Reinforcement Learning Enhanced Multi-hop Reasoning for Temporal Knowledge Question Answering

Wuzhenghong Wen[1]*, Chao Xue[2]*, Su Pan[1]†, Yuwei Sun[1], Minlong Peng[3]

[1]School of Internet of Things, Nanjing University of Posts and Telecommunications
[2]School of Software, Beihang University
[3]Fudan University
{2022070804, supan, 2021070706}@njupt.edu.cn, xuechao@buaa.edu.cn, mlpeng16@fudan.edu.cn

Abstract

Temporal knowledge graph question answering (TKGQA) involves multi-hop reasoning over temporally constrained entity relationships in the knowledge graph to answer a given question. However, at each hop, large language models (LLMs) retrieve subgraphs with numerous temporally similar and semantically complex relations, increasing the risk of suboptimal decisions and error propagation. To address these challenges, we propose the multi-hop reasoning enhanced (MRE) framework, which enhances both forward and backward reasoning to improve the identification of globally optimal reasoning trajectories. Specifically, MRE begins with prompt engineering to guide LLM in generating diverse reasoning trajectories for the given question. Valid reasoning trajectories are then selected for supervised fine-tuning, serving as a cold-start strategy. Finally, we introduce Tree-Group Relative Policy Optimization (T-GRPO)—a recursive, tree-structured learning-by-exploration approach. At each hop, exploration establishes strong causal dependencies on the previous hop, while evaluation is informed by multi-path exploration feedback from subsequent hops. Experimental results on two TKGQA benchmarks indicate that the proposed MRE-based model consistently surpasses state-of-the-art (SOTA) approaches in handling complex multi-hop queries. Further analysis highlights improved interpretability and robustness to noisy temporal annotations.

Introduction

Temporal Knowledge Graph Question Answering (TKGQA) aims to answer temporally-aware questions by constructing a time-sensitive subgraph centered on a target entity, performing multi-hop reasoning over relevant entities and temporal relations, and ultimately inferring the correct answer. Traditional TKGQA approaches mainly rely on embedding-based methods, which align temporal relational graphs with natural language queries through latent representations (Chen et al. 2022; Xue et al. 2024; Chen, Liao, and Zhao 2023). Although effective for in-distribution queries, these methods often exhibit poor generalization to out-of-distribution or temporally complex scenarios. Leveraging the extensive pre-trained knowledge of large language models (LLMs) and recent advances in long-context modeling (Su et al. 2024; Chen et al. 2023), there is growing momentum toward applying both commercial (Achiam et al. 2023) and open-source (Touvron et al. 2023; Yang et al. 2024) LLMs to TKGQA. LLMs show particular strength in multi-hop temporal reasoning, where answering a question requires traversing a sequence of temporally grounded facts (Chen et al. 2024a; Hu et al. 2025). This paradigm shift from embedding-based to LLM-based TKGQA opens new avenues for integrating pretrained linguistic knowledge with temporal graph structures, leading to improvements in both reasoning accuracy and robustness.

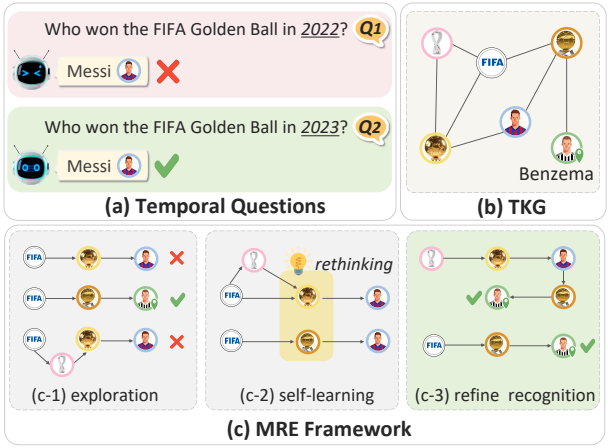

Figure 1: Retrieval errors under complex temporal facts (b) and the resolution by the MRE framework (c).

However, in LLM-based TKGQA approaches, the complexity of event retrieval (Dziri et al. 2021; Shi et al. 2024) and ambiguous temporal relations (Qian et al. 2024; Zha et al. 2024) can mislead language models(LM), resulting in suboptimal decisions during intermediate reasoning steps. As illustrated in Figure 1(a), when querying Golden Ball recipients in 2023, the absence of competing candidates enables the LLM to identify the correct answers directly from the subgraphs. In contrast, the year 2022 presents a more challenging case: FIFA issues two distinct Golden Ball awards, and Messi's World Cup triumph received unprecedented media attention. This prominence misleads the LLM,

*These authors contributed equally.
†Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

causing confusion in identifying the correct award recipient (Ballon d'Or). Although selecting Messi as the final answer is incorrect, including him as an intermediate reasoning node reflects a suboptimal but plausible inference. These cases highlight the inherent difficulty in guiding LLMs toward globally optimal trajectories in multi-hop reasoning.

Reinforcement Learning from Human Feedback (RLHF) (Rafailov et al. 2023; Schulman et al. 2017) is a fine-tuning paradigm that steers models toward globally optimal solutions through preference optimization. As an advanced extension of RLHF, group relative policy optimization (GRPO) (Guo et al. 2025), improves reasoning by comparing multiple candidate outputs and leveraging contrastive learning and enhanced exploration. Compared to traditional methods such as PPO (Schulman et al. 2017), GRPO offers greater robustness and a stronger capacity for global optimality in reasoning tasks.

Unlike single-turn question answering (QA) tasks, where immediate feedback is available from the final answer, TKGQA involves multi-hop reasoning to retrieve intermediate information before arriving at the final answer. This inherently introduces the problem of reward sparsity (Guo et al. 2020; Devidze, Kamalaruban, and Singla 2022; Zhang et al. 2023). As a result, it becomes challenging to guide LLMs to escape local optima during intermediate steps and achieve global optimality throughout the reasoning trajectory, while simultaneously mitigating the effects of sparse rewards in the RLHF optimization process.

