisfying combined or multi-granular temporal constraints.

In recent years, TKGQA has shifted from traditional task-specific architectures to Large Language Model (LLM)-based frameworks. Traditional embedding-based methods (Mavromatis et al., 2022; Chen et al., 2023) map questions and

TKG facts into low-dimensional vector spaces and rank candidate answers with scoring functions (as illustrated in Figure 1(a)), but they often struggle to capture the complex semantics of temporal constraints in natural language questions (Chen et al., 2023). In contrast, LLMs have shown strong performance on complex natural language tasks (Yang et al., 2025; Peng et al., 2025). Consequently, recent studies have increasingly explored the potential of LLMs for TKGQA (Chen et al., 2024b; Qian et al., 2024; Gong et al., 2025; Qian et al., 2025). These methods typically incorporate dynamic graph knowledge into LLMs through retrieval-augmented generation or fine-tuning under predefined workflows. Nevertheless, existing approaches still exhibit three key limitations:

(1) Rigid Workflows and Cascading Errors. Existing methods typically constrain LLMs within fixed, human-designed reasoning pipelines rather than allowing them to interact with the TKG environment to determine their next actions. This rigidity limits the model's ability to autonomously explore its reasoning trajectory and dynamically self-correct during execution. As illustrated in Figure 1(b), when strictly following a predefined decomposition workflow, an initial retrieval failure inevitably cascades through subsequent steps, amplifying the error and ultimately yielding an incorrect final answer.

(2) Prohibitive Training Cost and Limited Generalization. To bridge the mismatch between LLMs and TKGs, prior work largely depends on costly post-training, ranging from SFT-based methods (Gao et al., 2024; Qian et al., 2024, 2025) to recent RL-based agents such as Temp-R1 and TKG-Thinker, the latter even requiring a two-stage SFT+RL pipeline (Gong et al., 2026; Jiang et al., 2026). However, RL training is widely recognized as computationally costly and complex (Sidahmed et al., 2024). Beyond the hardware burden, graph-specific post-training can also reduce plug-and-play transfer to unseen or dynamically evolving TKGs.

(3) Limited Interpretability. While recent LLM-based methods offer intermediate reasoning chains as explanations (Gong et al., 2025; Qian et al., 2025), their rigid pipelines separate reasoning from dynamic graph exploration. As a result, when early retrieval steps fail, these explanations can become ungrounded and susceptible to hallucination. Without a verifiable audit trail, it is difficult to determine whether failures arise from flawed reasoning or incorrect retrieved evidence.

To overcome these limitations, we propose **AT2QA** (**A**utonomous and **T**raining-free **A**gent for **T**KG **Q**uestion **A**nswering), a framework that enables autonomous temporal reasoning without parameter updates. As shown in Figure 1(c), AT2QA equips an LLM with a generic search tool and allows it to interact with the TKG environment in an iterative manner. Instead of following a fixed workflow, the model can iteratively verify retrieved evidence and reformulate queries when current evidence is insufficient or contradictory, thereby improving recovery from intermediate errors in multi-step reasoning.

To make such autonomous reasoning more reliable on complex queries, AT2QA further introduces a training-free experience mining mechanism. Using rule-based rewards, it extracts high-quality demonstration trajectories from the model's own successful explorations and uses them as few-shot guidance, without any parameter updates. In addition, because each internal `<think>` process, autonomous `<search>` action, and environmental `<observation>` is logged, AT2QA provides an explicit and auditable evidence chain for each prediction. Our main contributions are summarized as follows:

(1)An Autonomous Agent Framework for TKGQA: We introduce AT2QA, a novel autonomy-first agent framework. By enabling the LLM to iteratively interact with the TKG environment, our method inherently enables dynamic self-correction and mitigates cascading errors in complex temporal reasoning.

(2)Training-Free Experience Mining: We propose a highly efficient experience mining strategy. By distilling a compact few-shot library from the model's self-generated successful trajectories, this mechanism further enhances the model's potential for complex temporal reasoning without any parameter updates and facilitates broad plug-and-play generalization.

(3)State-of-the-Art Performance and Traceability: AT2QA achieves state-of-the-art performance on three challenging TKGQA benchmarks (MultiTQ, Timeline-CronQuestion, and Timeline-ICEWS-Actor) with Hits@1 scores of 88.7%, 75.4%, and 75.4%, outperforming the best baselines by 10.7, 4.9, and 11.2 absolute points, respectively. AT2QA also yields a transparent and verifiable audit trail for every prediction.

2 Related Work

2.1 Traditional TKGQA

Traditional Temporal knowledge graph question answering (TKGQA) primarily relied on representation learning and logical parsing. Embedding-based methods (Saxena et al., 2021; Mavromatis et al., 2022; Chen et al., 2023) and approaches like TSQA (Shang et al., 2022), which formulate the task as temporal knowledge graph completion, encode entities and temporal relations into low-dimensional spaces and rely on scoring functions to evaluate the plausibility of candidate facts. Although these methods laid a crucial foundation for the field, traditional embedding representations often act as opaque "black boxes" with weak system interpretability (Cai et al., 2023). In contrast, semantic parsing-based methods (Jia et al., 2018; Neelam et al., 2022; Ding et al., 2022; Chen et al., 2024a) attempt to translate natural language questions into logical query expressions, while Graph Neural Networks (GNNs) (Jia et al., 2021; Mavromatis et al., 2023; Liu et al., 2023) have been introduced to capture the complex structural dependencies within the graphs. These traditional paradigms suffer from a common bottleneck: they typically demand prohibitive resources for specialized training, making it exceedingly difficult to generalize to unseen TKGs.

2.2 LLM-based TKGQA Workflows

In recent years, the introduction of Large Language Models has profoundly driven a paradigm shift in TKGQA, rapidly steering the research focus toward leveraging the powerful in-context learning and semantic parsing capabilities of LLMs for RAG or interactive querying over TKGs. Existing LLM integration methods primarily focus on RAG and prompt engineering: ARI (Chen et al., 2024b) enhances the temporal adaptability of models via time-aware training signals; TimeR^4 (Qian et al., 2024) and PoK (Qian et al., 2025) generate more comprehensive reasoning plans by improving retrieval components; and TempAgent (Hu et al., 2025) adapts the ReAct paradigm to the temporal domain by designing a toolkit with 10 specific temporal tools. RTQA (Gong et al., 2025) adopts a bottom-up decomposition strategy to solve sub-questions recursively, while MemoTime (Tan et al., 2025) utilizes closed-source APIs for reasoning and stores solution paths as memories. Nevertheless, these approaches either rely on rigid, human-designed hard-coded paths, thereby constraining LLMs of their intrinsic global planning and autonomous self-correction capabilities, or necessitate prohibitively expensive and time-consuming SFT training. Recently, Temp-R1 (Gong et al., 2026) and TKG-Thinker (Jiang et al., 2026) have explored the use of reinforcement learning to equip agents with temporal reasoning capabilities. Despite their contributions to optimization efficiency, these methods still fundamentally depend on updating model parameters.

3 Intuition

Before presenting our methodology AT2QA, we detail two pivotal empirical observations that establish the foundation of our framework. These findings challenge the prevailing assumption in TKGQA that high performance necessitates either complex supervised fine-tuning or rigid, human-crafted reasoning workflows.

