

Article

Subgraph Reasoning on Temporal Knowledge Graphs for Forecasting Based on Relaxed Temporal Relations

Meini Yang ^{1,2}, Kerong Ben ¹, Tao He ^{3,*} and Feipeng Wang ⁴

¹ School of Electronic Engineering, Naval University of Engineering, Wuhan 430033, China; 0907032001@nue.edu.cn (M.Y.); 20101002@nue.edu.cn (K.B.)

² Department of Basic Courses, Naval University of Engineering, Wuhan 430033, China

³ Department of Information Security, Naval University of Engineering, Wuhan 430033, China

⁴ School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, China; 20101001@nue.edu.cn

* Correspondence: taohe@whu.edu.cn

Abstract

Reasoning over Temporal Knowledge Graphs (TKGs) aims to forecast future events based on historical ones. Existing approaches typically enforce strict temporal order constraints among past events; however, such rigidity limits the effective exploitation of path information during reasoning, thereby reducing both model flexibility and predictive performance. To address this limitation, this paper introduces SR-RTR (Sub-graph Reasoning based on Relaxed Temporal Relation), an interpretable subgraph reasoning framework designed to fully harness path information within TKGs. By incorporating a relaxed temporal factor, the proposed method softens the chronological constraints on historical events, broadens the sampling scope of candidate nodes during subgraph reasoning, and enhances the efficiency of path information utilization. This mechanism reflects human cognitive intuition: when the temporal gap between two events falls within a certain threshold, their sequence can be considered interchangeable. SR-RTR constructs a query-specific inference subgraph from the TKG and iteratively performs two core operations—subgraph expansion and pruning—until the entity with the highest attention score is identified as the prediction result. Extensive experiments on four benchmark datasets demonstrate that SR-RTR uncovers a greater number of reasoning paths relevant to the target prediction, leading to substantial improvements in both reasoning accuracy and computational efficiency.



Academic Editor: Francesco Calimeri

Received: 8 October 2025

Revised: 9 November 2025

Accepted: 12 November 2025

Published: 17 November 2025

Citation: Yang, M.; Ben, K.; He, T.; Wang, F. Subgraph Reasoning on Temporal Knowledge Graphs for Forecasting Based on Relaxed Temporal Relations. *Mathematics* **2025**, *13*, 3688. <https://doi.org/10.3390/math13223688>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: temporal knowledge graphs forecasting; subgraph reasoning; temporal relaxation factor

MSC: 68T05

1. Introduction

Driven by the rapid advances in big data and artificial intelligence, knowledge graphs—an essential graph-based framework for structured knowledge representation—have shown significant potential across domains such as question answering [1,2], information extraction [3,4], and recommendation systems [5,6]. Conventional static knowledge graphs encode factual information as triples (e_s, r, e_o) , where e_s and e_o denote real-world entities (represented as graph nodes), and r indicates the relation between them (depicted as a directed edge). For instance, the triple (Anna, born in, New York) serves as a canonical example. This representation effectively supports knowledge organization and inference. Yet, real-world facts evolve over time; for example, the triple (Anna, live in, New York) becomes

obsolete once Anna relocates to Washington. To model such temporally dynamic, multi-relational data, temporal knowledge graphs (TKGs) have been introduced [7,8]. Extending static triples with temporal context, TKGs augment them into quadruples (e_s, r, e_o, t) , where t specifies the timestamp of the event (e_s, r, e_o) . A typical instance is (Bob, visit, Tom, 2023-09-13). Unlike static knowledge graphs, TKGs not only preserve structural relationships among entities but also capture their temporal evolution patterns, enabling more precise modeling of dynamic real-world processes.

As a central task in TKG research, Temporal Knowledge Graph Reasoning (TKGR) focuses on predicting future events based on historical observations. It generally comprises two subtasks: entity prediction and relation prediction. The former involves identifying a missing subject or object entity at a given timestamp, whereas the latter aims to infer a missing relation at a specific time. Existing studies [9,10] typically impose a strict temporal order constraint during historical event processing, allowing only forward inference—i.e., later events can be deduced solely from earlier ones. In the hypothetical TKG illustrated in Figure 1, which contains five distinct entities (nodes) and five temporal edges, performing entity prediction (e.g., completing the quadruple (Tom, visit, ?, 2025-06-01)) requires tracing historical events associated with Tom that occurred before 2025-06-01. Under the strict temporal constraint, the path from (Tom, talk, Bob, 2025-05-01) to (Bob, visit, France, 2025-05-05) is excluded from consideration, as Bob's visit occurs later than Tom's conversation, thereby violating the rule that only earlier events can inform subsequent ones. However, in human cognitive reasoning, when the temporal gap between two events remains within a certain threshold, their order of occurrence can be treated as negligible. For instance, if the threshold is defined as four days, the interval between Bob's visit to France and Tom's conversation with Bob falls within this range, rendering the order of events inconsequential for reasoning. Under such conditions, the path from (Tom, talk, Bob, 2025-05-01) to (Bob, visit, France, 2025-05-05) can be regarded as a valid backtracking path, thereby enhancing the exploitation of informative temporal connections within the graph.

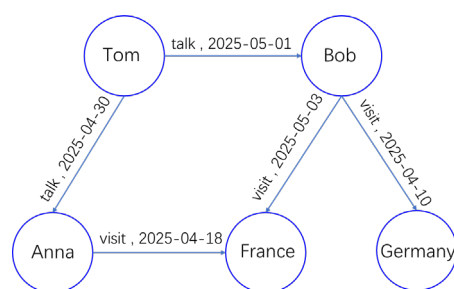


Figure 1. An example of temporal Knowledge Graph.

The main contributions of this work are summarized as follows:

- (1) Traditional TKG reasoning models enforce rigid temporal order constraints when processing historical events, which restricts reasoning paths and limits the effective utilization of available information. To mitigate this limitation, a temporal relaxation factor (δ) is introduced to soften these constraints, offering theoretical support for enhancing the flexibility of temporal reasoning.
- (2) The proposed SR-RTR model employs query-specific subgraph construction as its core mechanism for entity prediction. During subgraph expansion, the model integrates the temporal relaxation factor δ to sample valid events satisfying the relaxed temporal condition, thus expanding the scope of reasoning paths. In the subsequent pruning stage, an attention-based neighbor feature aggregation module identifies and retains events most influential for prediction. Through multiple dynamic iterations of these two stages—subgraph expansion and pruning—the model generates

reasoning subgraphs that provide a meaningful and interpretable foundation for final predictions.

- (3) Comparative experiments conducted on four TKG datasets demonstrate that the SR-RTR model consistently surpasses baseline methods. By extracting a wider range of relevant reasoning paths, it yields richer supporting evidence for prediction, thereby confirming the effectiveness of the relaxed temporal constraint paradigm and the proposed model architecture.

The remainder of this paper is organized as follows. Section 2 reviews related work on reasoning over Temporal Knowledge Graphs. Section 3 introduces the preliminaries, while Section 4 elaborates on the proposed model. Section 5 presents the experimental results, and Section 6 concludes the paper with discussions on future research directions.

2. Related Work

Early research on TKG reasoning primarily focused on integrating temporal representations into existing static knowledge graph reasoning frameworks. Jiang et al. [7] first introduced temporal constraints to the static TransE model [11], proposing the t-TransE model to encode temporal knowledge. Building upon TransH [12], Dasgupta et al. [13] developed HyTE, which projects entities and relations onto temporal hyperplanes to better exploit temporal information. Subsequent models such as Know-Evolve [14], ATiSE [15], and TiRGN [16] further enhanced temporal modeling by incorporating sequential encoding, temporal feature classification, and multi-encoder fusion, respectively. In these approaches, the validity of quadruples is evaluated by computing semantic similarity between entity and relation embeddings. However, these methods tend to depend heavily on the implicit correlations within embedding vectors, resulting in limited interpretability and the absence of explicit evidence to support predictions.

To enhance interpretability, later studies reformulated the reasoning process within a probabilistic framework. Jin et al. [17] proposed RE-Net, which models TKG reasoning as the conditional probability computation $P(\mathcal{G}_t | \mathcal{G}_{t-m:t-1})$, capturing temporal evolution patterns through probabilistic distributions. Building on RE-Net, GyGNet [18] and RE-GCN [19] further strengthened the exploitation of historical information. Despite offering a degree of interpretability, these approaches generally exhibit high computational complexity due to their reliance on intricate graph convolutional embedding operations. To address efficiency concerns, Liu et al. [20] introduced FS-Net, which achieves lightweight reasoning through dynamic frequency statistics, thereby reducing computational costs at the expense of semantic interpretability. Sun et al. [21] proposed CEGRL-TKGR, which integrates causal reasoning into probabilistic modeling to enhance interpretability; nevertheless, its causal mechanism depends heavily on complete event sequences, resulting in strong data dependency and constrained generalizability.

To further improve the interpretability of reasoning, subsequent studies have explored structure-based reasoning approaches. Han et al. [9] proposed xERTE, which constructs reasoning subgraphs through iterative sampling and leverages these subgraphs as the explanatory foundation for predictions. Liu et al. [10] introduced TLogic, which extracts temporal logical rules via temporal random walks and employs these rules as the basis for inference. Mu et al. [22] developed TIAR, which integrates time-aware relation representations with an attention-based aggregation mechanism and incorporates a path constraint module to enhance the semantic alignment between queries and subgraphs—thereby strengthening the explanatory capacity of subgraphs in prediction tasks. These approaches center on the structural properties of knowledge graphs, enabling reasoning by mining path features formed through entity–relation interactions. Structure-based reasoning not only enriches the semantic context beyond entity and relation embeddings, thereby enhancing

inference performance, but also provides traceable and interpretable evidence through the generated reasoning paths and subgraphs. This framework effectively balances predictive accuracy and interpretability. Nonetheless, most existing structure-based TKG reasoning methods still enforce strict temporal order constraints, which overrestrict reasoning paths and hinder the effective utilization of valid historical information—ultimately constraining their inferential potential.

3. Preliminary

3.1. Temporal Knowledge Graph

A TKG, denoted as $\mathcal{G}$, is a collection of fact-based quadruples that incorporate temporal information and can be formally expressed as $\mathcal{G} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$. Here, $\mathcal{E}$ represents a finite set of entities with cardinality $|\mathcal{E}|$; $\mathcal{R}$ denotes a finite set of relations with cardinality $|\mathcal{R}|$; and $\mathcal{T}$ indicates a finite set of timestamps with cardinality $|\mathcal{T}|$. Each fact quadruple in $\mathcal{G}$ can be expressed as $s = (e_s, r, e_o, t)$, where the subject entity e_s and the object entity e_o are connected via a relation r . The timestamp t denotes the time when the triple event (e_s, r, e_o, t) occurs, and t can be specified in units of hours, days or years. For any $t_1, t_2 \in \mathcal{T}$, $t_1 < t_2$ signifies that the event at time t_1 occurs earlier than that at time t_2 .

3.2. Temporal Knowledge Graph Forecasting

This study addresses the entity prediction task. Let $\mathcal{F}$ denote the set of all ground-truth quadruples, and $(e_s, r, e_o, t) \in \mathcal{F}$ represent the target quadruple. For a query event $(e_s, r, ?, t_q)$, the objective is to predict the missing object entity at time t_q , where the set of preceding events before t_q is denoted as $\mathcal{O} = \{(e_i, r_k, e_j, t_l) \in \mathcal{F} | t_l < t_q\}$. During prediction, all entities in $\mathcal{E}$ are treated as candidates; for each candidate entity, the probability of forming a true quadruple is computed based on historical information, and the entity with the highest probability is selected as the final prediction.