To extend GRPO's global optimization capability from single-turn QA to multi-hop trajectory reasoning, we propose the **Multi-hop Reasoning Enhanced (MRE)** framework, which comprises three key components: **(1) Multi-Trajectory Sampling.** Leveraging GPT-4 with prompt engineering, we sample diverse multi-hop reasoning trajectories from a few-shot dataset under varying temperature settings. Trajectories that produce the correct final answers are identified as positive examples. From these, we construct a fine-grained dataset by extracting each intermediate reasoning step as an independent training instance. **(2) Cold Start Supervised Fine-Tuning**. Building on the dataset constructed in (1), we apply supervised fine-tuning to the target model, guiding it to imitate the multi-hop reasoning process and improving its adherence to instructions. **(3) Tree-Group Relative Policy Optimization.** To address the challenge of sparse rewards during exploration (T-GRPO), T-GRPO adopts a tree-based search strategy, leveraging the exploration results of constructed subtrees for evaluation and learning. Specifically, a search tree is constructed to perform g rounds of reasoning on the given subgraph, guided by the input question. At each hop, the subgraph expands along different branching directions of subsequent subtrees, and the exploration process continues until the final answer is derived. Upon receiving evaluation signals from the g reasoning trajectories returned by their respective subtrees, group reward is applied to compute their contributions. Finally, the GRPO algorithm is employed to model relative preferences among the explored trajectories within each tree, enabling iterative updates of decision policies back to the root.

Our contributions are summarized as follows: 1) We propose the MRE framework, a LLM-based multi-hop reasoning enhancement framework for TKGQA. 2)We propose T-GRPO, a reinforcement learning method for training LLMs in multi-hop reasoning for TKGQA. 3) Experiments with several TKGQA datasets demonstrate the effectiveness of MRE framework.

Related Work

LLM-based TKGQA With the development of commercial and open-source LLMs, their application to reasoning tasks in TKGQA is garnering increased attention. (Chen et al. 2024a; Hu et al. 2025) propose multi-turn interactive prompts for multi-hop reasoning, while (Lee et al. 2023) leverage in-context learning to enhance open-source model performance. (Xia et al. 2024) introduce a hybrid method that combines GNN with LLM voting for multi-hop QA. Despite their strong reasoning capabilities, commercial LLMs struggle to effectively integrate TKG knowledge, limiting their applicability to complex TKGQA tasks. Consequently, an increasing number of studies turn to constructing specialized TKGQA models using open-source backbones. (Yuan et al. 2024; Xiong et al. 2024; Chen et al. 2024b) focus on generating high-quality temporal chain-of-thought (CoT) data for fine-tuning, with (Yuan et al. 2024; Xiong et al. 2024) emphasizing multi-path reasoning and (Chen et al. 2024b) adopting in-context learning. Separately, (Qian et al. 2024) fine-tunes a rewriter to transform complex temporal constraints into explicit time points, while (Yang et al. 2023) constructs a temporally sensitive pretraining model. However, these approaches mainly aim to optimize single-hop reasoning accuracy, neglecting global optimality across the entire multi-hop reasoning trajectory.

RL for Reasoning With CoT based methods (Wei et al. 2022; Yao et al. 2023; Jin et al. 2024) that enhance LLM reasoning via static supervision, their effectiveness remain constrained by pretrained knowledge of the model. GRPO (Guo et al. 2025) addresses this limitation by autonomously discovering CoT trajectories and leveraging high-quality reasoning for self-improvement. To improve training stability and reasoning precision, recent efforts (Yu et al. 2025; Zheng et al. 2025) focus on enhancing token-level differentiation, allowing the model to assign more informative and fine-grained credit signals across sequences. In parallel, other studies explore rule-guided reasoning by injecting symbolic constraints through RL (Fang, Ma, and Wang 2025; Jiang et al. 2025; Ma et al. 2025). Beyond core reasoning, GRPO is also adapted for downstream tasks (Chen et al. 2025; Liu et al. 2025b,a). Our MRE combines reinforcement learning with a tree-structured exploration and learning strategy to address sparse reward signals in multi-hop reasoning while identifying globally optimal trajectories.

MRE Framework

In this section, we present a comprehensive description of our proposed MRE framework, which comprises three core components: (1) Multi-trajectory sampling, (2) Cold start supervised fine-tuning and (3) T-GRPO based reinforcement