3.1 The Language Model is Smart Enough to Decide What to Do Next

Recent research in TKGQA typically constrains LLMs within static decomposition frameworks or predefined reasoning paths(Gong et al., 2026; Jiang et al., 2026). Such approaches implicitly assume that LLMs lack the capability to independently navigate the intricate temporal constraints and structural dependencies of knowledge graphs(Qian et al., 2025). However, our preliminary experiments suggest a contrary conclusion.

We observe that when an off-the-shelf LLM is equipped with a generic `search` tool and granted the *autonomy* to determine when and what to retrieve, it exhibits strong planning proficiency. As illustrated in Figure 2, even in a **zero-shot setting without any parameter updates**, AT2QA significantly outperforms current state-of-the-art (SOTA) methods that rely on extensive supervised training or rigid, human-crafted reasoning workflows. This phenomenon indicates that modern LLMs already possess the intrinsic intelligence required to solve complex temporal queries. **The bottleneck lies not in the model's reasoning capacity, but in the lack of an appropriate interface that allows the model to exercise its autonomy for information retrieval**.

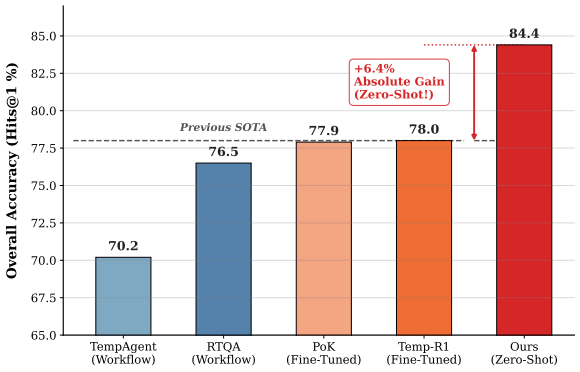

Figure 2: Performance comparison on the MultiTQ benchmark. In a zero-shot setting, the autonomous LLM agent surpasses existing baselines, highlighting the efficacy of unlocking the model's inherent decision-making capabilities.

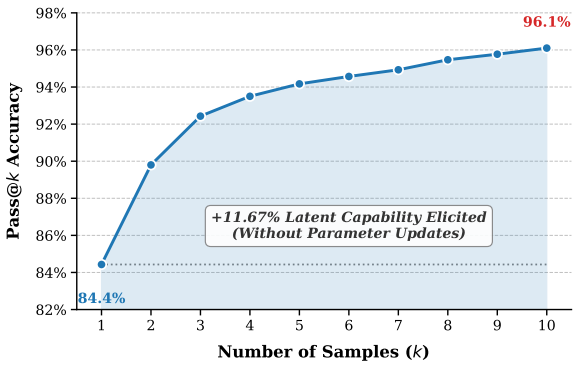

Figure 3: Pass@k analysis of our method. In a zero-shot setting, our method achieves >84% Pass@1 accuracy. For difficult queries that initially fail, repeated sampling ($k = 10$) successfully retrieves the correct answer, suggesting the reasoning capability is present but dormant.

3.2 Eliciting Capabilities Instead of Fine-Tuning

The second observation addresses the necessity of computational heavy-lifting. While Supervised Fine-Tuning is a standard paradigm to align models with TKG tasks, we hypothesize that the requisite reasoning patterns are already latent within the pre-trained weights of strong LLMs.

To validate this, we conducted a *Pass@k* analysis on a randomly sampled subset of 3,000 questions from the MultiTQ benchmark. Specifically, the *Pass@k* metric evaluates whether at least one out of k independent generation attempts yields the correct answer(Yue et al., 2025; Lv et al., 2025). As depicted in Figure 3, AT2QA achieves an impressive accuracy of over **84%** at *Pass@1* in a zero-shot setting. Crucially, for the subset of "hard" queries where the model initially failed, we observed that repeated sampling (increasing k) rapidly closes the performance gap. At $k = 10$, the model is able to generate at least one correct reasoning path for nearly all queries.

This finding indicates that **the model inherently possesses the capability to solve even the most challenging temporal reasoning problems.** Empirical evidence proves that the correct reasoning trajectory **already exists** within the model's latent space; the model effectively "knows" the solution but failed to assign the highest probability to the optimal path in a single inference step. Crucially, the observation that the model converges to the correct answer with a minimal number of trials ($k < 10$) strongly suggests that computationally expensive fine-tuning may be avoidable in TKGQA. Such a low k threshold implies that the reasoning capability is easily accessible and can likely be deterministically activated solely through appropriate prompting strategies(Snell et al., 2024; Wang et al., 2023). Consequently, this perspective suggests shifts from *injecting* new capabilities via parameter updates to *eliciting* dormant capabilities via optimal prompting. **Our problem thus evolves into identifying the most effective prompts**—synthesizing experience from successful samples—to stably trigger the LLM's latent potential without the need for gradient updates.

4 Methodology

In this section, we introduce AT2QA, a fully autonomous and training-free LLM agent framework capable of dynamic self-correction for TKGQA Question Answering. AT2QA consists of two core components, as shown in Figure 4: a *retrieval-augmented reasoning agent* equipped with a structured temporal search tool, and a *trajectory optimization strategy* that mines effective few-shot demonstrations from the model's self-generated experiences.

4.1 Preliminaries and Problem Formulation

A Temporal Knowledge Graph (TKG) is defined as a collection of quadruples $\mathcal{G} = \{(e_h, r, e_l, \tau)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the sets of entities, relations, and timestamps, respectively. Given a natural language question Q, the goal is to derive the correct answer y by reasoning over the facts in $\mathcal{G}$. Unlike static QA, Q typically contains implicit or explicit temporal constraints that require filtering facts based on τ.