Each quadruple contains two entities: a subject entity and an object entity. For modeling convenience, an inverse event is introduced for every factual quadruple (e_s, r, e_o, t) , expressed as (e_o, r^{-1}, e_s, t) , where r^{-1} denotes the inverse relation of r . For example, if r corresponds to “visit,” then r^{-1} represents “be visited by.” Without loss of generality, the present study focuses on the entity prediction task for query events of the form $(e_s, r, ?, t_q)$.

3.3. Inference Subgraph and Relaxed Temporal Neighborhood

Reasoning is conducted by constructing a query-specific inference subgraph, denoted as $\mathcal{G}_{\text{inf}}$. Nodes in the inference subgraph $\mathcal{G}_{\text{inf}}$ are entity-timestamp pairs, denoted as $v = (e, t)$, where $e \in \mathcal{E}$ and $t \in \mathcal{T}$. The core of inference subgraph construction lies in identifying the temporal neighborhood of each node.

In the xERTE model, the inference subgraph $\mathcal{G}_{\text{inf}}$ is a directed graph in which edges strictly follow temporal order—specifically, edges are directed from nodes with later timestamps to those with earlier timestamps. The neighbors identified under this rule are termed strict temporal neighbors. For a node $v = (e, t) \in \mathcal{G}_{\text{inf}}$, its one-hop strict prior neighbors and one-hop strict posterior neighbors are defined as follows:

$$\mathcal{N}_{v=(e_i,t)}^{\text{strict}} = \{(e_j, t') | (e_i, r_k, e_j, t') \in \mathcal{F} \wedge (t' < t)\} \quad (1)$$

$$\overline{\mathcal{N}}_{v=(e_i,t)}^{\text{strict}} = \{(e_j, t') | (e_j, r_k, e_i, t') \in \mathcal{F} \wedge (t' > t)\} \quad (2)$$

In Equations (1) and (2), $\mathcal{N}_{v=(e_i,t)}^{\text{strict}}$ and $\overline{\mathcal{N}}_{v=(e_i,t)}^{\text{strict}}$ can be abbreviated as $\mathcal{N}_v^s$ and $\overline{\mathcal{N}}_v^s$, respectively.

However, the strict temporal neighbor constraint neglects an essential aspect of human cognitive reasoning—when the temporal gap between two historical events falls within a certain threshold, their chronological order can be disregarded without significantly affecting inference. To accommodate this characteristic, a temporal relaxation factor δ is introduced to moderately relax the rigid temporal order constraint, thereby yielding relaxed temporal neighbor nodes. For a node $v = (e, t) \in \mathcal{G}_{\text{inf}}$, its one-hop relaxed prior neighborhood and one-hop relaxed posterior neighborhood are defined as follows:

$$\mathcal{N}_{v=(e_i,t)}^{\text{relaxed}} = \{(e_j, t') \mid (e_i, r_k, e_j, t') \in \mathcal{F} \wedge (t' < t + \delta) \wedge (t' < t_q)\} \quad (3)$$

$$\overline{\mathcal{N}}_{v=(e_i,t)}^{\text{relaxed}} = \{(e_j, t') \mid (e_j, r_k, e_i, t') \in \mathcal{F} \wedge (t' > t - \delta) \wedge (t' \leq t_q)\} \quad (4)$$

In Equations (3) and (4), $\mathcal{N}_{v=(e_i,t)}^{\text{relaxed}}$ and $\overline{\mathcal{N}}_{v=(e_i,t)}^{\text{relaxed}}$ can be abbreviated as $\mathcal{N}_v^r$ and $\overline{\mathcal{N}}_v^r$, respectively. When the relaxation factor $\delta = 0$, $\mathcal{N}_v^s = \mathcal{N}_v^r$ and $\overline{\mathcal{N}}_v^s = \overline{\mathcal{N}}_v^r$. After incorporating the temporal relaxation factor δ , edges in the inference subgraph $\mathcal{G}_{\text{inf}}$ are permitted to extend from nodes with slightly earlier timestamps to those with slightly later ones. This adjustment preserves the associative information among events whose temporal intervals fall near the threshold, thereby improving the completeness of information captured in the inference subgraph. It is worth noting that δ is a data-dependent hyperparameter, and its optimal value is directly correlated with the characteristics of the dataset—as a temporal interval threshold, its optimal value undergoes significant changes with the temporal granularity of the dataset (e.g., year, day). When $\delta > 0$, the model can capture richer inference path information, yet it inevitably introduces noise; thus, the optimal selection of δ is essentially a trade-off between noise interference and the performance gains derived from expanding the sampling scope.

Figure 2 illustrates a partial segment of the inference subgraph $\mathcal{G}_{\text{inf}}$ corresponding to the query $(e_q, r_q, ?, t_q)$ under relaxed temporal relations. In the figure, each ground-truth quadruple is represented by the entity at the tail of an arrow, the relation labeled on the arrow, and the entity–timestamp pair at the arrowhead. The corresponding quadruples are (e_q, r_0, e_0, t_4) , (e_0, r_1, e_2, t_1) , (e_0, r_1, e_1, t_2) , (e_0, r_2, e_1, t_3) , and (e_0, r_3, e_2, t_3) , respectively. Among these timestamps, t_1, t_2, t_3 and t_4 are posterior to t_q ; $t_1 + \delta, t_2 + \delta$ and $t_3 + \delta$ are posterior to t_4 .

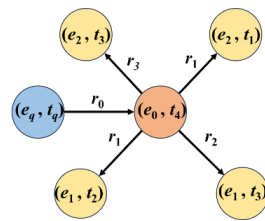


Figure 2. Partial view of the inference subgraph $\mathcal{G}_{\text{inf}}$ for the query $(e_q, r_q, ?, t_q)$ under relaxed temporal relations. In the figure, the blue node represents the initial node (e_q, t_q) obtained from the query; the orange nodes denote the one-hop strict prior neighbor node (e_0, t_4) of the initial node; and the yellow nodes represent the one-hop relaxed prior neighbor nodes of node (e_0, t_4) .

4. Proposed Method

4.1. Framework

This section presents the Subgraph Reasoning Model based on Relaxed Temporal Relations (SR-RTR). Starting from the subject entity e_q in the query $q = (e_q, r_q, ?, t_q)$, the model incrementally constructs an inference subgraph $\mathcal{G}_{\text{inf}}$ by sampling relaxed prior neighbors and ultimately predicts the missing object entity. The reasoning mechanism of SR-RTR comprises two primary modules: subgraph expansion and subgraph pruning. The

expansion module samples relaxed prior neighbors to grow the subgraph, whereas the pruning module applies an attention-based neighbor feature aggregation mechanism to compute attention scores for nodes and edges, thereby filtering key information. Figure 3 illustrates the overall reasoning workflow.

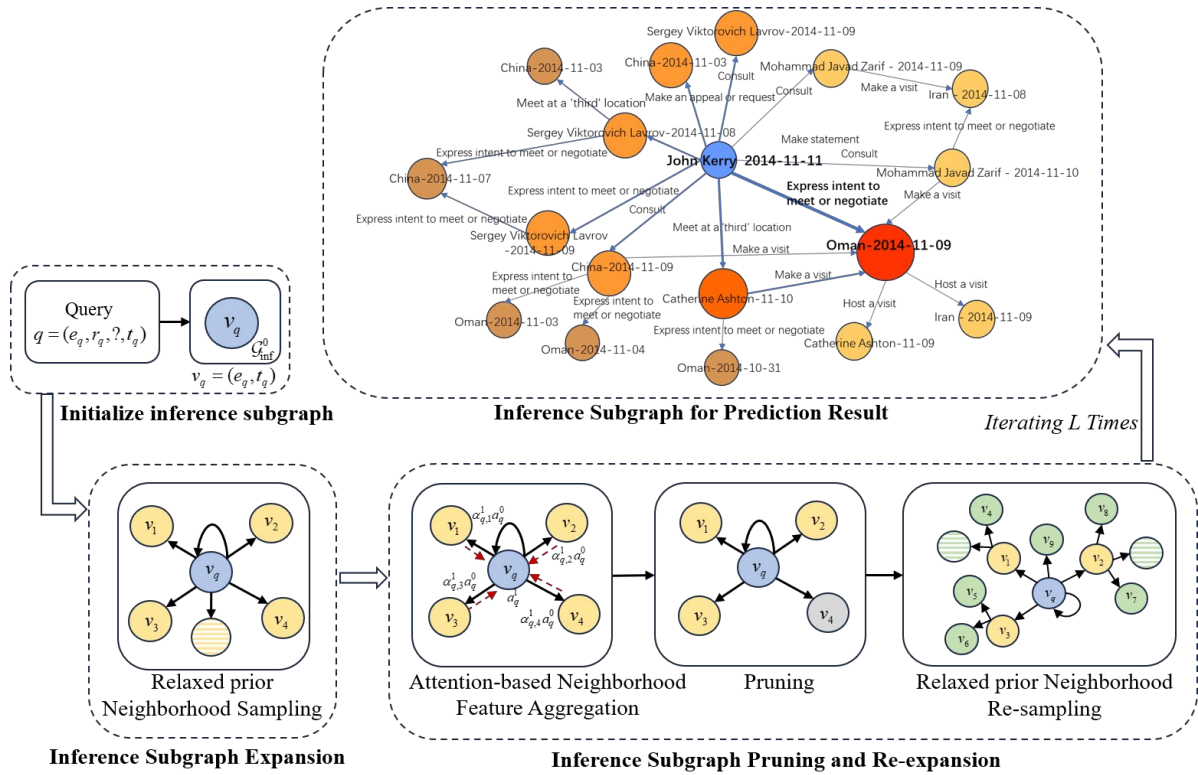


Figure 3. Demonstration of the Subgraph Inference Process. In the figure, the blue node represents the initial node v_q obtained from the query; the yellow nodes denote the one-hop strict prior neighbor candidate nodes of the initial node v_q ; the gray nodes represent the nodes pruned due to low contribution scores; and the green nodes represent the one-hop relaxed prior neighbor candidate nodes of node v_i retained after pruning; $\odot$ and $\ominus$ denote unsampled nodes; $\alpha_{i,j}^l$ represents the attention score of the edge between node v_i and its relaxed prior neighbor v_j during the l -th step of inference; α_i^l denotes the attention score of node v_i during the l -th step of inference. All edge arrows in $\mathcal{G}_{inf}$ point from the source node to its prior relaxed neighbor node.

Initialization: The inference subgraph is first initialized by extracting the subject entity e_q and query timestamp t_q from the query q . The entity–timestamp pair (e_q, t_q) is designated as the initial node of $\mathcal{G}_{inf}$, serving as the starting point for subsequent subgraph expansion.

Subgraph Expansion: Based on the initial node $v_q = (e_q, t_q)$, all quadruples that satisfy the relaxed temporal constraints within the TKG are sampled according to time-dependent exponential weighting. From each sampled quadruple, the object entity e_j and its timestamp t_j are extracted to form a new node $v_j = (e_j, t_j)$, which is then added to $\mathcal{G}_{inf}$. An edge labeled with relation r_k is created from v_q to v_j .