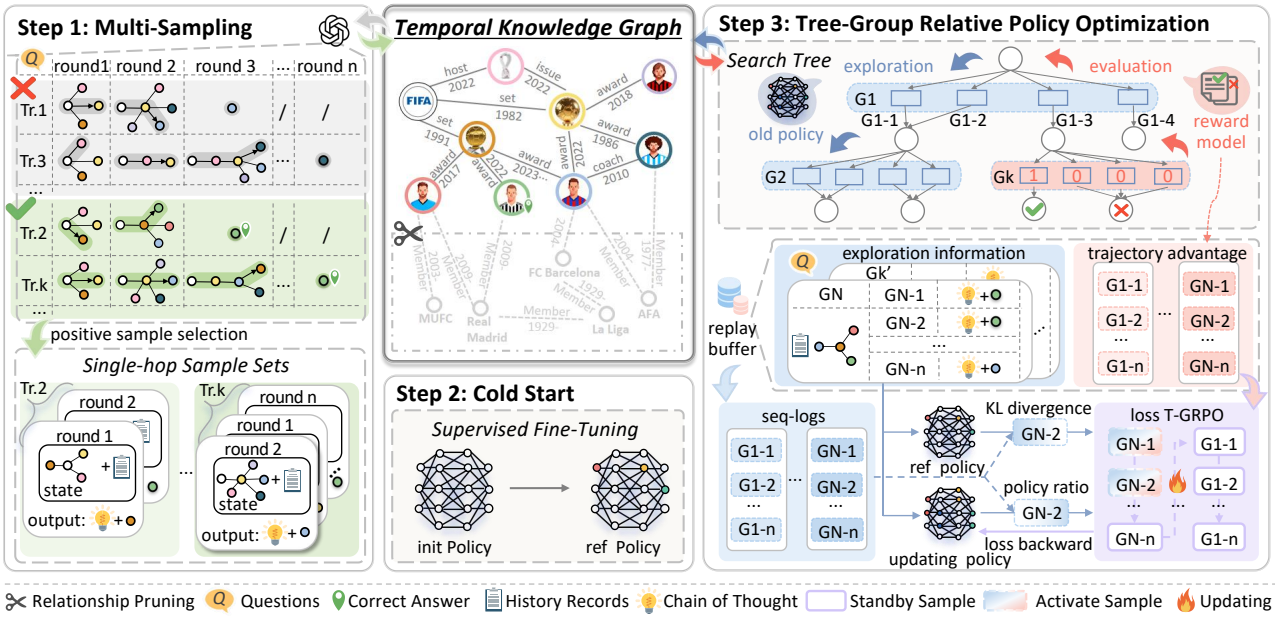

Figure 2: Overall architecture of the MRE framework. First, prompt engineering is used to guide the LLM in generating diverse multi-hop reasoning trajectories. Second, valid trajectories are selected for SFT, providing a cold-start policy. Finally, tree-structured exploration with T-GRPO recursively optimizes reasoning paths by leveraging forward decisions and backward feedback across each hops.

learning. The following subsections will provide detailed explanations of each component's implementation.

Task Definition

Let the temporal knowledge graph (TKG) be denoted as $\mathcal{K} := (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, and its 1-hop subgraph centered on an entity e represented as $\mathcal{K}_e := (\mathcal{E}_e, \mathcal{R}_e, \mathcal{T}_e, \mathcal{F}_e)$. A fact in $\mathcal{K}_e$ can be formalized as $(e, r, o, \tau) \subset \mathcal{F}_e$ or $(s, r, e, \tau) \subset \mathcal{F}_e$, where s, e, and $o \in \mathcal{E}_e$ denote the subject entity, the central entity and the object entity, respectively. The relation $r \in \mathcal{R}_e$ denotes the relationship between entities, where $\mathcal{R}_e \subset \mathcal{R}_{QA}$ is constrained by the QA task, and $\tau \in \mathcal{T}$ denotes the associated temporal information. LLM-based TKGQA tasks can be formulated as follows. Given a question Q and an initial entity e_h, the LLM first constructs a 1-hop subgraph $\mathcal{K}_{e_h}$ centered on the current entity e_h. It then performs multi-hop reasoning by selecting a next-hop entity e' from the facts $\mathcal{F}_{e_h}$ or directly producing the answer based on the retrieved information, where the answer can be an entity or a timestamp τ. The optimization objective for multi-hop reasoning in TKGQA is defined as follows:

$$\max_{\pi} \mathbb{E}\left[\sum_{i=0}^{K} \gamma R_i(\mathcal{F}_{e_h^i})\right] \quad \text{s.t.} \quad R_i = 0 \quad \forall i < K \tag{1}$$

Multi-Trajectory Sampling

To provide high-quality cold start samples for supervised fine-tuning, we employ GPT-4 to sample multiple reasoning trajectories for each QA under varying reasoning temperature settings . These trajectories are then used to construct a supervised fine-tuning dataset.

We construct multi-hop reasoning trajectories in TKGs by recursively applying 1-hop inference, where each predicted entity becomes one of a new central entity for the next hop. This iterative process proceeds until the final answer is derived. The 1-hop inference and its transition to subsequent hops can be formalized as follows:

$$a_k \leftarrow LLM(Q, H_k, S_k, E_k) \tag{2}$$

$$S_{\text{pruning}} = \text{Top-}P(Q, \mathcal{F}_E) \tag{3}$$

Let LLM denote the sampling model, and let Q denote the input question, and E_k be the central entity . Furthermore, we define $\mathcal{F}_{E_k}$ as the set of textual facts centered on E_k. The action a_k denotes the next-hop decision made by the LLM based on the current subgraph, which consists of the generated CoT and the selection of the next-hop entity $E_{k'}$. At each round, the retrieved subgraph is constructed by selecting P relevant facts from $\mathcal{F}_{E_k}$ using the relevance retrieval function in Function (3), where relevance scores are computed based on the similarity between the question Q and each candidate fact $(s, r, o, \tau) \in \mathcal{F}_{E_k}$. The historical context H_k accumulates the facts traversed in hops and is used to guide the decision-making of LLM. Following the definition of 1-hop reasoning in TKGQA, we extend it to the multi-hop setting.

$$Tr_j^t = \{Tr_j^k \sim LLM^t(Q, H_k, S_k, E_k)\}_{k=1}^{N} \tag{4}$$

Here, Tr_j^t denotes a reasoning trajectory generated by the LLM with temperature t, where N is the final step index of the j-th trajectory. The set Tr_{Q_i} contains M completed

trajectories sampled under different temperature settings for the question Q_i, and a'_j denotes the final answer produced by Tr^t_j. We then construct a positive trajectory set for Q_i by selecting each trajectory, whose output answer matches the ground-truth a^* as a positive sample.

$$Tr_{Q_i} = \{(a'_j, Tr^t_j)\}^M_{j=1}, t \in T \tag{5}$$

$$Tr^+_{Q_i} = \{Tr^t_j \in Tr_{Q_i} \mid a'_j = a^*\} \tag{6}$$

Finally, we build a subset d from the full dataset D to construct a positive trajectory set $Tr^+_{d\sim D}$, where $Tr^+_{d\sim D} = \{(Tr_{Q_i}, Q_i, a^*)\}^{|d|}_{i=1}$.