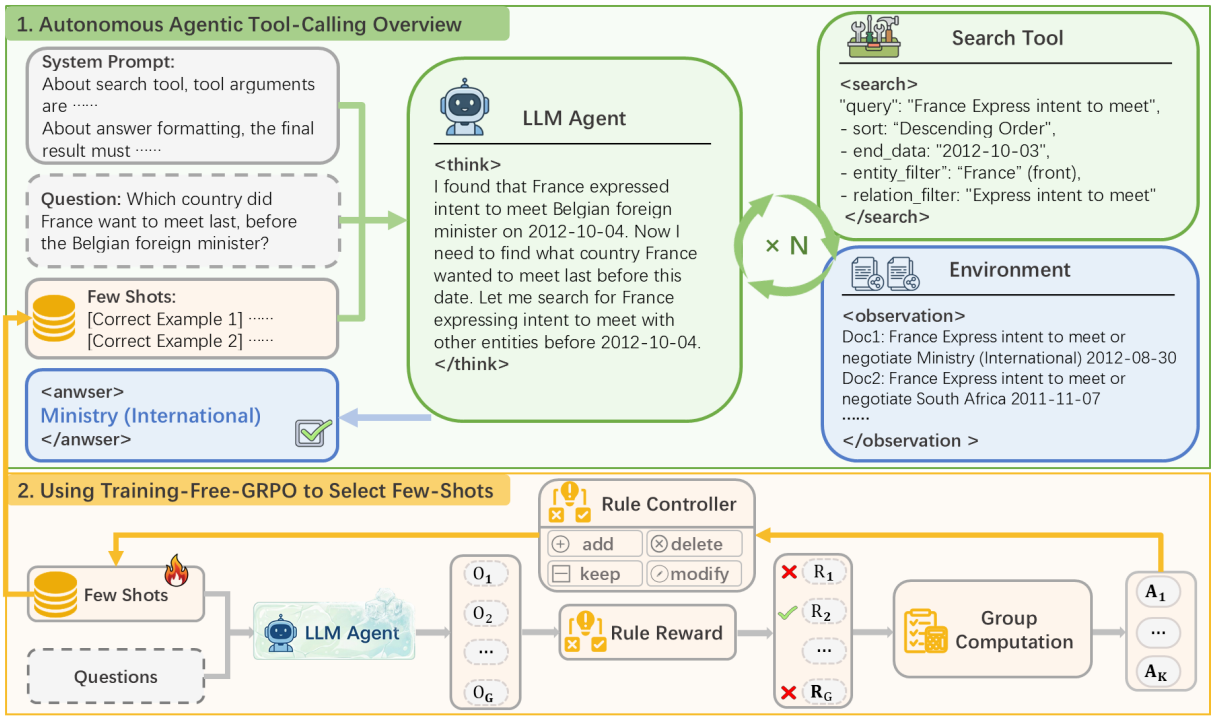

Figure 4: The overview of our proposed framework AT2QA. Top: At inference, an LLM agent repeatedly queries a `Search` tool to interact with the TKG environment until sufficient evidence is collected. Inputs include a system prompt, the question, and few-shot demonstrations. Bottom: The few-shot library is selected from candidate trajectories via training-free GRPO-style rule editing with rule-based rewards.

4.2 Tool-Augmented Reasoning Framework

To bridge the gap between the LLM's parametric knowledge and the structured data in $\mathcal{G}$, we develop a `Search` tool that supports both semantic retrieval and symbolic filtering.

Structured Temporal Retrieval. The `Search` tool accepts a query string q along with a set of structured constraints $C = \{c_{time}, c_{entity}, c_{rel}\}$.

- **Filtering:** Before semantic matching, the tool filters $\mathcal{G}$ to obtain a candidate set $\mathcal{G}_{sub} \subset \mathcal{G}$. Constraints act as a conjunctive filter: c_{time} specifies an inclusive time window $[\tau_{start}, \tau_{end}]$; c_{entity} restricts involved entities (e.g., head/tail candidates); and c_{rel} filters for exact relation matches. This ensures that retrieval is grounded in valid temporal and structural contexts.

- **Semantic Ranking:** We employ a dense retrieval approach. Each fact $f \in \mathcal{G}_{sub}$ and the query q are encoded into a shared embedding space. Facts are ranked based on the cosine similarity score $s(f, q) = \cos(enc(f), enc(q))$.

- **Hybrid Sorting:** To handle temporal nuances, the tool supports two sorting modes: (1) *Relevance-based*, sorting solely by $s(f, q)$, and (2) *Time-based*, where the top-m relevant facts are re-ranked chronologically. This exposes the temporal evolution of facts to the agent.

Reasoning Process. The LLM operates as an autonomous agent. At each step t, it generates a thought and a tool call action $a_t = \texttt{Search}(q_t, C_t)$ based on the history of previous actions and observations. The iterative loop continues until the agent generates a special termination token or reaches the maximum step limit T_{max}. This multi-turn interaction allows the agent to perform *Self-Correction*: if retrieved evidence conflicts with the hypothesis, the agent can refine its constraints C_{t+1} (e.g., expanding the time window) in subsequent turns.

4.3 Training-Free Experience Mining

To enhance the agent without any parameter updates, we adopt a **Training-Free GRPO-style** experience selection scheme (Cai et al., 2025). Instead of optimizing model weights, we generate a pool of candidate reasoning traces and distill a small set of *high-value* "advantage experiences" that most effectively improve the LLM when used

as few-shot demonstrations.

Trajectory Sampling. Given a mini-batch of N training questions and a group size g, we sample G interaction traces per question using stochastic decoding, resulting in $N \times G$ traces in total. For a question Q_i, the sampled group is denoted as $\mathcal{O}_i = \{O_{i,1}, \ldots, O_{i,G}\}$, where each $O_{i,j}$ contains the full tool-interaction transcript and a final predicted answer.

Rule-Based Reward. We assign each trace a binary rule reward $R_{i,j} \in \{0, 1\}$ by exact match:

$$R_{i,j} = \mathbf{1}[\hat{y}_{i,j} = y_i^*],$$

where $\hat{y}_{i,j}$ is the answer extracted from $O_{i,j}$ and y_i^* is the gold answer for Q_i. We retain only successful traces $\mathcal{O}_i^+ = \{O_{i,j} \in \mathcal{O}_i \mid R_{i,j} = 1\}$.

LLM-Guided Group Computation. Among correct traces $\mathcal{O}_i^+$, demonstrations vary in usefulness. We thus let the LLM rank them by *marginal instructional value* (i.e., how much they provide new, "aha"-style guidance beyond what the model already knows). The top-ranked traces are distilled into *advantage experience texts* $\{A_{i,1}, \ldots, A_{i,K}\}$.

Experience Library Rule-Controller. We then validate each advantage candidate by measuring the validation-set gain after adding it to the current library. With a fixed library budget of K shots, we keep the K experiences that yield the largest validation improvements (from both existing and newly mined candidates), forming the final library $\mathcal{D}_{demo}$ for test-time inference.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate AT2QA on two challenging TKGQA benchmarks: MultiTQ (Chen et al., 2023) and TimelineKGQA (Sun et al., 2025), which comprises Timeline-CronQuestion and Timeline-ICEWS-Actor. These datasets collectively assess diverse reasoning capabilities, spanning multi-granular timestamp constraints, chronological event tracking, and actor-centric temporal dynamics. Detailed statistics and metrics are deferred to Appendix A.

Baselines. We compare AT2QA against strong baselines from two paradigms of TKGQA: **(1) TKG embedding-based methods**, including EmbedKGQA (Jin et al., 2023), CronKGQA (Saxena et al., 2021), and MultiQA (Chen et al., 2023); and **(2) LLM-based methods**, including prompt-based workflows such as ARI (Chen et al., 2024b), TempAgent (Hu et al., 2025), MemoTime (Tan et al., 2025), and RTQA (Gong et al., 2025), as well as training-based approaches such as Search-R1 (Jin et al., 2025), TimeR^4 (Qian et al., 2024), PoK (Qian et al., 2025), and Temp-R1 (Gong et al., 2026). These baselines cover both traditional TKGQA systems and recent LLM-centered paradigms, providing a comprehensive comparison against prior state-of-the-art methods. Following standard practice in TKGQA, we report the **Hits@1** (Exact Match) metric for all methods.

5.2 Implementation Details

AT2QA uses DeepSeek-V3.2 with the server default decoding configuration (temperature = 1.0). All facts are embedded offline by GLM-Embedding-3 (256-d), and the search tool performs brute-force cosine-similarity retrieval over structurally filtered candidates, returning at most 10 facts per call. We cap the maximum interaction rounds at $T_{\max} = 20$ and use a fixed library of $K = 3$ demonstrations for training-free optimization. For ablation, we reimplemented RTQA in an OpenAI-compatible tool-calling form while preserving its rigid workflow, and replaced its original retriever with our advanced search tool to obtain the "RTQA + advanced tool" variant, enabling controlled comparison under the same tool interface. At official API rates, AT2QA costs under $150 on MultiTQ and under $50 on TimelineKGQA, while training-free GRPO costs under $7 per dataset..