Attention-Based Pruning: To regulate the growth of $\mathcal{G}_{inf}$, node and relation embeddings within the subgraph are used to compute attention scores for both nodes and edges through an attention-based feature aggregation mechanism. Edges with lower scores are pruned to remove redundant or less informative connections, while edges with higher scores are retained to preserve critical reasoning paths.

Re-Expansion and Iteration: The retained nodes in the pruned subgraph are treated as new starting points for further relaxed prior neighbor sampling and subsequent pruning. This alternating process of expansion and pruning is repeated iteratively.

Final Prediction: After L iterative rounds of expansion and pruning, the entity associated with the highest attention score is selected as the predicted object entity for the query. The resulting inference subgraph $\mathcal{G}_{\text{inf}}$ provides a transparent and interpretable explanation for the model's reasoning outcome. The algorithm framework of SR-RTR is as shown in Algorithm 1.

Algorithm 1: SR-RTR Framework

Input: Temporal Knowledge Graph $\mathcal{G}$, query q , total number of inference iteration steps L , temporal relaxation factor δ , number of edges to keep per step K , number of sampled neighbors for each node S

Output: The predicted object entity e_{pred}

1. Initialize inference subgraph $\mathcal{G}_{\text{inf}}$ with only the query node $v_q = (e_q, t_q)$
 2. Initialize node attention scores: $a_{v_q}^0 = 1$, and $a_v^0 = 0$ for all other nodes
 3. Initialize embeddings $\mathbf{h}_v^0$ for all nodes v (using Equations (7) and (8))
 4. Initialize embeddings $\mathbf{r}_k^0$ for all relations r_k
 5. // Iterative Reasoning
 6. **for** $l = 1$ **to** L **do**:
 7. $\mathcal{G}_{\text{exp}} = \text{Subgraph Expansion}(\mathcal{G}_{\text{inf}}, \mathcal{G}, \delta, S)$
 8. $(\mathcal{G}_{\text{inf}}, a_v^l, \alpha_{vu}^l, \mathbf{h}_v^l, \mathbf{r}_k^l) = \text{Subgraph_Prune}(\mathcal{G}_{\text{exp}}, q, \mathbf{h}_v^{l-1}, \mathbf{r}_k^{l-1}, a_v^{l-1}, \delta, K)$
 9. // Final Prediction
 10. **for** entity e_i in $\mathcal{E}$ **do**:
 11. $a_{e_i}(q) = 0$
 12. // Aggregate attention scores for this entity across all timestamps
 13. **for** node $v = (e_i, t)$ in $\mathcal{G}_{\text{inf}}$ **do**:
 14. $a_{e_i}(q) = \text{sum}(a_v(q) \mid v(e) = e_i)$ // Equation (15)
 15. $e_{\text{pred}} = \text{ArgMaxAttentionScore}(e_i)$
 16. **return** e_{pred}
-

4.2. Inference Subgraph Expansion

The expansion of the inference subgraph is accomplished by sampling the one-hop relaxed prior neighbor nodes of the initial node v_q . For a given query q , the entity–timestamp pair $v_q = (e_q, t_q)$ serves as the initial node of the inference subgraph $\mathcal{G}_{\text{inf}}$. Since the outcome of an event is influenced by all preceding events, the timestamps t' of sampled prior neighbors must satisfy the constraint $t' < t_q$.

During the initialization phase of $\mathcal{G}_{\text{inf}}$, only the initial node v_q is present. According to the definition of one-hop relaxed prior neighbors in Equation (3), the timestamp of v_q is t_q , and the timestamps of its neighbors satisfy $(t' < t_q + \delta) \wedge (t' < t_q) = t' < t_q$. Consequently, the one-hop strict prior neighbor coincides with its one-hop relaxed prior neighbor set, i.e., $\mathcal{N}_{v_q}^s = \mathcal{N}_{v_q}^r$. At the first inference step ($l = 1$), the candidate sampling nodes correspond exclusively to the one-hop strict prior neighbors of v_q . When $l > 1$, $\mathcal{G}_{\text{inf}}$ has undergone $l - 1$ rounds of expansion; aside from the initial node v_q , all other nodes $v = (e_i, t)$ in the subgraph possess timestamps t_i earlier than t_q . Sampling their one-hop relaxed prior neighbors as candidate nodes allows the inclusion of additional events occurring close to t , thereby enriching the temporal context for reasoning.

Let the set S_v comprise all quadruples connecting node v in $\mathcal{G}_{\text{inf}}$ to its one-hop relaxed prior neighbor set $\mathcal{N}_v^r$. Denote the cardinality of S_v as $|S_v|$, and an arbitrary quadruple $s_v \in S_v$ can be expressed as:

$$s_{v=(e_i, t)} = \{(e_i, r_k, e_j, t') \in \mathcal{F} \wedge (t' < t + \delta) \wedge (t' < t_q)\} \quad (5)$$

Because $|S_v|$ is typically large, sampling is applied to reduce computational overhead during each expansion step. Only a subset of S_v is retained to update $\mathcal{G}_{\text{inf}}$. The sampled one-hop relaxed prior and posterior neighbor sets of node v are denoted as $\hat{\mathcal{N}}_v^r$ and $\hat{\mathcal{N}}_v^r$, respectively. Considering that events temporally closer to the current timestamp t exert a stronger influence on prediction, a time-dependent exponential weighting strategy is adopted for sampling, as defined in Equation (6):

$$P(s_{v=(e_i,t)} = (e_i, r_k, e_j, t_j)) = \frac{\exp(-|t_j - t|)}{\sum_{(e_i, r_l, e_m, t_m) \in S_v} \exp(-|t_m - t|)} \quad (6)$$

where $\Delta t = |t' - t|$ measures the temporal interval between the candidate node's timestamp t' and the current node's timestamp t , allowing for the sampling of events that occur slightly later than t . Equation (6) ensures that the probability distribution over the candidate set S_v satisfies the normalization property. Furthermore, as the temporal interval $\Delta t \rightarrow \infty, P(s_v) \rightarrow 0$, thereby enabling the model to prioritize the sampling of temporally proximate events. Through this strategy, the sampled relaxed prior neighbor nodes are incorporated into the relaxed prior neighbor set $\hat{\mathcal{N}}_v^r$.

The xERTE samples nodes from the one-hop strict prior neighborhood $\mathcal{N}_v^s$, where the occurrence time of events corresponding to the sampled nodes satisfies $t' < t$; SR-RTR samples nodes from the one-hop relaxed prior neighborhood $\mathcal{N}_v^r$, where the occurrence time of events corresponding to the sampled nodes satisfies $t' < t + \delta$. When the temporal relaxation factor $\delta = 0$, $t' < t + 0$ is completely equivalent to $t' < t$, so SR-RTR and xERTE have exactly the same sampling neighborhood during the inference subgraph expansion phase. The subgraph expansion workflow of SR-RTR is illustrated in Algorithm 2.

Algorithm 2: Subgraph_Expansion

Input: Current inference subgraph $\mathcal{G}_{\text{inf}}$, temporal Knowledge Graph $\mathcal{G}$, temporal relaxation factor δ , number of sampled neighbors for each node S

Output: Expanded subgraph $\mathcal{G}_{\text{exp}}$

1. $\mathcal{G}_{\text{exp}} = \mathcal{G}_{\text{inf}}$ // Copy current subgraph
 2. **for each** node $v = (e, t)$ in $\mathcal{G}_{\text{inf}}$ **do**:
 3. // Find all one-hop relaxed prior neighbors of v
 4. $\mathcal{N}_{v=(e_i,t)}^{\text{relaxed}} = \{(e_j, t') | (e_i, r_k, e_j, t') \in \mathcal{F} \wedge (t' < t + \delta) \wedge (t' < t_q)\}$ // Equation (3)
 5. // Compute sampling probability
 6. $s_{v=(e_i,t)} = \{(e_i, r_k, e_j, t') \in \mathcal{F} \wedge (t' < t + \delta) \wedge (t' < t_q)\}$ // Equation (5)
 7. $P(s_{v=(e_i,t)} = (e_i, r_k, e_j, t_j)) = \frac{\exp(-|t_j - t|)}{\sum_{(e_i, r_l, e_m, t_m) \in S_v} \exp(-|t_m - t|)}$ // Equation (6)
 8. // Sample a subset $S_{v_sampled}$ from S_v
 9. $S_{v_sampled} = \text{Sample}(S_v, P, S)$
 10. **for each** quadruple (e, r_k, e_j, t_j) in $S_{v_sampled}$ **do**:
 11. $u = (e_j, t_j)$
 12. **if** u not in $\mathcal{G}_{\text{exp}}$ **then**
 13. Add node u to $\mathcal{G}_{\text{exp}}$
 14. Add edge (v, r_k, u) to $\mathcal{G}_{\text{exp}}$
 15. **return** $\mathcal{G}_{\text{exp}}$
-

4.3. Inference Subgraph Pruning

To regulate the expansion scale of the inference subgraph $\mathcal{G}_{\text{inf}}$, attention scores for nodes and edges are computed after each expansion step based on the embedding repre-

sentations of the sampled nodes and relations. This computation is performed through attention-based neighbor feature aggregation, whereby edges with lower scores are pruned and those with higher scores are retained. This selective filtering both controls the size of the subgraph and ensures that the most informative connections are preserved.

4.3.1. Embedding of Entities and Relations

Because quadruples in TKGs encapsulate dynamic temporal information, it is essential to integrate temporal features into the embedding representations. The SR-RTR model employs a temporal encoding strategy [19] to encode temporal information, defined in Equation (7):

$$\mathbf{F}(t) = \sqrt{1/d_T} [\cos(\omega_1 t + \phi_1), \dots, \cos(\omega_{d_T} t + \phi_{d_T})] \quad (7)$$

where $\mathbf{F}(t)$ denotes the temporal embedding of timestamp t with dimension d_T ; $\omega = (\omega_1, \dots, \omega_{d_T})$ and $\Phi = (\phi_1, \dots, \phi_{d_T})$ represent learnable frequency and phase parameters, respectively. Given that the semantic characteristics of entities vary over time, the embedding representation of an entity $e_i \in \mathcal{E}$ is decomposed into two components: a static embedding and a temporal embedding, as shown in Equation (8):

$$\mathbf{e}_i(t) = [\bar{\mathbf{e}}_i \parallel \mathbf{F}(t)]^T \in R^{d_s+d_T} \quad (8)$$

where $\bar{\mathbf{e}}_i \in R^{d_s}$ represents the static embedding with dimension d_s . Since relation semantics are generally stable over time, the embedding representation of a relation $r_k \in \mathcal{R}$ is modeled solely through a static embedding, denoted as $\mathbf{r}_k$.

4.3.2. Attention Based Relaxed Neighborhood Aggregation

To more effectively identify key reasoning chains associated with the query q within the inference subgraph $\mathcal{G}_{\text{inf}}$, this study extends the Temporal Relational Graph Attention (TRGA) layer of the xERTE model and introduces the Relaxed Temporal Relational Graph Attention (RTRGA) layer. The improvement primarily enables information propagation from relaxed prior neighbors to their corresponding relaxed posterior neighbors, aligning with the relaxed temporal constraints of the inference subgraph. The RTRGA layer takes as input the embedding representations of entities and relations within the inference subgraph, aggregates local neighborhood information through an attention mechanism, and outputs the updated hidden-layer embeddings of entities and relations. In addition, it computes the attention score of each edge (v, r_k, u) , which reflects its importance in the reasoning process conditioned on the query q .