Cold start supervised fine-tuning

To enhance the instruction-following ability of the target LLM in multi-hop TKGQA reasoning and to cultivate core reasoning patterns, we apply supervised fine-tuning based on step-wise decisions extracted from $Tr^+_{d\sim D}$. The fine-tuning objective is defined as follows:

$$\underset{\theta}{\text{minimize}} \quad \sum_{i=1}^{V}\sum_{j=1}^{M}\sum_{k=1}^{N} \text{Loss}\left(y_{ijk}, \pi(Q_i, S_{ijk}, H_{ijk}, E_{ijk}; \theta)\right) \tag{7}$$

In the above objective, π denotes the LLM to be optimized, parameterized by trainable parameters θ, and $y_{ijk} = (CoT_{ijk}, a_{ijk})$ represents the reasoning process and action at step k of the j-th trajectory for the i-th question. Here, V represents the total number of questions in $Tr^+_{d\sim D}$, M denotes the number of trajectories associated with a given question Q, and N corresponds to the length of the current trajectory Tr^t_j. Based on this formulation, we develop a basic multi-hop reasoning model π^{SFT}_θ for TKGQA.

T-GRPO

As π^{SFT}_θ tends to converge to locally optimal solutions during trajectory reasoning, we introduce T-GRPO to enhance exploration and incorporate the contrastive learning paradigm of GRPO, thereby guiding the model toward globally optimal reasoning paths.

Exploration GRPO performs group-wise policy optimization by sampling multiple candidates under the same question and leveraging their relative performance within the group. The sampling process is defined as: $G \leftarrow \{a_i \sim \pi_\theta\}^g_{i=1}$, where G represents a group of g samples generated by the policy π_θ for a given input.

Multi-hop reasoning in TKGQA exhibits strong causal dependencies, where each decision is conditioned on the inference of the previous hop. Building on this, we define the search tree centered on E, denoted as:

$$Tree_{k'} = \{E, \text{Searching}(E, S, H, Q, \ell), R_E\}, \quad k' > 0 \tag{8}$$

where $Searching$ function refers to the multi-hop search (Algorithm 1), R_E denotes the evaluation of the central entity E obtained after the search, while $Tree_{k'}$ refers to the k'-th tree constructed during the search process at depth ℓ. The core of GRPO is to construct the sampling set G and perform preference optimization iteratively. Therefore, we construct the sampling expression for $G_{k'}$ based on the $Tree_{k'}$:

$$G_{k'} = \{a^{(j')}_{k'} \sim \pi_\theta(Q, H, (S, E) \leftarrow a^{(j)}_k)\}^g_{j'=1}, \quad k' > 0 \tag{9}$$

where $G_{k'}$ denotes the group constructed by the policy network π_θ in the k'-th tree using the subgraph S, which is determined by E. Moreover, both S and E are influenced by the information on the j-th output in the k-th tree, denoted $a^{(j)}_k$, where $a^{(j)}_k = (CoT^{(j)}_k, E^{k,j}_{\text{next}})$. Consequently, the next-hop subgraph $S^{k,j}_{\text{next}}$ is constructed accordingly (line 12–14 in Algorithm 1).

Evaluation The sparsity of trajectory-level rewards in multi-hop reasoning poses significant challenges to the convergence of learning algorithms (Devidze, Kamalaruban, and Singla 2022; Zhang et al. 2023). To address this, T-GRPO builds on the search strategy defined in the previous hop and adopts a tree-structured multi-hop search process. It assigns backward credit by evaluating each node based on the aggregated scores of its downstream search paths. By propagating reward signals along the tree, this approach effectively mitigates the impact of sparse supervision and facilitates more stable and efficient learning.

For a next-hop entity $E^{k,j}_{\text{next}}$, if action $a^{(j)}_k$ selected in j-th inference of the k-th group corresponds to the final answer, the associated central entity is denoted $E^{k,j}_{\text{leaf}}$. Otherwise, if $a^{(j)}_k$ is the entity chosen for the next hop, we denote it as

Algorithm 1: Tree-group Searching

Input: π^{Old}_θ, g, max_depth, $buffer$, E_{init}, S_{init}, Q
Output: Sampled buffer $buffer^*$

1: **function** Searching(E, S, H, Q, ℓ)
2: **if** $\ell \geq max\_depth$ **then**
3: **return** 0
4: **end if**
5: $Group, Reward \leftarrow \emptyset$
6: **for** $i \leftarrow 1$ **to** g **do**
7: $a_i \leftarrow \pi^{Old}_\theta(Q, E, H, S)$
8: **if** a_i is answer **then**
9: compute $Score_i$ based on (10)
10: go to line (17)
11: **end if**
12: Extract E_{next} from a_i
13: $\mathcal{F}_{E_{\text{next}}} \leftarrow \mathcal{K}_{E_{\text{next}}}$
14: $S_{\text{next}} \leftarrow$ Pruning $\mathcal{F}_{E_{\text{next}}}$ using (3)
15: $H_{\text{next}} \leftarrow H \cup S$
16: $Score_i \leftarrow$ Searching($E_{\text{next}}, S_{\text{next}}, H_{\text{next}}, Q, \ell + 1$)
17: $Group \leftarrow Group \cup \{(a_i, Q, H, S, E)\}$
18: $Reward \leftarrow Reward \cup \{Score_i\}$
19: **end for**
20: Compute R_E using (11)
21: Push $\{Group, Reward\}$ to $buffer$ ▷ asynchronous
22: **return** R_E
23: **End function**
24: Searching(E_{init}, S_{init}, $\emptyset$, Q, 0)

$E^{k,j}_{\text{root}}$. The evaluation of the leaf entity $E^{k,j}_{\text{leaf}}$ is conducted by directly comparing it with the ground-truth answer a^*, as formalized in Function (10):

$$R_{E^{k,j}_{\text{leaf}}} = \begin{cases} 1, & a^{(j)}_k = a^* \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The evaluation of non-terminal $E^{k,j}_{\text{root}}$ is based on the reward computed over the group $G^{k,j}_{\text{next}}$, which is determined by performing a search on the subgraph $S^{k,j}_{\text{next}}$.

$$R_{E^{k,j}_{\text{root}}} = \frac{1}{g} \sum_{j'=1}^{g} R(a^{(j')}_{k'}) \tag{11}$$

In Function (11), $a^{(j')}_{k'}$ denotes an inference result generated in $G^{k,j}_{\text{next}}$, which produces a total of g responses. The quality of each $a^{(j')}_{k'}$ is assessed according to its corresponding next-hop information.