5.3 Main Results

The main results on MultiTQ are shown in Table 2. AT2QA achieves a new state-of-the-art with an overall accuracy of **88.7%**, outperforming the previous best model, Temp-R1 (78.0%), by **10.7** points. The advantage is most pronounced on **multiple**-answer questions, where AT2QA reaches **75.1%**, exceeding the previous best result (55.0%) by **20.1** points, which highlights its strength in exhaustive temporal multi-hop reasoning. AT2QA also achieves the best performance on **entity** answers (86.4%) and remains competitive on **time** answers (94.5%), falling only slightly below Temp-R1. Overall, these results show that AT2QA is highly effective for complex TKGQA.

To further evaluate generalization, we test AT2QA on TimelineKGQA-CronQuestion and

Method	TimelineKGQA-CronQuestion				TimelineKGQA-ICEWS-Actor			
	Overall	Simple	Medium	Complex	Overall	Simple	Medium	Complex
RAG Baseline	0.235	0.704	0.092	0.009	0.265	0.660	0.128	0.011
LLaMA2-7B	0.169	0.049	0.143	0.282	0.111	0.035	0.066	0.322
GPT-4o	0.206	0.069	0.130	0.376	0.113	0.051	0.035	0.353
RTQA	0.298	0.608	0.218	0.135	–	–	–	–
PoK	0.651	0.737	0.539	0.683	0.602	0.744	0.456	0.578
Temp-R1	0.705	**0.960**	0.486	0.672	0.642	**0.866**	0.388	0.595
AT2QA	**0.754**	0.831	**0.631**	**0.803**	**0.754**	0.859	**0.627**	**0.768**

Table 1: Performance comparison on TimelineKGQA-CronQuestion and TimelineKGQA-ICEWS-Actor datasets. **Bold** indicates the best performance, and underline indicates the second best.

Method	Overall	Question Type		Answer Type	
		multiple	single	entity	time
TKG Embedding-based Methods					
EmbedKGQA	0.206	0.134	0.235	0.290	0.001
CronKGQA	0.279	0.134	0.337	0.328	0.156
MultiQA	0.293	0.159	0.347	0.349	0.157
LLM-based Static Workflows					
Search-R1	0.352	0.094	0.474	0.230	0.705
ARI	0.380	0.210	0.680	0.394	0.344
TempAgent	0.702	0.316	0.857	0.624	0.870
TimeR[4]	0.728	0.335	0.887	0.639	0.945
MemoTime	0.730	0.459	0.829	0.677	0.846
RTQA	0.765	0.424	0.902	0.692	0.942
PoK	0.779	0.409	0.929	0.696	0.962
Temp-R1	0.780	0.550	0.888	0.714	**0.969**
Ours (Autonomous Training-free Agent)					
AT2QA	**0.887**	**0.751**	**0.942**	**0.864**	0.945

Table 2: Performance comparison on the MultiTQ test set.**Bold** indicates the best performance, and underline indicates the second best.

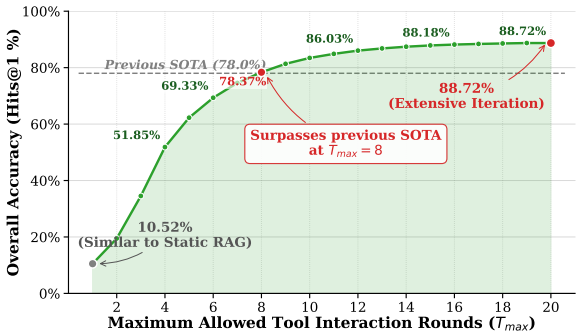

Figure 5: Effect of the maximum interaction rounds $T_{\max}$ on MultiTQ. AT2QA reaches most of its gain with a moderate interaction budget, rather than relying on very deep search.

TimelineKGQA-ICEWS-Actor (Table 1). AT2QA achieves the best overall accuracy on both datasets, reaching **75.4%** on each and surpassing Temp-R1 by **4.9** and **11.2** points, respectively. Its gains are concentrated on the **medium** and **complex** subsets, where it consistently outperforms prior methods, while Temp-R1 remains slightly stronger on **simple** questions; part of this apparent gap is due to benchmark undercounting issues (Appendix C). This suggests that AT2QA is particularly effective on harder TKGQA cases requiring adaptive search, multi-step evidence accumulation, and dynamic reasoning.

5.4 Ablation Study

We conduct ablations to answer four questions: (1) whether AT2QA improves simply by allowing more interaction rounds, (2) whether the main gains come from better tools or from autonomy, (3) how much training-free GRPO-selected few-shot demonstrations contribute, and (4) how sensitive the framework is to retrieval depth and backbone choice.

5.4.1 Effect of the Interaction Budget

A natural concern is that the improvement comes from allowing many search attempts. Figure 5 shows that this is not the case. Performance increases rapidly as the maximum interaction budget grows, indicating that multi-turn interaction is important; however, the gain does not rely on very deep search. AT2QA already surpasses previous SOTA at $T_{\max} = 8$, after which the curve shows clear diminishing returns, and the best result is reached at around 19 rounds. We therefore use $T_{\max} = 20$ in the final system.

5.4.2 Main Sources of Improvement

Table 3 isolates the contributions of the tool, the reasoning paradigm, and the training-free GRPO-selected few-shot demonstrations. Under the rigid RTQA workflow, replacing the base tool with our advanced tool brings only a marginal gain from

Paradigm	Configuration	Hits@1
LLM only	DeepSeek V3.2 without search tool	0.100
Rigid workflow (RTQA)	+ Base tool (semantic relevance)	0.762
	+ Advanced tool (all components)	0.767
Autonomous agent (zero-shot)	+ Base tool (semantic relevance)	0.791
	+ Base tool + time limit	0.817
	+ Base tool + time-aware sorting	0.814
	+ Base tool + entity filter	0.795
	+ Base tool + relation filter	0.799
	+ Advanced tool (all components)	0.844
Autonomous agent+ training free GRPO	Advanced tool + selected few-shot demonstrations	**0.887**

Table 3: Ablation of the main performance drivers. For the autonomous-agent rows, the four single-component settings are variants built on top of the base tool, while "Advanced tool" combines all tool components.

76.2% to 76.7%, showing that the tool itself is not the main source of improvement. With the same advanced tool, our autonomous agent reaches 84.4% in the zero-shot setting, a **+7.7 point gain** over rigid RTQA, indicating **that the major gain comes from autonomy**. On top of that, adding the GRPO-selected few-shot demonstrations further improves performance from **84.4% to 88.7% (+4.3)**, showing that **training-free GRPO provides another substantial boost**.

We also ablate the components of the advanced tool in the zero-shot autonomous setting. Temporal modules contribute the most: time limit and time-aware sorting improve accuracy to 81.7% and 81.4%, while entity and relation filtering yield smaller gains (79.5% and 79.9%). Combining all components reaches 84.4%, suggesting that **the advanced tool offers auxiliary gains**.

5.4.3 Effect of Retrieval Depth

Table 4 shows the effect of retrieval depth on a random 700-question subset. Increasing depth from Top-10 to Top-30 yields only a marginal 0.8% improvement. Balancing this limited gain against token budgets, we adopt Top-10 for the main experiments. Note that subset evaluation may cause slight deviations from our main results. We also ablate the embedding models in Appendix 9.

5.4.4 Backbone Generality

Finally, we evaluate the zero-shot autonomous framework with different backbone LLMs. As shown in Table 5, it remains effective across all tested backbones, including Qwen3-Max (79.3%), Kimi-2.5 (78.9%), DeepSeek-R1 (84.2%), and DeepSeek V3.2 (84.4%). Even in the zero-shot setting, all variants surpass the previous state-of-the-art (78.0%), indicating that the framework generalizes well across backbone models.