Let u denote a sampled one-hop relaxed prior neighbor of node v , i.e., $u \in \hat{\mathcal{N}}_v^{\text{r}}$. Since u and v may be connected through multiple relations r_k , and the importance of different relational edges varies, the attention score of edge (v, r_k, u) with respect to query q is defined as shown in Equation (9):

$$e_{vu}^l(q, r_k) = \mathbf{W}_s^l(\mathbf{h}_v^{l-1} \parallel \mathbf{r}_k^{l-1} \parallel \mathbf{h}_{e_q}^{l-1} \parallel \mathbf{r}_q^{l-1}) \mathbf{W}_o^l(\mathbf{h}_u^{l-1} \parallel \mathbf{r}_k^{l-1} \parallel \mathbf{h}_{e_q}^{l-1} \parallel \mathbf{r}_q^{l-1}) \quad (9)$$

where $\mathbf{r}_k$ denotes the static embedding of relation r_k , and $\mathbf{r}_q$ represents the static embedding of the query relation r_q ; $\mathbf{h}_v^{l-1}$ is the hidden-layer embedding of node v at the $(l-1)$ -th inference step. When $l=1$, $\mathbf{h}_v^0 = \mathbf{W}_v \mathbf{e}_i(t) + \mathbf{b}_v$, and $\mathbf{W}_v$ and $\mathbf{b}_v$ are the initial weight matrix and initial bias matrix for node v , respectively; $\mathbf{W}_s^l$ and $\mathbf{W}_o^l$ are used to capture the feature dependency between query q and nodes. The attention scores of all edges (v, r_k, u) connected to node v are normalized using the softmax function, as shown in Equation (10):

$$\alpha_{vu}^l(q, r_k) = \frac{\exp(e_{vu}^l(q, r_k))}{\sum_{w \in \hat{\mathcal{N}}_v^{\text{r}}} \sum_{r_z \in \mathcal{R}_{vw}} \exp(e_{vw}^l(q, r_z))} \quad (10)$$

where $\hat{\mathcal{N}}_v^r$ is the set of sampled one-hop relaxed prior neighbors of node v , and $\mathcal{R}_{vw}$ is the set of relations connecting node v and node w .

As illustrated in Figure 4, the aggregated neighbor representation $\tilde{\mathbf{h}}_v^l(q)$ of node v at step l is obtained by the weighted summation of the embedding representations of the sampled prior relaxed neighbors u_k at step $l - 1$, with the weights corresponding to the normalized attention scores of the associated edges. This process is described in Equation (11):

$$\tilde{\mathbf{h}}_v^l(q) = \sum_{u_k \in \hat{\mathcal{N}}_v^r} \sum_{r_k \in \mathcal{R}_{vu}} \alpha_{vu_k}^l(q, r_k) \mathbf{h}_{u_k}^{l-1}(q) \quad (11)$$

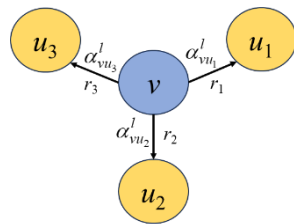


Figure 4. Schematic diagram of node aggregating embedding representations of prior relaxed neighbor nodes. α_{v,u_i}^l represents the attention score of the edge between node v and its relaxed prior neighbor u_i during the l -th step of inference. The blue node in the figure represents the current node v , and the orange nodes denote the one-hop relaxed prior nodes u_i corresponding to node v .

After fusing the node embedding from the previous step with the aggregated neighbor representation, the resulting fused vector is passed through a fully connected layer with LeakyReLU activation, yielding the inferential embedding representation of node v at step l .

$$\mathbf{h}_v^l(q) = \sigma\left(\mathbf{W}_h^l\left(\gamma \mathbf{h}_v^{l-1}(q) + (1 - \gamma) \tilde{\mathbf{h}}_v^l(q)\right) + \mathbf{b}_h^l\right) \quad (12)$$

Among the parameters, γ is a hyperparameter, while $\mathbf{W}_h^l$ and $\mathbf{b}_h^l$ are learnable parameters. Concurrently, the embedding representation of relation r_k is updated as follows:

$$\mathbf{r}_k^l(q) = \mathbf{W}_h^l \mathbf{r}_k^{l-1}(q) + \mathbf{b}_h^l \quad (13)$$

The computed edge attention scores $\alpha_{vu}^l(q, r_k)$, node embedding representations $\mathbf{h}_v^l(q)$, and relation embedding representations $\mathbf{r}_k^l(q)$ are used for subsequent attention propagation and subgraph pruning. When the RTRGA layer calculates the attention scores of nodes and edges in the inference subgraph, the prior and posterior nodes of node v in the graph are directly determined by the sampling results of the inference subgraph expansion phase. When the relaxation factor $\delta = 0$, the sampling neighborhoods of SR-RTR and xERTE are completely consistent, and SR-RTR degenerates into xERTE.

4.3.3. Attention Propagation and Pruning

Once the edge attention scores are obtained, the attention score of node v with respect to query q at step l is further computed. As shown in Figure 5, this score is derived by aggregating the attention scores from the previous step of the one-hop relaxed posterior nodes u_i of node v , where the aggregation coefficients are the attention scores of the corresponding edges. The specific calculation is detailed in Equation (14):

$$a_v^l(q) = \sum_{u_i \in \hat{\mathcal{N}}_v^r} \sum_{r_i \in \mathcal{R}_{uv}} \alpha_{u_i v}^l(q, r_i) a_{u_i}^{l-1}(q) \quad (14)$$

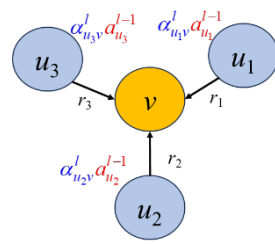


Figure 5. Schematic diagram of propagation of posterior neighbor attention scores to node v . The orange node in the figure represents the current node v , and the blue nodes denote the one-hop relaxed posterior nodes u_i corresponding to node v .

Since the same entity may correspond to multiple nodes with different timestamps, the final attention score for entity e_i is obtained by summing the attention scores of all nodes in the inference subgraph $\mathcal{G}_{\text{inf}}$ whose entity component is e_i , as specified in Equation (15):

$$a_{e_i}^l(q) = \text{sum}(a_v^l(q) \mid v(e) = e_i) \quad (15)$$

To regulate the scale of the inference subgraph, the contribution score of each edge within the subgraph is calculated. The higher the contribution score, the more it contributes to the prediction task. The contribution score of each edge (v, r_k, u) in the inference subgraph is defined as follows:

$$c_{vu}(q, r_k) = \alpha_{vu}^l(q, r_k) a_v^l(q) \quad (16)$$

After each inference step, the top- K edges with the highest contribution scores are retained, while edges with lower scores are discarded, allowing for subgraph expansion in the next inference step.

After L iterations of subgraph expansion and pruning, the attention score for each entity e_i in the $\mathcal{G}$ is determined. If e_i appears in the inference subgraph $\mathcal{G}_{\text{inf}}$, its attention score is computed according to Equation (15); otherwise, its attention score is set to 0.

The quadruples of the $\mathcal{G}$ are partitioned into training, validation, and test sets, with the temporal constraint that the timestamps of the training set are earlier than those of the validation set, and the validation set timestamps precede those of the test set. For training, the binary cross-entropy loss function is employed. To avoid overfitting and parameter redundancy, L2 regularization is adopted in this work to impose constraints on all learnable parameters. Ultimately, the model predicts the entity with the highest attention score as the predicted object entity for the query q . The subgraph expansion workflow of SR-RTR is illustrated in Algorithm 3.

4.4. Computational Complexity Analysis

This section analyzes the computational complexity of the SR-RTR model. The SR-RTR model consists of L rounds of inference iterations, and each iteration comprises two core modules: subgraph expansion and subgraph pruning. We define K as the number of edges retained in the inference subgraph after each pruning step, S as the sampling size of relaxed prior neighbors during each subgraph expansion, and d as the feature embedding dimension.

In the subgraph expansion phase, the model samples S relaxed prior neighbors from the nodes corresponding to K edges, with a computational complexity of $O(K \cdot S)$ for this phase. The subgraph pruning phase is centered on the Relaxed Temporal Relational Graph Attention (RTRGA) layer: the computational complexity of calculating attention scores for edges and nodes is $O(K \cdot S \cdot d)$ (where d is a constant-level feature embedding dimension), the complexity of normalization and feature aggregation operations is $O(K \cdot S)$, and the final step of filtering and pruning the top- K edges has a computational complexity of $O((K \cdot S) \cdot \log K)$.

Therefore, for a given number of inference iterations L , the total complexity of the model can be simplified to $O(L \cdot K \cdot S \cdot \log K)$. Since L , K , and S are all fixed hyperparameters, the model complexity exhibits constant-order growth.

Algorithm 3: Subgraph_Prune

Input: Expanded subgraph $\mathcal{G}_{\text{exp}}$, query q , node hidden representation at the $(l-1)$ -th step $\mathbf{h}_v^{l-1}$, relation hidden representation at the $(l-1)$ -th step $\mathbf{r}_k^{l-1}$, node attention score at the $(l-1)$ -th step a_v^{l-1} , temporal relaxation factor δ , number of edges to keep per step K
Output: Pruned subgraph $\mathcal{G}_{\text{inf}}$, node attention score at the l -th step a_v^l , edge attention score α_{vu}^l , node hidden representation at the l -th step $\mathbf{h}_v^l$, relation hidden representation at the l -th step $\mathbf{r}_k^l$

1. // Compute edge attention and aggregate neighbor representations
 2. **for each** node v in $\mathcal{G}_{\text{exp}}$ **do**:
 3. **for each** one-hop relaxed prior neighbor u of v via relations r_k in $\mathcal{R}_{vu}$ **do**:
 4. // Compute unnormalized edge attention
 5.
$$e_{vu}^l(q, r_k) = \mathbf{W}_s^l(\mathbf{h}_v^{l-1} \parallel \mathbf{r}_k^{l-1} \parallel \mathbf{h}_{e_q}^{l-1} \parallel \mathbf{r}_q^{l-1}) \mathbf{W}_o^l(\mathbf{h}_u^{l-1} \parallel \mathbf{r}_k^{l-1} \parallel \mathbf{h}_{e_q}^{l-1} \parallel \mathbf{r}_q^{l-1})$$
 // Equation (9)
 6.
$$\alpha_{vu}^l(q, r_k) = \frac{\exp(e_{vu}^l(q, r_k))}{\sum_{w \in \hat{\mathcal{N}}_v^r \sum_{z \in \mathcal{R}_{vw}} \exp(e_{vw}^l(q, r_z))}$$
 // Equation (10)
 7. // Update node embedding
 8.
$$\tilde{\mathbf{h}}_v^l(q) = \sum_{u_k \in \hat{\mathcal{N}}_v^r \sum_{r_k \in \mathcal{R}_{vu}} \alpha_{vu_k}^l(q, r_k) \mathbf{h}_{u_k}^{l-1}(q)$$
 // Equation (11)
 9.
$$\mathbf{h}_v^l(q) = \sigma(\mathbf{W}_h^l(\gamma \mathbf{h}_v^{l-1}(q) + (1 - \gamma) \tilde{\mathbf{h}}_v^l(q)) + \mathbf{b}_h^l)$$
 // Equation (12)
 10. // Update relation embeddings
 11.
$$\mathbf{r}_k^l(q) = \mathbf{W}_h^l \mathbf{r}_k^{l-1}(q) + \mathbf{b}_h^l$$
 // Equation (13)
 12. // Propagate node attention via posterior neighbors
 13. **for each** node v in $\mathcal{G}_{\text{exp}}$ **do**:
 14. **for each** one-hop relaxed posterior neighbor u of v via relations r_k in $\mathcal{R}_{vu}$ **do**:
 15.
$$a_v^l(q) = \sum_{u_i \in \hat{\mathcal{N}}_v^r \sum_{r_i \in \mathcal{R}_{uv}} \alpha_{u_i v}^l(q, r_i) a_{u_i}^{l-1}(q)$$
 // Equation (14)
 16. // Prune to Top-K edges by contribution score
 17. $c_{vu}(q, r_k) = \alpha_{vu}^l(q, r_k) a_v^l(q)$ // Compute edge contribution Equation (16)
 18. Select Top-K edges with highest values $c_{vu}(q, r_k)$, build $\mathcal{G}_{\text{inf}}$ with these edges and their nodes
 19. **return** $\mathcal{G}_{\text{inf}}$, a_v^l , α_{vu}^l , $\mathbf{h}_v^l$, $\mathbf{r}_k^l$
-