Storage After evaluating a group of sampled trajectories, we store the resulting group-level sampling data into an asynchronous buffer, decoupling trajectory evaluation from the GRPO training process. Specifically, the storage structure for the k-th tree is defined as: $G_k = \{(a_j, Q, H_j, S_j, E_j, R_{E_j})\}^g_{j=1}$ (line 21 in Algorithm 1). Once a storage structure is fully sampled, it is forwarded to the GRPO algorithm for parameter updates.

Training In the GRPO training framework, three model variants are maintained throughout the optimization process: the update model π_θ, the reference model π^{Ref}_θ, and the sampling model π^{Old}_θ. After every μ policy updates on a batch of sampled graphs G, the sampling model π^{Old}_θ is synchronized with the latest parameters of π_θ to ensure stable exploration. In contrast, T-GRPO adopts a subtree-level optimization paradigm. Rather than updating the policy at fixed intervals, it postpones gradient updates until a complete traversal and evaluation of a sampled subtree is performed. By computing gradients based on coherent and trajectory-consistent feedback, this design produces richer learning signals with enhanced contextual awareness. Once π_θ completes learning from the sampling results across all subtrees, we set $\pi^{\text{Old}}_\theta \leftarrow \pi_\theta$. The complete training procedure is presented formally in Algorithm 2.

Algorithm 2: Single sample in Tree-Group Relative Policy Optimization

Input: π^{SFT}_θ, $buffer$, I, μ,E_{init}, S_{init}, Q
Output: π^*_θ

policy model $\pi_\theta \leftarrow \pi^{\text{SFT}}_\theta$
reference model $\pi^{\text{Ref}}_\theta \leftarrow \pi^{\text{SFT}}_\theta$
for step = $1, \ldots,$ I **do**
  Update the old policy model $\pi^{\text{Old}}_\theta \leftarrow \pi_\theta$
  $buffer \leftarrow$ Searching (E_{init}, S_{init}, $\emptyset$, Q, 0)
  for $\{Group, Reward\}$ in $buffer$ **do**
    Compute token-level group relative advantage estimation based on $Group$ and $Reward$
    for GRPO iteration $= 1, \ldots, \mu$ **do**
      Update π_θ by maximizing GRPO objective
    end for
  end for
end for

Experimental Settings

Datasets We evaluate our proposed MRE framework on two challenging and complementary TKGQA benchmarks. The first, CRONQUESTIONS (Saxena, Chakrabarti, and Talukdar 2021), is a large-scale dataset constructed from Wikidata, featuring 410K questions that involve 1- to 3-hop temporal reasoning. With rich annotations of fine-grained timestamps and a diverse mix of entity- and time-centric queries, it has become a standard benchmark for temporal QA. The second, TIMEQUESTIONS (Sharma et al. 2023), unifies 13.5K questions from five existing datasets into a comprehensive benchmark that emphasizes multi-hop temporal reasoning. By covering explicit, implicit, comparative, and ordinal question types, thereby offering a rigorous testbed for evaluating models' temporal commonsense and ordinal reasoning capabilities.

Evaluation Metrics We adopt the commonly used evaluation metrics **Hits@1** and **Hits@10**, which are defined as follows for $K \in \{1, 10\}$:

$$\text{Hits@}K = \frac{1}{|\mathcal{T}|} \sum_{q \in \mathcal{T}} \mathbf{1}\left(\text{rank}(q) \leq K\right), \tag{12}$$

where $\mathcal{T}$ denotes the test set. For a given question q, $\text{rank}(q)$ is the rank assigned by the model to the correct answer within the list of candidates. The indicator function $\mathbf{1}(\cdot)$ returns 1 if the condition inside holds, and 0 otherwise.

Baseline Methods For the **CRONQUESTIONS** dataset, we compare MRE with several baselines, including EaE (Feng et al. 2020), EmbedKGQA (Saxena, Tripathi, and Talukdar 2020), CronKGQA (Saxena, Chakrabarti, and Talukdar 2021), EntityQR (Mavromatis et al. 2022), TMA (Liu et al. 2023), TSQA (Shang et al. 2022), CTRN(Jiao et al. 2022), TempoQR (Mavromatis et al. 2022) as well as language models BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), and ChatGPT. For the **TimeQuestions** dataset, the baselines comprise CronKGQA, TempoQR and TwiRGCN (Sharma et al. 2023) .

Experimental Results and Analysis

Overall Performance

Results on CRONQUESTIONS As shown in Table 1, our MRE framework sets a new SOTA with 98.2% Hits@1 and 99.6% Hits@10 on the overall test set—outperforming the previous best (TempoQR) by +6.4% and +1.8%, respectively. This substantial gain highlights the effectiveness of our trajectory-level temporal reasoning. MRE generalizes well across both simple and complex questions. It achieves near-perfect accuracy on Simple questions 99.9% (Hits@1), and consistently strong performance across Entity (98.2%) and Time (99.4%) answer types—demonstrating robust factual and temporal grounding. On more challenging complex