Retrieval Depth	Hits@1
Top-10	0.898
Top-15	0.898
Top-20	0.904
Top-25	0.904
Top-30	0.906

Table 4: Effect of retrieval depth.

Backbone LLM (Zero-shot)	Hits@1
Qwen3-Max	0.793
DeepSeek-R1	0.842
Kimi-2.5	0.789
DeepSeek V3.2	**0.844**

Table 5: Zero-shot performance across different LLMs.

5.5 Analysis of Interpretability

AT2QA provides a complete evidence chain for its prediction process, including retrieval, filtering, fact selection, and answer determination, rather than exposing only the final answer. Figures 7, 8, and 9 present representative case studies, where the highlighted facts show the key supporting evidence. Beyond this, Appendix B provides additional supporting evidence that autonomy brings extra benefits by inducing self-correction and self-validation, thereby fully unlocking the model's capabilities.

6 Conclusion

In this paper, we introduced **AT2QA**, an autonomy-first and training-free agentic framework for temporal knowledge graph question answering. Departing from fixed, human-designed workflows and expensive fine-tuning pipelines, AT2QA enables the LLM to determine *which actions are necessary next* while iteratively interacting with the TKG environment through a generic search tool. This design inherently allows the agent to dynamicly verify retrieved evidence, reformulate queries, and self-correct its reasoning trajectory when errors or contradictions arise. To further elicit complex temporal reasoning without gradient updates, we proposed a training-free experience mining strategy that distills a compact few-shot library from successful self-generated trajectories. Experiments on three challenging benchmarks show that AT2QA achieves state-of-the-art results, while also producing transparent and verifiable reasoning traces.

Limitations

Efficiency and scalability. Our implementation performs nearest-neighbor retrieval over a structurally filtered candidate set and allows up to $T_{\max} = 20$ interaction turns. While this design improves robustness, it increases latency and inference cost compared to single-pass RAG. Scaling to substantially larger graphs or tighter latency budgets may require more efficient indexing (e.g., ANN) and better turn-level early stopping policies.

Autonomy can be inefficient and unstable. Granting the LLM full autonomy improves robustness, but it may also lead to *extra exploration turns* or occasional looping behaviors under ambiguous queries or strong distractors. As a result, latency and inference cost can be higher than single-pass RAG or fixed pipelines, and performance may be more sensitive to decoding randomness and stopping criteria.

References

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. Temporal knowledge graph completion: A survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6545–6553.

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. 2025. Training-free group relative policy optimization. *Preprint*, arXiv:2510.08191.

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, Torino, Italia. ELRA and ICCL.

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, Bangkok, Thailand. Association for Computational Linguistics.

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.

Wentao Ding, Hao Chen, Huayu Li, and Yuzhong Qu. 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*, pages 1867–1877, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 (EMNLP)*, pages 9853–9870.

Zhaoyan Gong, Zhiqiang Liu, Songze Li, Xiaoke Guo, Yuanxiang Liu, Xinle Deng, Zhizhen Liu, Lei Liang, Huajun Chen, and Wen Zhang. 2026. Temp-r1: A unified autonomous agent for complex temporal kgqa via reverse curriculum reinforcement learning. *Preprint*, arXiv:2601.18296.

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, Albuquerque, New Mexico. 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*, pages 1807–1810.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 792–802.

Zihao Jiang, Miao Peng, Zhenyan Shan, Wenjie Xu, Ben Liu, Gong Chen, Ziqi Gao, and Min Peng. 2026. Tkg-thinker: Towards dynamic reasoning over temporal knowledge graphs via agentic reinforcement learning. *Preprint*, arXiv:2602.05818.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

Weiqiang Jin, Biao Zhao, Hang Yu, Xi Tao, Ruiping Yin, and Guizhong Liu. 2023. Improving embedded knowledge graph multi-hop question answering by introducing relational chain reasoning: W. jin et al. *Data Mining and Knowledge Discovery*, 37(1):255–288.

Yonghao Liu, Di Liang, Mengyu Li, Fausto Giunchiglia, Ximing Li, Sirui Wang, Wei Wu, Lan Huang, Xiaoyue Feng, and Renchu Guan. 2023. Local and global: Temporal question answering via information fusion. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5141–5149. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xufei Lv, Kehai Chen, Haoyuan Sun, Xuefeng Bai, Min Zhang, Houde Liu, and Kehai Chen. 2025. The hidden link between rlhf and contrastive learning. *Preprint*, arXiv:2506.22578.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Adesoji Bello, Hassan Ghaffari, Santosh K. Srivastava, and George Karypis. 2022. TempoQR: Temporal question reasoning over knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Costas Mavromatis, Prasanna Lakkur Subramanyam, and George Karypis. 2023. TwiRGCN: Temporally weighted graph convolution for question answering over temporal knowledge graphs. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2049–2060. Association for Computational Linguistics.

Sumit Neelam, Udit Sharma, Hima Karanam, Shajith Ikbal, Pavan Kapanipathi, Ibrahim Abdelaziz, Nandana Mihindukulasooriya, Young-Suk Lee, Santosh Srivastava, Cezar Pendus, Saswati Dana, Dinesh Garg, Achille Fokoue, G P Shrivatsa Bhargav, Dinesh Khandelwal, Srinivas Ravishankar, Sairam Gurajada, Maria Chang, Rosario Uceda-Sosa, and 6 others. 2022. SYGMA: A system for generalizable and modular question answering over knowledge bases. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3866–3879, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Miao Peng, Nuo Chen, Zongrui Suo, and Jia Li. 2025. Rewarding graph reasoning process makes LLMs more generalized reasoners. In *Proceedings of the 31st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2257–2268. ACM.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, and Xiaojie Yuan. 2025. Plan of knowledge: Retrieval-augmented large language models for temporal knowledge graph question answering. *Preprint*, arXiv:2511.04072.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. 2024. TimeR^4 : 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, Miami, Florida, USA. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 6663–6676, Online. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Hakim Sidahmed, Samrat Phatale, Alex Hutcheson, Zhuonan Lin, Zhang Chen, Zac Yu, Jarvis Jin, Simral Chaudhary, Roman Komarytsia, Christiane Ahlheim, Yonghao Zhu, Bowen Li, Saravanan Ganesh, Bill Byrne, Jessica Hoffmann, Hassan Mansoor, Wei Li, Abhinav Rastogi, and Lucas Dixon. 2024. Parameter efficient reinforcement learning from human feedback. *Preprint*, arXiv:2403.10704.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Qiang Sun, Sirui Li, Du Huynh, Mark Reynolds, and Wei Liu. 2025. Timelinekgqa: A comprehensive question-answer pair generator for temporal knowledge graphs. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 797–800, New York, NY, USA. Association for Computing Machinery.

Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. 2025. Memotime: Memory-augmented temporal knowledge graph enhanced large language model reasoning. *Preprint*, arXiv:2510.13614.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Preprint*, arXiv:2504.13837.

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**

Table 6: Statistics of question categories in MultiTQ.

Category	Train	Dev	Test
Simple	7,200	2,400	2,400
Medium	8,252	2,751	2,751
Complex	9,580	3,193	3,193
Total	**25,032**	**8,344**	**8,344**

Table 7: Statistics of question categories in Timeline-CronQuestion.

Category	Train	Dev	Test
Simple	17,982	5,994	5,994
Medium	15,990	5,330	5,330
Complex	19,652	6,550	6,550
Total	**53,624**	**17,874**	**17,874**

Table 8: Statistics of question categories in Timeline-ICEWS-Actor.