It is important to note that the inference subgraph of the SR-RTR model only samples query-related nodes from the temporal knowledge graph $\mathcal{G}$, and the maximum number of nodes contained in the subgraph is much smaller than the total number of entities $|\mathcal{E}|$ in the TKG. Built upon the xERTE framework and its extension of the TRGA layer, the SR-RTR model has nearly the same computational complexity as xERTE. Although the introduction of the temporal relaxation factor δ expands the scope of candidate sampling nodes, it only adds a constant-time relaxation check. In contrast, baseline models such as RE-Net and RE-GCN often need to process all entities in the TKG, with a computational complexity of $O(|\mathcal{E}| \cdot d)$ that increases significantly as the TKG scale grows.

5. Experiments

This section presents experiments conducted to validate the effectiveness of the proposed SR-RTR model.

5.1. Datasets

The experiments are performed on benchmark datasets for TKGs, specifically the YAGO dataset [23] and the Integrated Crisis Early Warning System (ICEWS) dataset [24]. The YAGO dataset is a temporal knowledge base that integrates data from Wikipedia and the English WordNet dataset [25]. For the purpose of this study, a subset of YAGO is used, retaining only year-level temporal granularity while discarding month and day information, consistent with the experimental setup of xERTE. The ICEWS dataset draws from over 100 data sources, covering more than 250 international and regional news events. It includes dynamic data across various domains such as politics, economics, and society, and has become a key benchmark for TKG research. To assess model performance, three subsets of ICEWS are selected: ICEWS14, ICEWS18, and ICEWS05-15, covering events from 2014, 2018, and the period from 2005 to 2015, respectively. Each dataset is split into training, validation, and test sets, adhering to the temporal constraint that the timestamps of the training set precede those of the validation set, and the timestamps of the validation set precede those of the test set. Detailed statistical information for each dataset is provided in Table 1.

Table 1. Statistics of YAGO, ICEWS14, ICEWS18 and ICEWS05-15 datasets.

Dataset	$ \mathcal{E} $	$ \mathcal{R} $	N_{train}	N_{valid}	N_{test}	$N_{timestamp}$	Time Granularity
YAGO	10,038	10	51,205	10,973	10,973	194	year
ICEWS14	7128	230	63,685	13,823	13,222	365	day
ICEWS18	23,033	256	373,018	45,995	49,545	304	day
ICEWS05-15	10,488	251	322,958	69,224	69,147	4017	day

5.2. Baseline

The model proposed in this paper is compared with both static and TKG reasoning models. For static knowledge graph reasoning, the comparison baselines include TransE [11], DistMul [26], and ComplEx [27]; for TKG reasoning, the baselines include TTransE [28], TA-DistMult/TA-TransE [29], DE-Simple [30], TNTComplEx [31], CyGNet [18], RE-Net [17], and xERTE [9].

5.3. Evaluation Criteria and Experimental Setting

The evaluation metrics used in this experiment are hits@k and Mean Reciprocal Rank (MRR). Specifically, for each quadruple (e_s, r, e_o, t) in the test set $\mathcal{G}_{test}$, two queries are constructed: $(?, r, e_o, t)$ and $(e_s, r^{-1}, ?, t)$, where r^{-1} denotes the reciprocal relation of r . For each query, the model ranks all entities in the final inference subgraph based on their attention scores. If the correct entity does not appear in the final inference subgraph, its rank is set to $|\mathcal{E}|$ (i.e., the total number of entities in the dataset). Here, $\text{rank}(e_s)$ represents the predicted rank of the correct entity e_s in the query $(?, r, e_o, t)$, and $\text{rank}(e_o)$ represents the predicted rank of the correct entity e_o in the query $(e_s, r^{-1}, ?, t)$. The calculation formula for MRR is given in Equation (17).

$$\text{MRR} = \frac{1}{2|\mathcal{G}_{test}|} \sum_{q \in \mathcal{G}_{test}} \left(\frac{1}{\text{rank}(e_s)} + \frac{1}{\text{rank}(e_o)} \right) \quad (17)$$

All experiments are conducted in a hardware environment consisting of an Intel Xeon® Gold 5218 CPU @ 2.30GHz (64 cores) (Intel Corporation, Santa Clara, CA, USA), NVIDIA GeForce 3090 GPU (24 GB memory) and 256 GiB RAM (NVIDIA Corporation, Santa Clara, CA, USA). Experiments based on the SR-RTR model are run five times, with each run trained for 10 epochs. The Adam optimizer is used for parameter training, and the learning rate is set to 0.0002. For experiments on the four datasets (YAGO, ICEWS14,

ICEWS0515, and ICEWS18), the maximum number of retained edges is set to 60, 40, 40, and 60, respectively.

5.4. Experimental Results

The optimal values of the inference iteration step L and the temporal relaxation factor δ are determined through experimental validation. Table 2 presents the entity prediction results for the YAGO dataset and the three ICEWS subsets.

Table 2. Prediction results on datasets.

Dataset	YAGO-Filtered				ICEWS14-Filtered				ICEWS18-Filtered				ICEWS0515-Filtered			
Model	MRR	hits@1	hits@3	hits@10	MRR	hits@1	hits@3	hits@10	MRR	hits@1	hits@3	hits@10	MRR	hits@1	hits@3	hits@10
TransE [11]	11.69	10.37	11.96	13.83	22.48	13.36	25.63	41.23	12.24	5.84	12.81	25.10	22.55	13.05	25.61	42.05
DisMult [26]	11.98	10.20	12.31	14.93	27.67	18.16	31.15	46.96	10.17	4.52	10.33	21.25	28.73	19.33	32.19	47.54
ComplEx [27]	12.07	10.42	12.36	14.82	30.84	21.51	34.48	49.58	21.01	11.87	23.47	39.87	31.69	21.44	35.74	52.04
TTransE [28]	5.68	1.42	9.04	11.21	13.43	3.11	17.32	34.55	8.31	1.92	8.56	21.89	15.71	5.00	19.72	38.02
TA-DisMult [29]	11.50	10.21	11.90	13.88	26.47	17.09	30.22	45.41	16.75	8.61	18.41	33.59	24.31	14.58	27.92	44.21
TA-TransE [29]	6.74	2.13	11.01	12.28	17.41	0.00	29.19	47.41	12.59	0.01	17.92	37.38	19.37	1.81	31.34	50.33
DE-Simple [30]	11.73	10.70	12.10	13.51	32.67	24.43	35.69	49.11	19.30	11.53	21.86	34.80	35.02	25.91	38.99	52.75
TNTComplEx [31]	12.00	11.12	12.13	13.57	32.12	23.35	36.03	49.13	21.23	13.28	24.02	36.91	27.54	19.52	30.80	42.86
GyGNet [18]	12.48	11.00	12.66	14.82	32.73	23.69	36.31	50.67	24.93	15.90	28.28	42.61	34.97	25.67	39.09	52.94
RE-Net [17]	54.87	47.51	57.84	65.81	38.28	28.68	41.34	54.52	28.81	19.05	32.44	47.51	42.97	31.26	46.85	63.47
xERTE [9]	53.62	48.53	<u>58.42</u>	60.53	<u>40.79</u>	<u>32.70</u>	<u>45.67</u>	<u>57.30</u>	<u>29.31</u>	<u>21.03</u>	<u>33.51</u>	46.48	<u>46.62</u>	<u>37.84</u>	<u>52.31</u>	<u>63.92</u>
SR-RTR	<u>53.66</u>	<u>48.18</u>	58.77	<u>61.37</u>	41.85	33.74	46.90	58.08	29.75	21.53	34.30	<u>47.02</u>	47.26	38.55	52.94	64.42

The best results are shown in bold, and the second-best results are shown with an underline “_”.

Specifically, all evaluation criteria of SR-RTR are reported as the final results as the average of three independent experiments, whereas the results of all evaluation criteria for the other baseline models are retrieved from their published papers. All results are obtained using a time-dependent filtering scheme, where only the real triples that exist at the time of the query are considered. Experimental results show that the reasoning mechanism with the temporal relaxation factor proposed in this paper outperforms other models overall: compared to the baseline models, it achieves improvements across all evaluation metrics, with its prediction results reaching optimal or near-optimal levels.

The datasets used in this experiment differ in temporal granularity: the YAGO dataset has a temporal granularity of “year,” while the three ICEWS subsets (ICEWS14, ICEWS18, and ICEWS0515) have a temporal granularity of “day.” Considering that the optimal values of the temporal relaxation factor δ correspond to different time granularities, we focus on discussing δ with shorter spans on the YAGO dataset, while we focus on δ with longer spans in the three subsets of the ICEWS dataset. Consequently, separate experiments are conducted for these two categories of datasets to identify the optimal temporal relaxation factor and inference iteration step, followed by result analysis.

5.4.1. Result Analysis on the YAGO Dataset

For the YAGO dataset, the impact of selecting inference steps L on the experimental results is first analyzed. Figure 6a shows that there is no significant difference in time cost between $\delta = 0$ and $\delta = 1$; however, both increase substantially as the reasoning step L increases. Figure 6b demonstrates that the MRR values for $\delta = 0$ and $\delta = 1$ first increase and then decrease with the increase in L , reaching their optimal values when $L = 3$. Furthermore, the MRR value for $\delta = 1$ consistently outperforms that for $\delta = 0$. When $L = 4$, the MRR values for both δ settings decrease, but the reduction is significantly smaller for $\delta = 1$ than for $\delta = 0$. The degradation trend persists as L further increases to 5. These results suggest that as L increases, more prior neighbor nodes introduce new information into the reasoning process, enhancing the MRR value. However, this information also introduces noise, and further

increases in L reduce reasoning accuracy due to the noise. Introducing an appropriate temporal relaxation factor helps significantly mitigate the impact of noise.