Model	Hits@1					Hits@10				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time		Complex	Simple	Entity	Time
EmbedKGQA	0.288	0.286	0.290	0.411	0.057	0.672	0.632	0.725	0.850	0.341
EaE	0.288	0.257	0.329	0.318	0.231	0.678	0.623	0.753	0.668	0.698
CronKGQA	0.647	0.392	0.987	0.699	0.549	0.884	0.802	0.990	0.898	0.857
EntityQR	0.745	0.562	0.990	0.831	0.585	0.944	0.906	0.993	0.962	0.910
TMA	0.784	0.632	0.987	0.792	0.743	0.943	0.904	0.995	0.947	0.936
TSQA	0.831	0.713	0.987	0.829	0.836	0.980	0.968	0.997	0.981	0.978
TempoQR	0.918	0.864	0.990	0.926	0.903	0.978	0.967	0.993	0.980	0.974
BERT *w/o tkg*	0.071	0.086	0.052	0.077	0.06	0.213	0.205	0.225	0.192	0.253
RoBERTa *w/o tkg*	0.07	0.086	0.05	0.082	0.048	0.202	0.192	0.215	0.186	0.231
ChatGPT *w/o tkg*	0.151	0.144	0.160	0.134	0.182	0.308	0.308	0.307	0.257	0.402
BERT *w/ tkg*	0.243	0.239	0.249	0.277	0.179	0.620	0.598	0.649	0.628	0.604
RoBERTa *w/ tkg*	0.225	0.217	0.237	0.251	0.177	0.585	0.542	0.644	0.583	0.591
ChatGPT *w/ tkg*	0.754	0.579	0.987	0.689	0.873	0.852	0.746	0.992	0.808	0.933
MRE(Ours)	**0.982**	**0.970**	**0.999**	**0.982**	**0.994**	**0.996**	**0.994**	**0.998**	**0.996**	**0.998**

Table 1: Performance comparison of different models on CRONQUESTIONS. The best and second best results are marked in **bold** and underlined, respectively. *w/o tkg* indicates that LMs answer the questions directly without using TKG information, and *w/ tkg* indicates that LMs answer the questions with TKG background knowledge.

Model	Overall	Explicit	Implicit	Temporal	Ordinal
CronKGQA	0.462	0.466	0.445	0.511	0.369
TempoQR	0.416	0.465	0.360	0.400	0.349
TwiRGCN(average)	**0.605**	**0.602**	0.586	0.641	0.518
TwiRGCN(interval)	0.603	0.599	0.603	**0.646**	0.494
MRE(Ours)	0.594	0.598	**0.631**	0.576	**0.578**

Table 2: Hits@1 for different models on TimeQuestions.

multi-hop questions, MRE surpasses TempoQR by +10.6% in Hits@1, validating the advantage of global trajectory optimization over local reasoning. Table 3 further breaks down performance by reasoning types. Our MRE framework exhibits substantial improvements on challenging temporal categories such as *before/after*, *first/last*, and *temporal joins*, where conventional models often struggle due to temporal ambiguity, sparse supervision, and multi-hop error propagation. These gains highlight the ability of MRE to model complex temporal dependencies and reason across distant events. At the same time, it maintains near-perfect accuracy in factoid-style subsets—achieving 99.6% Hits@1 in both *Simple-Entity* and *Simple-Time*—demonstrating its robustness in handling both shallow retrieval and deep temporal inference within a unified reasoning framework.

Clearly, embedding-based methods (e.g. EmbedKGQA, EaE) perform poorly on time-centric questions ($\leq$ 23.1% Hits@1), and language models without explicit temporal input (e.g. BERT, RoBERTa, ChatGPT *w/o tkg*) score even lower (7.0%–15.1%). Even in temporal context, ChatGPT *w/ tkg* reaches only 75.4%, falling 22.8% short of MRE. These results underscore the necessity of structured, trajectory-aware temporal reasoning for the TKGQA task.

Results on TimeQuestions As shown in Table 2, MRE delivers strong overall performance on the TIMEQUESTIONS benchmark, achieving 59.4% Hits@1—closely matching the best result reported by TwiRGCN while clearly outperforming all prior methods in key reasoning categories. In particular, MRE establishes SOTA results on *implicit* and *ordinal* questions-two of the most demanding types that require abstract temporal inference beyond surface-level timestamp matching. These improvements demonstrate MRE's ability to handle event salience, temporal abstraction, and relative ordering more effectively than existing models. These findings reinforce the superior generalization of MRE in complex temporal QA scenarios.

Category	Complex Question			Simple Question		All
	Before/ After	First/ Last	Time Join	Simple Entity	Simple Time	
EmbedKGQA	0.199	0.324	0.223	0.421	0.087	0.288
T-EaE-add	0.256	0.285	0.175	0.296	0.321	0.278
T-EaE-replace	0.256	0.288	0.168	0.318	0.346	0.288
CronKGQA	0.288	0.371	0.511	0.988	0.985	0.647
TMA	0.581	0.627	0.675	0.988	0.987	0.784
TSQA	0.504	0.721	0.799	0.988	0.987	0.831
TempoQR	0.714	0.853	0.978	0.988	0.987	0.918
CTRN	0.747	0.880	0.897	0.991	0.987	0.920
MRE(Ours)	**0.926**	**0.948**	**0.994**	**0.992**	**0.996**	**0.982**

Table 3: Hits@1 for different question types.

Notably, these results highlight the effectiveness of the MRE framework in improving LLM with structured subgraph retrieval and trajectory-aware temporal reasoning. MRE enables robust, interpretable multi-hop inference, setting a new performance standard for TKGQA across both synthetic and real-world benchmarks.

Ablation Study

To evaluate the contribution of each component within the MRE framework, we perform a series of ablation studies systematically by removing or altering key modules, as summarized in Table 4. The single-hop experiment yields

Model Variant	Hits@1		Hits@10	
	Overall	Complex	Overall	Complex
Full MRE	**0.982**	**0.970**	**0.996**	**0.994**
w/o T-GRPO	0.921	0.892	0.978	0.965
w/o Cold Start	0.904	0.871	0.962	0.948
w/o Multi-Sampling	0.876	0.842	0.945	0.931
Single-Hop	0.812	0.774	0.910	0.892

Table 4: Ablation results on CRONQUESTIONS. *w/o* indicates removal of the corresponding module from the full MRE pipeline.