A Dataset Details

MultiTQ. Constructed from the ICEWS05-15 dataset, MultiTQ is a large-scale benchmark for multi-granularity temporal question answering. It contains approximately 500K question-answer pairs and more than 461K temporal facts represented as quadruples. The dataset spans multiple temporal granularities, including years, months, and days, and requires models to handle diverse temporal reasoning patterns under explicit or implicit constraints. Following the original benchmark design, questions are organized into six categories: *Equal*, *Before/After*, and *First/Last* under the *Single* setting, as well as *Equal Multi*, *After First*, and *Before Last* under the *Multiple* setting. These categories cover direct temporal retrieval, comparative reasoning, and multi-hop temporal inference over multiple entities. Owing to its large scale, diverse temporal granularity, and strong compositionality, MultiTQ provides a challenging testbed for evaluating AT2QA's autonomous search, temporal grounding, and consistency checking abilities. The detailed statistics are summarized in Table 6.

Timeline-CronQuestion. Derived from the CronQuestion knowledge graph, Timeline-CronQuestion is a TimelineKGQA benchmark designed for temporal question answering over time-point-centric knowledge graphs. It contains 41,720 question-answer pairs. Compared with MultiTQ, this benchmark places stronger emphasis on timeline-centric temporal reasoning, especially temporal arithmetic and semantic operations over intervals. In particular, models must handle duration reasoning, interval composition, and set-like operations over temporal spans, with answers extending beyond entities and timestamps to include time ranges or durations. Following the TimelineKGQA taxonomy, questions are grouped into three difficulty levels: *Simple*, *Medium*, and *Complex*, corresponding to reasoning over one, two, and multiple context events, respectively. This balanced difficulty structure makes Timeline-CronQuestion well suited for evaluating whether AT2QA can generalize from direct temporal retrieval to compositional temporal inference. The detailed statistics are summarized in Table 7.

Timeline-ICEWS-Actor. Constructed from the ICEWS Coded Event Data, Timeline-ICEWS-Actor is a TimelineKGQA benchmark for actor-centric temporal question answering over dynamic event sequences. It contains 89,372 question-answer pairs. In contrast to Timeline-CronQuestion, this dataset is grounded in event-centric political interactions and focuses more directly on reasoning about actors, event positions, and temporally evolving relations in a dynamic timeline. Questions are likewise divided into three difficulty levels—*Simple*, *Medium*, and *Complex*—which require progressively more challenging temporal reasoning over one, two, and multiple events. Its relatively balanced distribution across difficulty levels, together with its event-centric structure, makes Timeline-ICEWS-Actor a valuable benchmark for assessing AT2QA's robustness in multi-step timeline reasoning and actor-focused evidence aggregation. The detailed statistics are summarized in Table 8.

B Quantitative Analysis of Agentic Behaviors

A fundamental limitation of static LLM workflows is their inability to recover from intermediate retrieval errors. To prove that our massive

performance gain on multi-hop questions stems from genuine agentic behaviors—specifically *Self-Correction* and *Self-Validation*—we conduct a quantitative trajectory analysis.

We define a *Gold Fact* as the specific retrieved evidence required to deduce the correct answer. We track the relative position (in percentage) of the **first appearance of the Gold Fact** within the agent's entire interaction trajectory. Figure 6 plots the Cumulative Density Function (CDF) of this metric for all successfully answered *Multiple-target* questions that involved more than 3 reasoning rounds.

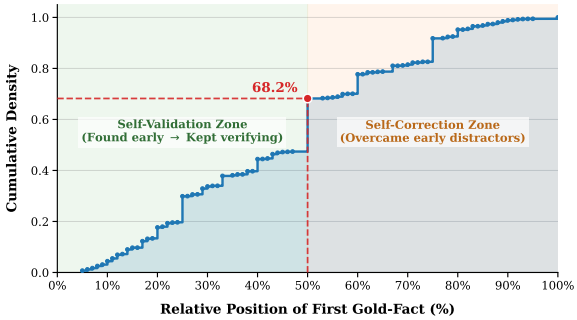

Figure 6: Cumulative Density of First Gold-Fact Position for complex multiple-target queries. The distribution provides quantitative proof of the agent's self-validation (left) and self-correction (right) capabilities.

The CDF curve reveals two distinct and profound agentic behaviors:

Self-Correction (The Right Tail). Remarkably, in approximately 32% of the successful cases, the very first Gold Fact does not appear until the *latter half* (>50%) of the total reasoning rounds. This indicates that during the early stages of navigation, the agent frequently retrieved irrelevant, distracting, or contradictory facts. Instead of hallucinating an answer based on this noisy context, the agent autonomously recognized the failure, adjusted its search constraints (e.g., modifying temporal windows or entity roles), and repeatedly tried until the correct evidence was surfaced. This confirms a robust **Self-Correction** mechanism.

Self-Validation (The Left Tail). Conversely, in cases where the Gold Fact was discovered early in the trajectory (<50%), the agent *did not* immediately terminate the session. Since the questions demand multiple answers, premature termination would lead to partial failures. The log shows that the agent retained the initial Gold Fact in its memory, recognized that the evidence was insufficient to holistically answer the query, and deliberately continued searching to verify and gather the remaining pieces. This **Self-Validation** behavior demonstrates a high degree of meta-cognitive planning, sharply contrasting with naive RAG pipelines that stop after a single semantic match.

Backbone Embedding Model (Zero-shot)	Hits@1
text-embedding-3-large	0.883
text-embedding-3-small	0.892
gemini-embedding-001	0.889
Baidu-Embedding-V1	0.877

Table 9: Performance across different embedding.

B.1 Impact of Backbone Embedding Models

To investigate whether our framework's performance heavily relies on the semantic matching capabilities of a specific dense retriever, we conducted an ablation study evaluating the impact of different backbone embedding models. To ensure a fair and efficient comparison, this evaluation was performed on a fixed subset of 500 questions randomly sampled from the test set.

We substituted the default embedding model with several leading alternatives, including OpenAI's text-embedding-3-large and text-embedding-3-small, Google's gemini-embedding-001, and Baidu-Embedding-V1. All other configurations, including the LLM agent and the maximum tool interaction rounds, remained identical.

As shown in Table 9, the choice of embedding model has a negligible impact on the final performance. The Hits@1 scores remain highly stable across all tested models, fluctuating tightly within a narrow margin of 1.5% (from 0.877 to 0.892). Notably, lighter models such as text-embedding-3-small perform on par with, or even slightly surpass, heavier models. This suggests that the minor 1.5% variance is primarily attributable to random fluctuations

C Additional Analysis on Single-Gold Undercounting in Timeline-CronQuestion

During error analysis, we found that a substantial portion of AT2QA's officially counted errors on Timeline-CronQuestion are factually correct predictions that are penalized by the benchmark's single-gold exact-match evaluation. In particular, some questions admit multiple valid answers in the underlying temporal knowledge graph, while the dataset

QLevel	Reason	Example	AT2QA Answer Relevant Facts
Simple	insufficient gold (91.4%)	**Question:** "Burnley F.C. is member of sports team by who from 1960-01-01 to 1968-01-01?" **Gold:** Willie Morgan **AT2QA:** Andy Lochhead	**Search fact:** 301957| Andy Lochhead|member of sports team|Burnley F.C.|1960-01-01| 1968-01-01
Medium	insufficient gold (23%)	**Question:** "Cell 211 nominated for which object after Romeo Menti member of sports team A.C. Milan?" **Gold:** Goya Award for Best Film **AT2QA:** Goya Award for Best Producer	**Search fact:** 120994| Cell 211|nominated for| Goya Award for Best Producer|2010-01-01| 2010-01-01 289197|Romeo Menti| member of sports team| A.C. Milan|1944-01-01| 1944-01-01

Table 10: Representative examples of answer undercounting in Timeline-CronQuestion.

keeps only one of them as the annotated gold answer. As a result, a prediction can be correct with respect to the question and graph facts, yet still be counted as incorrect if it does not exactly match the single annotated answer.