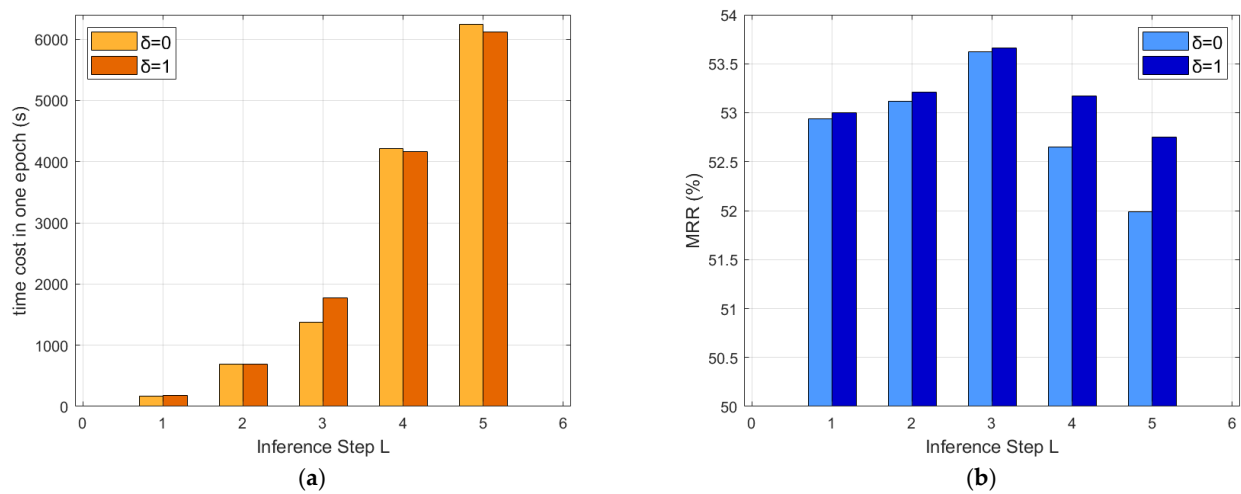


Figure 6. Influence of inference step L values on training time and MRR. (a) Comparison of time cost across different inference step L values in one epoch; (b) Comparison of MRR across different inference step L values.

The experiment further explores the impact of the temporal relaxation factor δ on the experimental results. With the inference steps fixed at $L = 1$, $L = 2$, and $L = 3$, the results for each reasoning evaluation metric at δ values of 0, 1, 2, and 3 are shown in Figure 7. Figure 7a demonstrates that for $L = 1$ and $L = 2$, the MRR values for $\delta = 1$, 2, and 3 are all superior to those for $\delta = 0$. However, when $L = 3$, the MRR value for $\delta = 1$ remains better than for $\delta = 0$, while the MRR values for $\delta = 2$ and $\delta = 3$ are significantly lower. This suggests that as the number of reasoning steps increases, a further increase in δ results in the negative impact of mixed noise outweighing the benefits of expanding the sampling candidate range. As shown in Figure 7b, when $L = 3$, the hits@1 values for $\delta = 1$, 2, and 3 are lower than for $\delta = 0$. However, when $L = 2$, the hits@1 value for $\delta = 1$ (48.58%) and $\delta = 3$ (48.53%) are both equal to or greater than the hits@1 value for $\delta = 0$ at $L = 3$ (48.53%). This indicates that introducing the temporal relaxation factor can reduce the number of reasoning steps, improving the search efficiency of the hits@1 metric. Figure 7c,d show that for hits@3 and hits@10 metrics, the advantage of the temporal relaxation factor is more pronounced, consistently outperforming the corresponding $\delta = 0$ metrics across different inference steps L .

To determine the optimal combination of the inference step L and temporal relaxation factor δ on the YAGO dataset, we fixed L at 1, 2, and 3 and analyzed the impact of their interaction on model performance by combining Pearson correlation coefficients. When $L = 1$ or 2, δ exerts a weak influence on all metrics, with Pearson correlation coefficients all satisfying $|r| < 0.3$ —which indicates that the subgraph scale is limited under low inference steps, and the relaxation of temporal constraints brought by δ cannot significantly alter model performance. When $L = 3$, the influence of δ on the metrics increases significantly with differentiated directions: it exhibits a strong negative correlation with MRR ($r = -0.82$) and hits@1 ($r = -0.88$), which implies that an excessively large δ will introduce redundant noise and offset the gains from expanding the candidate sampling pool; it shows a moderate negative correlation with hits@3 ($r = -0.45$), as hits@3 has higher fault tolerance for candidate entities and is less affected by noise; and it presents a moderate positive correlation with hits@10 ($r = 0.68$), because the paths obtained through the temporal constraint relaxation introduced by δ can cover more potential correct entities, and hits@10 has

stronger noise tolerance. A comprehensive analysis of the four evaluation metrics reveals that the optimal reasoning performance occurs when $\delta = 1$ and $L = 3$. This combination not only captures valid reasoning paths through moderate temporal relaxation but also avoids performance degradation caused by an excessively large δ , with specific results shown in Table 2.

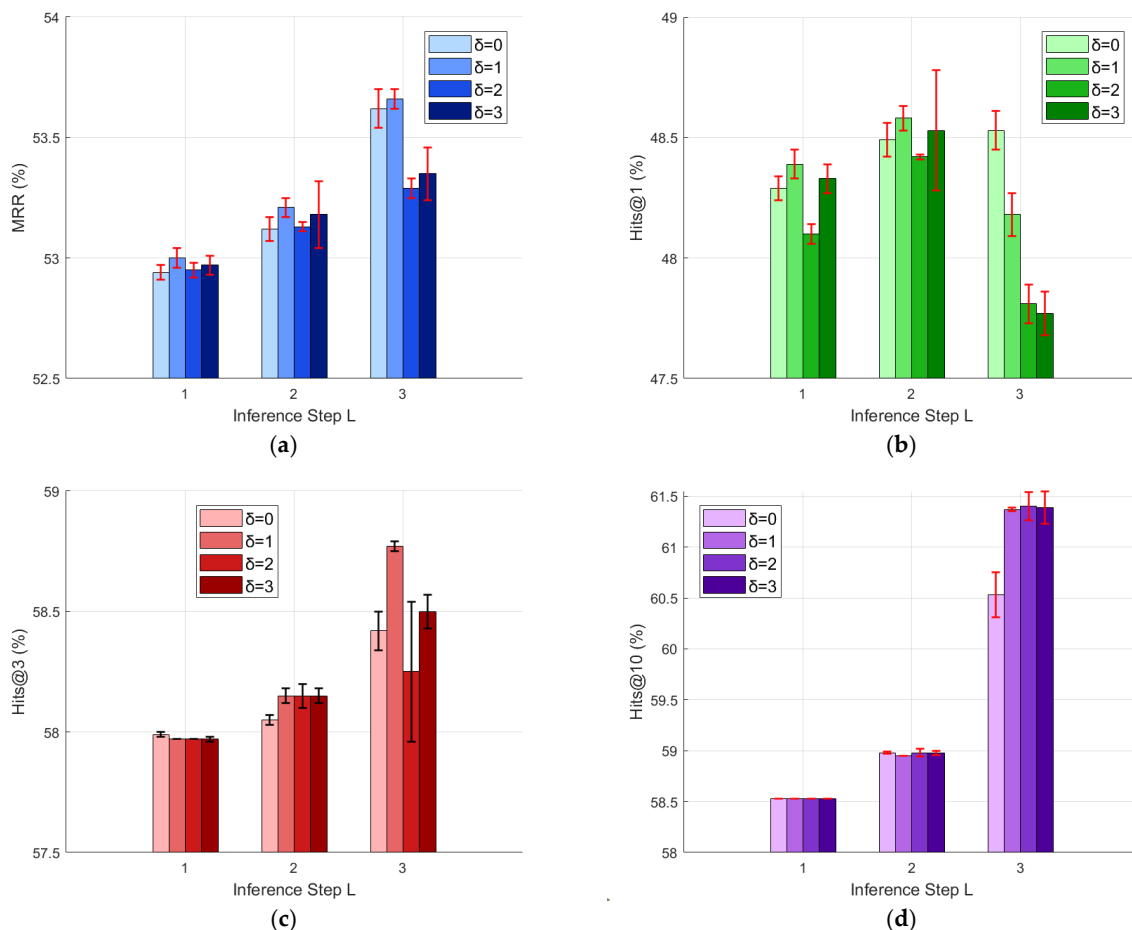


Figure 7. The impact of temporal relaxation factor δ values on MRR and hits@k ($k = 1, 3, 10$) on the YAGO dataset. The black and red vertical lines in the figure represent the standard deviation. (a) Comparison of MRR across different δ values; (b) Comparison of hits@1 across different δ values; (c) Comparison of hits@3 across different δ values; (d) Comparison of hits@10 across different δ values.

5.4.2. Result Analysis on the ICEWS14, ICEWS18, and ICEWS0515 Datasets

Since the ICEWS14 dataset shares a similar source with the ICEWS18 and ICEWS0515 datasets but is smaller in scale, this study focuses on examining the impact of different temporal relaxation factors δ on various evaluation metrics in more detail using this dataset. Existing research has already established conclusions regarding the value of inference step L for the xERTE model, so the experiment limits the comparison to scenarios where $L \leq 3$. Given the finer temporal granularity of the ICEWS14 dataset, the experiment selects δ values of 0, 2, 3, 7, 15, and 30 to explore their effects on the four evaluation metrics, with results presented in Figure 8.

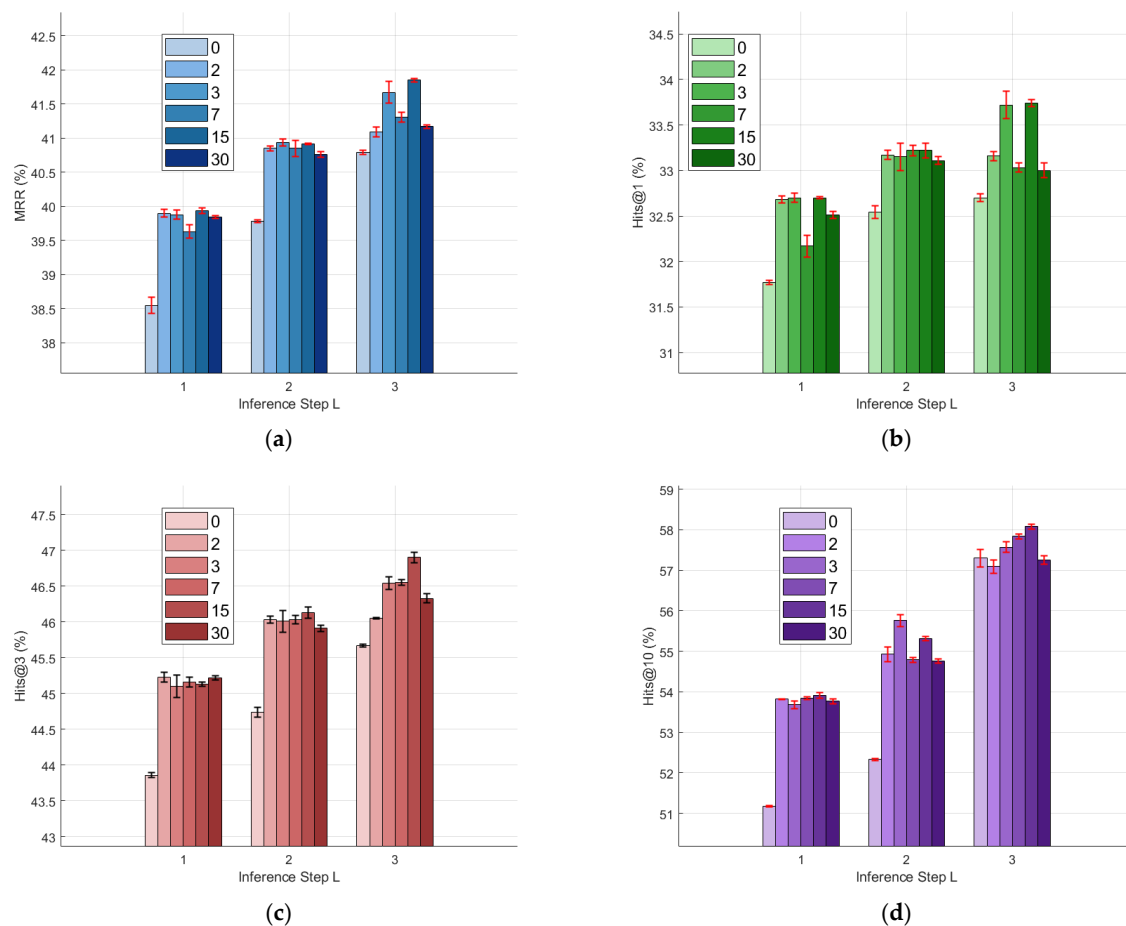


Figure 8. The impact of temporal relaxation factor δ values on MRR and hits@k ($k = 1, 3, 10$) on the ICEWS14 dataset. The black and red vertical lines in the figure represent the standard deviation. (a) Comparison of MRR across different δ values; (b) Comparison of hits@1 across different δ values; (c) Comparison of hits@3 across different δ values; (d) Comparison of hits@10 across different δ values.