Model	Hits@1@1-hop	Hits@1@2-hop	Hits@1@3-hop
CronKGQA	0.991	0.873	0.512
TempoQR	0.992	0.952	0.754
MRE (Ours)	**0.999**	**0.981**	**0.943**

Table 5: Performance comparison of different models at various reasoning depths.

only 81.2% Hits@1 on overall questions and 77.4% on complex ones, indicating that shallow reasoning over limited evidence chains is insufficient to handle the relational and temporal complexity inherent in TKGQA. When the T-GRPO module is removed, performance significantly drops to 92.1% (Overall) and 89.2% (Complex), despite the model still being trained on positive trajectories. This suggests that without structured trajectory exploration, the model tends to overfit local patterns and struggles to make globally coherent decisions. The tree-structured exploration enabled by T-GRPO, coupled with backward credit assignment, proves crucial for contrastive optimization and effective reasoning across multiple hops. Similarly, removing cold-start fine-tuning leads to a further decline in performance, underscoring the importance of supervised initialization in providing inductive bias and stabilizing early policy learning. Finally, disabling multi-trajectory sampling leads to the most significant performance drop, indicating that exposing the model to diverse and consistent reasoning paths is crucial for effective multi-hop temporal reasoning. Rather than relying on a single trajectory, multi-trajectory sampling improves generalization, reduces overfitting, and enhances robustness to temporal ambiguity. In general, the ablation results validate the complementary roles of each module and highlight their necessity to achieve strong performance in deep temporal reasoning tasks.

Method	Hits@1@10k	Hits@1@50k	Peak
PPO	0.827	0.901	0.922
GRPO (Flat)	0.845	0.932	0.951
MRE (Ours)	**0.902**	**0.968**	**0.982**

Table 6: Training results compared with different RLHF approaches.

Multi-hop Temporal Reasoning Depth Analysis

To rigorously validate the superiority of our MRE framework in multi-hop temporal reasoning, we analyze its performance across different reasoning depths on the CRONQUESTIONS benchmark: As shown in Table 5, MRE consistently outperforms strong baselines across all reasoning depths. It achieves near-perfect accuracy on 1-hop questions (99.9%), and maintains robust performance on 3-hop questions (94.3%), reducing the 3-hop error rate by over 75% compared to TempoQR. In particular, as the depth of reasoning increases, the performance gap between MRE and existing methods widens significantly, highlighting MRE's superior capability in handling deep and complex multi-hop temporal queries. This advantage stems from two key design choices: (1) the tree-structured sampling strategy, which facilitates more effective exploration during training, and (2) the GRPO module, which propagates reward signals backward and optimizes each intermediate decision step to ensure global trajectory optimality.

RLHF Training Analysis

To quantitatively assess the advantages of tree-structured reward propagation in multi-hop temporal reasoning, we compare the performances of PPO, GRPO, and T-GRPO. As reported in Table 6, T-GRPO achieves a Hits@1 of 90.2% with 10k samples, surpassing PPO and GRPO (Flat) by 7.5% and 5.7%, respectively. This highlights a key limitation of GRPO (Flat). Although GRPO (Flat) replaces PPO's advantage estimate with group-relative optimization to enable deeper exploration of the trajectory under the same training budget, it still relies on sparse reward supervision that is assigned only in the final step. In contrast, T-GRPO leverages a tree-structured optimization framework that propagates supervised signals along the entire reasoning path, facilitating fine-grained intermediate rewards and thereby enabling more informed policy updates. With extended training, T-GRPO further attains a peak Hits@1 of 98.2%, demonstrating both superior sample efficiency and stronger convergence. These results confirm the effectiveness of tree-based credit assignment in guiding global policy optimization for complex multi-hop reasoning.

Conclusion

We propose MRE, a unified framework for enhancing multi-hop reasoning in TKGQA, which integrates trajectory sampling, supervised fine-tuning, and a novel T-GRPO algorithm. By jointly modeling forward exploration and backward evaluation, MRE substantially strengthens the stepwise reasoning capability of large language models, enabling the identification of globally optimal reasoning trajectories over temporal knowledge graphs. Extensive experiments across multiple benchmarks demonstrate that MRE consistently outperforms previous SOTA methods. Overall, this work highlights the effectiveness of combining preference-based optimization with structured multi-hop reasoning, and points toward a promising direction for building more accurate and explainable TKGQA systems.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62071244.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chen, S.; Wong, S.; Chen, L.; and Tian, Y. 2023. Extending Context Window of Large Language Models via Positional Interpolation. arXiv:2306.15595.

Chen, X.; Li, G.; Wang, Z.; Jin, B.; Qian, C.; Wang, Y.; Wang, H.; Zhang, Y.; Zhang, D.; Zhang, T.; et al. 2025. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*.

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

Chen, Z.; Liao, J.; and Zhao, X. 2023. Multi-granularity Temporal Question Answering over Knowledge Graphs. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (Vol. 1, pp. 11378–11392). Association for Computational Linguistics (ACL)*. Bangkok, Thailand: Association for Computational Linguistics.

Chen, Z.; Zhang, Z.; Li, Z.; Wang, F.; Zeng, Y.; Jin, X.; and Xu, Y. 2024b. Self-Improvement Programming for Temporal Knowledge Graph Question Answering. In Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 14579–14594. Torino, Italia: ELRA and ICCL.

Chen, Z.; Zhao, X.; Liao, J.; Li, X.; and Kanoulas, E. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems*, 251: 109134.

Devidze, R.; Kamalaruban, P.; and Singla, A. 2022. Exploration-Guided Reward Shaping for Reinforcement Learning under Sparse Rewards. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 5829–5842. Curran Associates, Inc.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *NAACL*, 4171–4186.