We manually audited AT2QA's officially incorrect predictions on the test set. For the **Simple** split, 371 out of 406 official errors (91.4%) were found to be factually correct answers excluded by the single-gold annotation. For the **Medium** split, the same issue was observed in 226 out of 1,012 official errors (23.0%). We also observed a smaller subset of cases in which the annotated gold answer itself appears to be incorrect. Representative examples are shown in Table 10.

A likely reason is the question construction process of Timeline-CronQuestion: questions are generated from a fixed subset of sampled temporal facts, while other graph facts that also satisfy the same constraints may be omitted from the final answer annotation. This can lead to under-specified answer sets and consequently undercount factual correctness under exact-match evaluation.

These findings suggest that the official exact-match score may underestimate AT2QA's factual correctness on Timeline-CronQuestion, especially on the Simple and Medium splits. Nevertheless, all main results in this paper are reported strictly under the official benchmark protocol; this analysis is intended only as a diagnostic supplement rather than a replacement for the standard evaluation.

D Case Study

To provide a deeper understanding of how AT2QA autonomously navigates TKGs to execute complex temporal reasoning, we present a qualitative analysis of several representative cases. These studies intuitively demonstrate the agent's robust dynamic self-correction and strict consistency checking capabilities, all achieved without any parameter updates.

Figure 7 illustrates the reasoning trajectory for an "Equal Multi" question from the MultiTQ dataset. This example highlights AT2QA's proficiency in accurately parsing implicit temporal constraints and leveraging the time window of a pivot event to effectively bound the search space for concurrent multi-entity activities. Figures 8 and 9 detail the reasoning chains for the highly challenging "Before Last" and "After First" question types, respectively. In these scenarios, AT2QA explicitly grounds the anchor event first, and subsequently conducts fine-grained evaluations of subsequent events within strictly defined temporal boundaries. Figure 8 illustrates AT2QA's consistency checking mechanism. Upon retrieving a highly plausible candidate answer (i.e., "Ministry (International)"), the agent resists a greedy acceptance strategy. Instead, it proactively initiates a more granular internal verification over the specified time slice to guarantee that no subsequent entities visited France within the window, thereby ensuring the global optimality of the final conclusion. Furthermore, Figure 9 demonstrates dynamic self-correction capability of AT2QA. Upon detecting that the current retrieval strategy fails to recall the requisite evidence, the agent autonomously rolls back its reasoning state and dynamically rewrites its query strategy until the crucial evidence is successfully pinpointed. This closed-loop correction mechanism effectively breaks the bottleneck of cascading error amplification inherent in traditional rigid workflows.

Question: Who praised the United Arab Emirates in the same month as Iran?
Gold Answer: Muhammad VI

<think> I'll search for documents where someone praised the United Arab Emirates and Iran in the same month. </think>
<search> praised United Arab Emirates Iran same month sort=relevance </search>
<information>
Doc 1: (line 194760, score 0.901) Iran Praise_or_endorse United_Arab_Emirates 2009-04-04
Doc 2: (line 131985, score 0.894) United_Arab_Emirates Praise_or_endorse Iran 2007-11-19
Doc 3: (line 266050, score 0.888) United_Arab_Emirates Praise_or_endorse Iran 2010-12-16
Doc 4: (line 292641, score 0.886) United_Arab_Emirates Praise_or_endorse Iran 2011-10-04
Doc 5: (line 369629, score 0.882) United_Arab_Emirates Praise_or_endorse Iran 2013-11-24
Doc 6: (line 406681, score 0.879) United_Arab_Emirates Praise_or_endorse Iran 2014-10-20
Doc 7: (line 248731, score 0.879) United_Arab_Emirates Praise_or_endorse Iran 2010-07-16
Doc 8: (line 302985, score 0.879) United_Arab_Emirates Praise_or_endorse Iran 2012-01-19
Doc 9: (line 141999, score 0.876) United_Arab_Emirates Make_a_visit Iran 2008-02-18
Doc 10: (line 371290, score 0.876) United_Arab_Emirates Praise_or_endorse Iran 2013-12-09
</information>

<think> I found several documents showing praise between Iran and the United Arab Emirates. Let me search more specifically for entities that praised the United Arab Emirates in the same month as Iran. </think>
<search> praise United Arab Emirates same month sort=relevance entity=United Arab Emirates pos=back rel=Praise_or_endorse </search>
<information>
Doc 1: (line 197020, score 0.872) Muhammad_VI Praise_or_endorse United_Arab_Emirates 2009-04-16
Doc 2: (line 37755, score 0.859) Ministry_(Morocco) Praise_or_endorse United_Arab_Emirates 2005-11-14
Doc 3: (line 188048, score 0.856) Vietnam Praise_or_endorse United_Arab_Emirates 2009-02-18
Doc 4: (line 239322, score 0.855) Ali_Abdullah_Saleh Praise_or_endorse United_Arab_Emirates 2010-04-26
Doc 5: (line 187893, score 0.855) Vietnam Praise_or_endorse United_Arab_Emirates 2009-02-17
Doc 6: (line 239266, score 0.852) Japan Praise_or_endorse United_Arab_Emirates 2010-04-25
Doc 7: (line 100214, score 0.851) Sheikh_Ahmed_Abdullah Praise_or_endorse United_Arab_Emirates 2007-03-12
Doc 8: (line 104261, score 0.851) China Praise_or_endorse United_Arab_Emirates 2007-04-10
Doc 9: (line 94846, score 0.850) China Praise_or_endorse United_Arab_Emirates 2007-01-30
Doc 10: (line 362603, score 0.850) Kuwait Praise_or_endorse United_Arab_Emirates 2013-09-13
</information>

<think> Now let me check when Iran praised the United Arab Emirates to find the specific month, and then see who else praised the UAE in that same month. From the first search result, I see Iran praised the United Arab Emirates on 2009-04-04. Let me check who else praised the UAE in April 2009. </think>
<search> praise United Arab Emirates April 2009 sort=relevance start=2009-04-01 end=2009-04-30 entity=United Arab Emirates pos=back rel=Praise_or_endorse </search>
<information>
Doc 1: (line 197020, score 0.886) Muhammad_VI Praise_or_endorse United_Arab_Emirates 2009-04-16
Doc 2: (line 194760, score 0.859) Iran Praise_or_endorse United_Arab_Emirates 2009-04-04
</information>

<think> Perfect! I found that in April 2009: 1. Iran praised the United Arab Emirates on April 4, 2009. 2. Muhammad VI praised the United Arab Emirates on April 16, 2009. Therefore, Muhammad VI praised the United Arab Emirates in the same month (April 2009) as Iran. </think>
<answer> Muhammad VI </answer>

Figure 7: Case Study for Equal Multi Questions.