Experimental findings show that, except for the hits@1 metric—where the value at $L = 3$ slightly decreases compared to $L = 2$ for $\delta = 2, 7$, and 30 —all other metrics exhibit an upward trend as L increases. In fact, for $\delta > 0$, the metrics MRR, hits@1, and hits@3 at $L = 2$ surpass their corresponding values at $L = 3$ when $\delta = 0$. This suggests that, similar to the pattern observed in the YAGO dataset, introducing a temporal relaxation factor into the reasoning model for ICEWS14 not only improves reasoning accuracy but also enhances reasoning efficiency.

When the inference step $L = 3$, the results of the four evaluation metrics for different relaxation factors δ are shown in Table 3. Experimental results indicate that the hits@10 metric is slightly lower than that at $\delta = 0$ only for $\delta = 2$ and $\delta = 30$; for all other δ values, all metrics outperform those at $\delta = 0$. The variation patterns of different metrics with δ differ: hits@3 and hits@10 first increase and then decrease as δ increases, while hits@1 and MRR show a fluctuating upward trend, with a decrease observed only at $\delta = 7$. This behavior suggests that increasing δ within an appropriate range can expand the candidate set of sampled nodes, thereby improving the multi-hit rate. However, excessively large δ introduces more noise. Thus, selecting an optimal δ is crucial to balance noise interference and the benefits of an expanded sampling range.

Table 3. Prediction results with different temporal relaxation factors δ for the model on ICEWS14 ($L = 3$).

δ	MRR	hits@1	hits@3	hits@10
0	40.93 $\pm$ 0.17	32.84 $\pm$ 0.24	45.86 $\pm$ 0.18	57.32 $\pm$ 0.15
2	41.09 $\pm$ 0.02	33.16 $\pm$ 0.08	46.05 $\pm$ 0.11	57.09 $\pm$ 0.25
3	41.67 $\pm$ 0.12	33.72 $\pm$ 0.11	46.54 $\pm$ 0.11	57.57 $\pm$ 0.12
7	41.31 $\pm$ 0.03	33.03 $\pm$ 0.08	46.55 $\pm$ 0.05	57.84 $\pm$ 0.08
15	41.85 $\pm$ 0.10	33.74 $\pm$ 0.11	46.90 $\pm$ 0.09	58.08 $\pm$ 0.04
30	41.17 $\pm$ 0.04	33.00 $\pm$ 0.05	46.33 $\pm$ 0.04	57.25 $\pm$ 0.08

The best results are shown in bold.

To determine the optimal combination of the number of inference steps L and the temporal relaxation factor δ on the ICEWS14 dataset, we analyze the impact of their interaction on model performance using Pearson correlation coefficients. When $L = 1$, δ shows a strong positive correlation with MRR ($r = 0.78$), hits@3 ($r = 0.83$), and hits@10 ($r = 0.92$), and a moderate positive correlation with hits@1 ($r = 0.65$), due to the small subgraph scale and low noise. When $L = 2$, δ still maintains a strong positive correlation with hits@10 ($r = 0.75$), shifts to a moderate positive correlation with MRR ($r = 0.45$) and hits@3 ($r = 0.61$), and decreases to a weak correlation with hits@1 ($r = 0.28$), indicating that the increase in subgraph depth makes the performance gain of δ more inclined to metrics with higher fault tolerance. When $L = 3$, δ exhibits a moderate positive correlation with all metrics ($r = 0.35$ – 0.58), all metrics achieve optimal values at $\delta = 15$; when δ increases to 30, all evaluation metrics experience a significant decline. This suggests that the subgraph depth is sufficient at this point, and excessive relaxation of temporal constraints is prone to introducing redundant noise, so $\delta = 15$ is needed to balance the expansion of sampling paths and noise suppression. Based on the comprehensive analysis above, compared with the YAGO dataset, the ICEWS14 dataset has a higher graph density, so moderately increasing δ can more fully capture valid inference paths. The model achieves the optimal performance when $L = 3$ and $\delta = 15$. Based on these findings, the parameter combination of $L = 3$ and $\delta = 15$ is chosen for the experiments conducted on the ICEWS18 and ICEWS0515 datasets, with the specific results shown in Table 2.

5.5. Statistical Significance Test

To verify whether the performance of the proposed SR-RTR model exhibits a statistically significant improvement, we employed a two-tailed t-test with a significance level of $\alpha = 0.05$ to examine the significance of differences in four evaluation metrics between the SR-RTR model and the baseline model xERTE. The results of xERTE on the four datasets are the mean values $\pm$ standard deviations of three independent runs, obtained by re-running its publicly available code. These results are slightly different from those reported in the original xERTE paper (see Table 2). The specific results are shown in Table 4 (* indicates a statistically significant improvement): with the exception of no significant differences in the MRR, hits@1, and hits@3 metrics on the YAGO dataset, the hits@10 metric on the YAGO dataset and all metrics on the three subsets of the ICEWS dataset are significantly superior to the corresponding metrics of xERTE. The results of the significance test demonstrate that the SR-RTR model is more effective for the three subsets of the ICEWS dataset. It is hypothesized that this is because the graph density of the three ICEWS subsets is greater than that of the YAGO dataset, and the introduction of the relaxation factor can more fully capture events with a closer time interval to the current time, thereby leading to a more significant improvement in model performance.

Table 4. Two-tailed t-test results (significance level $\alpha = 0.05$) of SR-RTR and XERTE models (YAGO-filtered: $L = 3$, $\delta = 1$; ICEWS14-filtered, ICEWS18-filtered, ICEWS0515-filtered: $L = 3$, $\delta = 15$).

Dataset	Model	MRR	hits@1	hits@3	hits@10
YAGO-filtered	xERTE	53.55 ± 0.08	48.51 ± 0.10	58.42 ± 0.10	60.20 ± 0.22
	SR-RTR	53.66 ± 0.06	48.18 ± 0.13	58.77 ± 0.04	$61.37 \pm 0.02^*$
ICEWS14-filtered	xERTE	40.93 ± 0.17	32.84 ± 0.24	45.86 ± 0.18	57.32 ± 0.15
	SR-RTR	$41.85 \pm 0.10^*$	$33.74 \pm 0.11^*$	$46.90 \pm 0.09^*$	$58.08 \pm 0.04^*$
ICEWS18-filtered	xERTE	29.19 ± 0.10	20.91 ± 0.10	33.40 ± 0.10	46.33 ± 0.13
	SR-RTR	$29.75 \pm 0.10^*$	$21.53 \pm 0.09^*$	$34.30 \pm 0.04^*$	$47.02 \pm 0.06^*$
ICEWS0515-filtered	xERTE	46.54 ± 0.09	37.79 ± 0.09	52.18 ± 0.13	63.83 ± 0.15
	SR-RTR	$47.26 \pm 0.05^*$	$38.55 \pm 0.12^*$	$52.94 \pm 0.08^*$	$64.42 \pm 0.05^*$

Symbol * indicates a statistically significant improvement.

5.6. Case Analysis

After L rounds of expansion and pruning iterations, the inference subgraph $\mathcal{G}_{\text{inf}}$ is obtained. Both the nodes and edges of this subgraph are assigned corresponding attention scores, enabling the tracing of key factual bases for the model's predictions and providing clear explanatory support for its reasoning. This study selects a query case “(John Kerry, Make a visit, ?, 2014-11-11)” from the ICEWS14 test dataset and visualizes the prediction result by presenting the inference subgraph $\mathcal{G}_{\text{inf}}$, as shown in Figure 9. Figure 9 displays a partial view of the inference subgraph, highlighting the top 6 entities with the highest attention scores: the wider the edge, the greater its contribution; the larger and darker the node, the higher its attention score. The largest and darkest node represents the prediction result. The final prediction for this query is Oman, with the primary supporting basis being “(John Kerry, express intent to meet or negotiate, Oman, 2014-11-09)”.

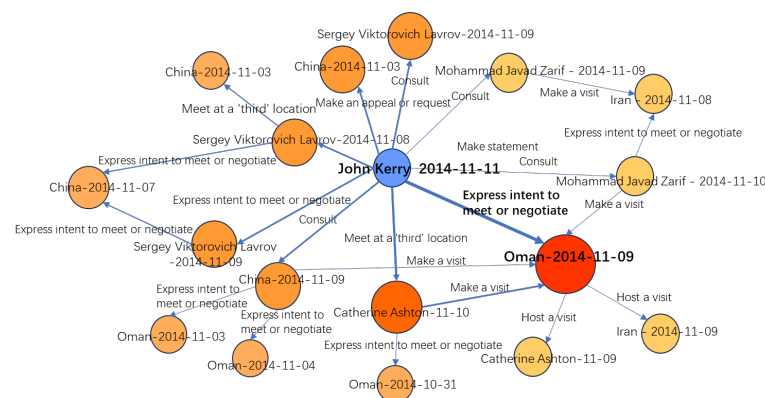


Figure 9. Reasoning subgraph for query (John Kerry, Make a visit, ?, 2014-11-11). The biggest red node represents the object predicted by SR-RTR. The blue node with the entity John Kerry and the timestamp 2014-11-11 represents the given query subject and the query timestamp. The node size and color indicate the value of the node attention score; the larger and darker the node, the higher its attention score. The edges' width indicates the contribution score of the edge; the wider the edge, the greater its contribution. The entity at an arrow's tail, the relation on the arrow, the entity and the timestamp at the arrow's head build a true quadruple.

Inference comparisons were conducted using the SR-RTR model with temporal relaxation factors $\delta = 0$ and $\delta = 15$, respectively, both yielding correct prediction results. When $\delta = 0$, the attention score for the entity Oman was 0.4064; when $\delta = 15$, the attention score for Oman increased to 0.6585. Table 5 presents the edges related to the predicted entity Oman in the final inference subgraph $\mathcal{G}_{\text{inf}}$: at $\delta = 0$, 7 edges were associated with Oman, while

at $\delta = 15$, the number of associated edges increased to 10. The identification of additional relevant clues effectively enhanced the confidence in the prediction results.