Dziri, N.; Madotto, A.; Zaïane, O.; and Bose, A. J. 2021. Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2197–2214. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Fang, G.; Ma, X.; and Wang, X. 2025. Thinkless: LLM Learns When to Think. *arXiv preprint arXiv:2505.13379*.

Feng, Y.; Chen, X.; Lin, B. Y.; Wang, P.; Yan, J.; and Ren, X. 2020. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. In *EMNLP*, 1295–1309.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Guo, Y.; Choi, J.; Moczulski, M.; Feng, S.; Bengio, S.; Norouzi, M.; and Lee, H. 2020. Memory Based Trajectory-conditioned Policies for Learning from Sparse Rewards. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 4333–4345. Curran Associates, Inc.

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

Jiang, L.; Wu, X.; Huang, S.; Dong, Q.; Chi, Z.; Dong, L.; Zhang, X.; Lv, T.; Cui, L.; and Wei, F. 2025. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*.

Jiao, S.; Zhu, Z.; Wu, W.; Zuo, Z.; Qi, J.; Wang, W.; Zhang, G.; and Liu, P. 2022. An improving reasoning network for complex question answering over temporal knowledge graphs. *Applied Intelligence*, 53(7): 8195–8208.

Jin, B.; Xie, C.; Zhang, J.; Roy, K. K.; Zhang, Y.; Li, Z.; Li, R.; Tang, X.; Wang, S.; Meng, Y.; and Han, J. 2024. Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 163–184. Bangkok, Thailand: Association for Computational Linguistics.

Lee, D.-H.; Ahrabian, K.; Jin, W.; Morstatter, F.; and Pujara, J. 2023. Temporal Knowledge Graph Forecasting Without Knowledge Using In-Context Learning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 544–557. Singapore: Association for Computational Linguistics.

Liu, Y.; Liang, D.; Fang, F.; Wang, S.; Wu, W.; and Jiang, R. 2023. Time-Aware Multiway Adaptive Fusion Network for Temporal Knowledge Graph Question Answering. In *ICASSP*, 1–5.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Liu, Z.; Guo, X.; Lou, F.; Zeng, L.; Niu, J.; Wang, Z.; Xu, J.; Cai, W.; Yang, Z.; Zhao, X.; et al. 2025a. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*.

Liu, Z.; Wang, P.; Xu, R.; Ma, S.; Ruan, C.; Li, P.; Liu, Y.; and Wu, Y. 2025b. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*.

Ma, W.; He, J.; Snell, C.; Griggs, T.; Min, S.; and Zaharia, M. 2025. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*.

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

Qian, X.; Zhang, Y.; Zhao, Y.; Zhou, B.; Sui, X.; Zhang, L.; and Song, K. 2024. TimeR^4 : Time-aware Retrieval-Augmented Large Language Models for Temporal Knowledge Graph Question Answering. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6942–6952. Miami, Florida, USA: Association for Computational Linguistics.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Saxena, A.; Chakrabarti, S.; and Talukdar, P. P. 2021. Question Answering Over Temporal Knowledge Graphs. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *ACL-IJCNLP*, 6663–6676.

Saxena, A.; Tripathi, A.; and Talukdar, P. P. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. In *ACL*, 4498–4507.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shang, C.; Wang, G.; Qi, P.; and Huang, J. 2022. Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs. In *ACL*, 8017–8026.

Sharma, A.; Saxena, A.; Gupta, C.; Kazemi, S. M.; Talukdar, P. P.; and Chakrabarti, S. 2023. TwiRGCN: Temporally Weighted Graph Convolution for Question Answering over Temporal Knowledge Graphs. In *EACL*, 2041–2052.

Shi, Y.; Tan, Q.; Wu, X.; Zhong, S.; Zhou, K.; and Liu, N. 2024. Retrieval-enhanced Knowledge Editing in Language Models for Multi-Hop Question Answering. CIKM '24, 2056–2066. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704369.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568: 127063.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xia, Y.; Wang, D.; Liu, Q.; Wang, L.; Wu, S.; and Zhang, X.-Y. 2024. Chain-of-History Reasoning for Temporal Knowledge Graph Forecasting. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 16144–16159. Bangkok, Thailand: Association for Computational Linguistics.

Xiong, S.; Payani, A.; Kompella, R.; and Fekri, F. 2024. Large Language Models Can Learn Temporal Reasoning. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10452–10470. Bangkok, Thailand: Association for Computational Linguistics.

Xue, C.; Liang, D.; Wang, P.; and Zhang, J. 2024. Question calibration and multi-hop modeling for temporal question answering. AAAI'24/IAAI'24/EAAI'24. AAAI Press. ISBN 978-1-57735-887-9.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, S.; Li, X.; Bing, L.; and Lam, W. 2023. Once Upon a Time in Graph: Relative-Time Pretraining for Complex Temporal Reasoning. In *EMNLP*, 11879–11895.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuan, C.; Xie, Q.; Huang, J.; and Ananiadou, S. 2024. Back to the Future: Towards Explainable Temporal Reasoning with Large Language Models. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701719.

Zha, Z.; Qi, P.; Bao, X.; Tian, M.; and Qin, B. 2024. M 3 TQA: Multi-View, Multi-Hop and Multi-Stage Reasoning for Temporal Question Answering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 10086–10090. IEEE.

Zhang, Y.; Du, Y.; Huang, B.; Wang, Z.; Wang, J.; Fang, M.; and Pechenizkiy, M. 2023. Interpretable Reward Redistribution in Reinforcement Learning: A Causal Approach. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 20208–20229. Curran Associates, Inc.

Zheng, C.; Liu, S.; Li, M.; Chen, X.-H.; Yu, B.; Gao, C.; Dang, K.; Liu, Y.; Men, R.; Yang, A.; et al. 2025. Group Sequence Policy Optimization. *arXiv preprint arXiv:2507.18071*.