Question: Which country did France want to meet last, before the Belgian foreign minister?
Gold Answer: Ministry (International)

<think> I'll search for information about France wanting to meet with the Belgian foreign minister first, then find what country France wanted to meet with last before that. </think>
<search> France Express intent to meet or negotiate Belgian foreign minister rel=Express_intent_to_meet_or_negotiate </search>
<information>
Doc 1: (line 32919, score 0.943) France Express_intent_to_meet_or_negotiate Foreign_Affairs_(Belgium) 2012-10-04
Doc 2: (line 415922, score 0.918) France Express_intent_to_meet_or_negotiate Head_of_Government_(Germany) 2015-01-05
Doc 3: (line 174708, score 0.913) France Express_intent_to_meet_or_negotiate Foreign_Affairs_(France) 2008-11-03
……
Doc 10: (line 29595, score 0.906) France Express_intent_to_meet_or_negotiate Representatives_(France) 2005-09-12
</information>

<think> I found that France expressed intent to meet with Foreign Affairs (Belgium) on October 4, 2012. Now I need to find what country France wanted to meet with last before this date. Let me search for France expressing intent to meet with other entities before October 4, 2012.</think>
<search> France Express_intent_to_meet_or_negotiate sort=time_desc end=2012-10-03 entity=France pos=front rel=Express_intent_to_meet_or_negotiate </search>
<information>
Doc 1: (line 325463, score 0.935) France Express_intent_to_meet_or_negotiate Ministry_(International) 2012-08-30
Doc 2: (line 295819, score 0.895) France Express_intent_to_meet_or_negotiate South_Africa 2011-11-07
Doc 3: (line 26001, score 0.899) France Express_intent_to_meet_or_negotiate Representatives_(Vietnam) 2010-10-26
……
Doc 10: (line 175043, score 0.913) France Express_intent_to_meet_or_negotiate North_America 2008-11-05
</information>

<think> I can see that France expressed intent to meet with "Ministry (International)" on August 30, 2012, which is the most recent before October 4, 2012. However, I need to check if there are any other meetings between August 30 and October 4, 2012. Let me search more specifically for that time period. </think>
<search> France Express_intent_to_meet_or_negotiate sort=time_desc start=2012-08-31 end=2012-10-03 entity=France pos=front rel=Express_intent_to_meet_or_negotiate </search>
<information>
</information>

Consistency Checking: Initiating self-verification to ensure optimality.

<think> There are no "Express_intent_to_meet_or_negotiate" events between August 30 and October 3, 2012. So the last country France wanted to meet before the Belgian foreign minister (Foreign Affairs (Belgium)) on October 4, 2012 was "Ministry (International)" on August 30, 2012. Now let me check more specifically for the exact line about France expressing intent to meet with Ministry (International). </think>
<search> France Express_intent_to_meet_or_negotiate Ministry (International) 2012-08-30 sort=relevance
</search>
<information>
Doc 1: (line 325463, score 0.966) France Express_intent_to_meet_or_negotiate Ministry_(International) 2012-08-30
Doc 2: (line 360778, score 0.956) France Express_intent_to_meet_or_negotiate Foreign_Affairs_(France) 2013-08-26
Doc 3: (line 174708, score 0.949) France Express_intent_to_meet_or_negotiate Foreign_Affairs_(France) 2008-11-03
……
Doc 10: (line 12618, score 0.936) Foreign_Affairs_(France) Express_intent_to_meet_or_negotiate France 2007-10-10
</information>

<think> Based on the evidence, France expressed intent to meet with Ministry (International) on August 30, 2012, which was the last such expression before meeting with Foreign Affairs (Belgium) on October 4, 2012. </think>
<answer> Ministry (International) </answer>

Figure 8: Case Study for Before Last Questions with Consistency Checking.

Question: Who was the first to visit France after Stephen Williams?
Gold: Andreas Lubitz

<think> I'll help you find who visited France after Stephen Williams. Let me search for information about Stephen Williams visiting France first. </think>
<search> Stephen Williams visited France sort=relevance </search>
<information>
Doc 1: (line 459595, score 0.923) Stephen_Williams Make_a_visit France 2015-12-11
Doc 2: (line 459539, score 0.894) France Host_a_visit Stephen_Williams 2015-12-11
Doc 3: (line 407836, score 0.790) Thomas_Dandois Make_a_visit France 2014-10-29
Doc 4: (line 163482, score 0.785) Brice_Hortefeux Make_a_visit France 2008-07-29
Doc 5: (line 168219, score 0.782) Brice_Hortefeux Make_a_visit France 2008-09-08
Doc 6: (line 163514, score 0.779) France Host_a_visit Brice_Hortefeux 2008-07-29
Doc 7: (line 131598, score 0.778) Seyoum_Mesfin Make_a_visit France 2007-11-16
Doc 8: (line 34086, score 0.778) Jack_Straw Make_a_visit France 2005-10-18
Doc 9: (line 423911, score 0.777) Staffan_de_Mistura Make_a_visit France 2015-03-05
Doc 10: (line 423592, score 0.776) Staffan_de_Mistura Make_a_visit France 2015-03-03
</information>

<think> Now I can see that Stephen Williams visited France on December 11, 2015. Let me search for visits to France after this date to find who visited next. </think>
<search> Make a visit France sort=time_asc start=2015-12-12 </search>
<information>
Doc 1: (line 459624, score 0.736) Foreign_Affairs_(France) Make_statement France 2015-12-12
Doc 2: (line 459672, score 0.720) Domestic_Affairs_(Vietnam) Make_a_visit China 2015-12-13
Doc 3: (line 459699, score 0.718) Thailand Consult France 2015-12-13
Doc 4: (line 459674, score 0.708) France Consult Thailand 2015-12-13
Doc 5: (line 459682, score 0.666) Member_of_Parliament_(France) Make_statement Government_(France) 2015-12-13
Doc 6: (line 459773, score 0.709) France Consult Education_Ministry_(France) 2015-12-14
Doc 7: (line 459852, score 0.685) Refugee_(Afghanistan) Make_a_visit Iran 2015-12-14
Doc 8: (line 459722, score 0.684) Barack_Obama Make_statement France 2015-12-14
Doc 9: (line 459755, score 0.680) Education_Ministry_(France) Consult France 2015-12-14
Doc 10: (line 459768, score 0.678) Foreign_Affairs_(Congo) Make_a_visit Angola 2015-12-14
</information>

Self-Correction: Recognizing missing info and autonomously adjusting strategy.

<think> The search results don't show any visits to France immediately after December 11, 2015. Let me expand the search to find the next visitor to France after Stephen Williams. </think>
<search> Make a visit France sort=time_asc start=2015-12-11 entity=France pos=back rel=Make_a_visit </search>
<information>
Doc 1: (line 459595, score 0.766) Stephen_Williams Make_a_visit France 2015-12-11
Doc 2: (line 460813, score 0.726) Andreas_Lubitz Make_a_visit France 2015-12-26
</information>

<think> Great! I found that Andreas Lubitz visited France on December 26, 2015, which is after Stephen Williams' visit on December 11, 2015. Let me verify this is the next visitor by checking if there are any visits between these dates. </think>
<search> Make a visit France sort=time_asc start=2015-12-11 entity=France pos=back rel=Make_a_visit </search>
<information>
</information>

<think> Perfect! There are no visits to France between December 12 and December 25, 2015. Therefore, the first person to visit France after Stephen Williams was Andreas Lubitz. </think>
<answer> Andreas Lubitz </answer>

Figure 9: Case Study for After First Questions with Self-Correction.