Table 5. Relevant edges of the entity Oman in the inference subgraph $\mathcal{G}_{\text{inf}}$ for the query “(John Kerry, make a visit, ?, 2014-11-11)”.

δ	The Relevant Edges Related to Entity Oman in $\mathcal{G}_{\text{inf}}$
0	(John Kerry, Express intent to meet or negotiate, Oman, 2014-11-10) (Iran, Make statement, Oman, 2014-11-09) (Oman, Host a visit, Iran, 2014-11-09) (Iran, Make a visit, Oman, 2014-11-09) (Catherine Ashton, Make a visit, Oman, 2014-11-10) (Mohammad Javad Zarif, Express intent to meet or negotiate, Oman, 2011-11-08) (Oman, Host a visit, Catherine Ashton, 2014-11-10)
15	(John Kerry, Express intent to meet or negotiate, Oman, 2014-11-09) (John Kerry, Express intent to meet or negotiate, Oman, 2014-11-10) (Oman, Host a visit, John Kerry, 2014-11-09) (Catherine Ashton, Make a visit, Oman, 2014-11-10) (Oman, Host a visit, Iran, 2014-11-09) (Mohammad Javad Zarif, Make a visit, Oman, 2011-11-09) (Catherine Ashton, Make a visit, Oman, 2014-11-09) (Catherine Ashton, Express intent to meet or negotiate, Oman, 2014-10-31) (Catherine Ashton, Express intent to meet or negotiate, Oman, 2014-11-03) (Oman, Host a visit, Mohammad Javad Zarif, 2014-11-09)

6. Conclusions

This study introduces a TKG subgraph reasoning method based on relaxed temporal relations. The method’s core involves extracting a query-related inference subgraph from a given TKG, comprising two main phases: subgraph expansion and subgraph pruning. In the expansion phase, a relaxed temporal factor is applied to loosen the order constraint on the occurrence times of past events. A time-dependent exponential weight strategy is then used to sample one-hop relaxed prior neighbor nodes, allowing the method to handle past events in a manner that better reflects real-world scenarios. During the pruning phase, an attention propagation mechanism computes attention scores for the nodes and edges in the subgraph, determining the pruning objects based on these scores. Through iterative rounds of subgraph expansion and pruning, the predicted entity is ultimately obtained. The introduction of the temporal relaxation factor enhances the model’s reasoning paths and increases its flexibility. Moreover, the computation of attention scores provides insight into the key influences of nodes and edges on the reasoning process, making it more intuitive and interpretable. Experimental results on four benchmark datasets demonstrate that the proposed SR-RTR method captures more relevant reasoning paths, thus improving the accuracy and efficiency of the model’s reasoning.

It is worth noting that the temporal relaxation strategy proposed in this paper still has certain limitations in its applicable scenarios: in scenarios such as political and social news where events are often clustered and strict chronological order within these clusters is less important than their collective occurrence, the effectiveness of this strategy has been verified by the experimental results in our study; however, in domains such as biological process modeling and financial transaction records where there are strict fine-grained causal dependencies and the order of events plays a decisive role, this method may be counterproductive. In future work, there remain several challenging research directions:

- (1) In this study, the temporal relaxation factor δ is set as a global hyperparameter, and its optimal value is determined through experimental tuning on different datasets. However, this setting implicitly assumes that the occurrence time of events follows a uniform distribution. In real-world scenarios, the event density across entities or time periods within a temporal knowledge graph is often not uniform. Therefore, designing an adaptive method to determine the optimal value of δ based on the local characteristics of nodes remains a challenging research direction.
- (2) Current research is still limited to qualitative analysis of the interpretability of model prediction results with the help of inference subgraphs. In the future, we may consider introducing standardized quantitative indicators such as faithfulness, combined with experimental designs like counterfactual reasoning, to systematically verify the causal correlation between inference subgraphs and prediction results, thereby enhancing the credibility of quantitative evaluation of interpretability and its reference value for practical applications.
- (3) This paper assumes that relation semantics are generally stable over time and models relations using the static embedding r_k ; nevertheless, this assumption has certain limitations when applied to domains with semantic drift. Therefore, integrating time-aware relation embedding representations and exploring the interaction between temporal structural relaxation and semantic-temporal dynamics will be a highly valuable extension direction.
- (4) Although we have verified the effectiveness of the SR-RTR model on four of the most widely used standard benchmark datasets, conducting comparative experiments on more large-scale datasets from other domains (e.g., GDELT or WIKIDATA) can further verify the model's generalization performance and clarify its applicable boundaries.

Author Contributions: Conceptualization, M.Y. and K.B.; methodology, M.Y. and T.H.; formal analysis, M.Y. and T.H.; investigation, M.Y. and F.W.; writing—original draft preparation, M.Y.; writing—review and editing, M.Y., K.B. and T.H.; visualization, M.Y. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology Research Project of Hubei Provincial Department of Education, grant number B2024544.

Data Availability Statement: The data presented in this study are available in the following public repositories: [23] at <https://yago-knowledge.org/data/yago3/> (accessed on 1 July 2025); [24] at <https://github.com/TemporalKGTeam/xERTE> (accessed on 1 July 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TKGs	Temporal Knowledge Graphs
SR-RTR	Subgraph Reasoning Model based on Relaxed Temporal Relations
RTRGA	Relaxed Temporal Relational Graph Attention

References

1. Saxena, A.; Tripathi, A.; Talukdar, P. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Online, 5–10 July 2020; pp. 4498–4507. [\[CrossRef\]](#)
2. Sun, Y.; Zhang, L.; Cheng, G.; Qu, Y. SPARQA: Skeleton-Based Semantic Parsing for Complex Questions over Knowledge Bases. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020), Online, 7–12 February 2020; pp. 5297–5304. [\[CrossRef\]](#)

3. Chang, C.; Tang, Y.; Long, Y.; Hu, K.; Li, Y.; Li, J.; Wang, C.D. Multi-Information Preprocessing Event Extraction with BiLSTM-CRF Attention for Academic Knowledge Graph Construction. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 2713–2724. [\[CrossRef\]](#)
4. Zhou, Q.; Zhang, Y.; Ji, D. Distantly supervised relation extraction with KB-enhanced reconstructed latent iterative graph networks. *Knowl.-Based Syst.* **2023**, *260*, 110108. [\[CrossRef\]](#)
5. Hu, Z.; Xia, F. Multi-stream graph attention network for recommendation with knowledge graph. *J. Web Semant.* **2024**, *82*, 100831. [\[CrossRef\]](#)
6. Zhang, Y.; Tian, J.; Sun, J.; Chan, H.; Qiu, A.; Liu, C. HKGAT: Heterogeneous knowledge graph attention network for explainable recommendation system. *Appl. Intell.* **2025**, *55*, 549. [\[CrossRef\]](#)
7. Jiang, T.; Liu, T.; Ge, T.; Sha, L.; Li, S.; Chang, B.; Sui, Z. Encoding Temporal Information for Time-Aware Link Prediction. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), Austin, TX, USA, 1–4 November 2016; pp. 2350–2354. [\[CrossRef\]](#)
8. Jiang, T.; Liu, T.; Ge, T.; Sha, L.; Chang, B.; Li, S.; Sui, Z. Towards time-aware knowledge graph completion. In Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016), Osaka, Japan, 11–16 December 2016; pp. 1715–1724.
9. Han, Z.; Chen, P.; Ma, Y.; Tresp, V. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In Proceedings of the International Conference on Learning Representations (ICLR 2021), Vienna, Austria, 4 May 2021. [\[CrossRef\]](#)
10. Liu, Y.; Ma, Y.; Hildebrandt, M.; Joblin, M.; Tresp, V. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022), Online, 22 February–1 March 2022; pp. 4120–4127. [\[CrossRef\]](#)
11. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS 2013), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
12. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014), Quebec City, QC, Canada, 27–31 July 2014; pp. 1112–1119. [\[CrossRef\]](#)
13. Dasgupta, S.S.; Ray, S.N.; Talukdar, P.P. HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), Brussels, Belgium, 31 October–4 November 2018; pp. 2001–2011. [\[CrossRef\]](#)
14. Trivedi, R.; Dai, H.; Wang, Y.; Song, L. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, NSW, Australia, 6–11 August 2017; pp. 5313–5327. [\[CrossRef\]](#)
15. Xu, C.; Nayyeri, M.; Alkhoury, F.; Yazdi, H.; Lehmann, J. Temporal knowledge graph completion based on time series gaussian embedding. In Proceedings of the 19th International Semantic Web Conference (ISWC 2020), Athens, Greece, 2–6 November 2020; pp. 654–671. [\[CrossRef\]](#)
16. Li, Y.; Sun, S.; Zhao, J. TiRGN: Time-Guided Recurrent Graph Network with Local-Global Historical Patterns for Temporal Knowledge Graph Reasoning. In Proceedings of the 31th International Joint Conference on Artificial Intelligence (IJCAI 2022), Vienna, Austria, 23–29 July 2022; pp. 2152–2158. [\[CrossRef\]](#)
17. Jin, W.; Qu, M.; Jin, X.; Ren, X. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), Online, 16–20 November 2020; pp. 6669–6683. [\[CrossRef\]](#)
18. Zhu, C.; Chen, M.; Fan, C.; Cheng, G.; Zhang, Y. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021), Online, 2–9 February 2021; pp. 4732–4740. [\[CrossRef\]](#)
19. Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; Cheng, X. Temporal knowledge graph reasoning based on evolutionary representation learning. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), Online, 11–15 July 2021; pp. 408–417. [\[CrossRef\]](#)
20. Liu, K.; Zhao, F.; Jin, H. FS-Net: Frequency Statistical Network for Temporal Knowledge Graph Reasoning. *J. Softw.* **2023**, *34*, 4518–4532. [\[CrossRef\]](#)
21. Sun, J.; Sheng, Y.; He, L.; Qin, Y.; Liu, M.; Jia, T. CEGRL-TKGR: A Causal Enhanced Graph Representation Learning Framework for Temporal Knowledge Graph Reasoning. *arXiv* **2024**, arXiv:2408.07911. [\[CrossRef\]](#)
22. Mu, C.; Zhang, L.; Ma, Y.; Tian, L. Temporal knowledge subgraph inference based on time-aware relation representation. *Appl. Intell.* **2023**, *53*, 24237–24252. [\[CrossRef\]](#)
23. Mahdisoltani, F.; Biega, J.; Suchanek, F.M. Yago3: A Knowledge Base from Multilingual Wikipedias. In Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR 2015), Asilomar, CA, USA, 4–7 January 2015.
24. Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; Ward, M. Icews Coded Event Data. 2015. Available online: <https://core.ac.uk/outputs/268481562/> (accessed on 1 July 2025).
25. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [\[CrossRef\]](#)

26. Yang, B.; Yih, W.T.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015. [[CrossRef](#)]
27. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the 33rd International Conference on Machine Learning (ICML 2016), New York, NY, USA, 19–24 June 2016; pp. 3021–3032. [[CrossRef](#)]
28. Leblay, J.; Chekol, M.W. Deriving Validity Time in Knowledge Graph. In Proceedings of the 27th International World Wide Web (WWW 2018), Lyon, France, 23–27 April 2018; pp. 1771–1776. [[CrossRef](#)]
29. García-Durán, A.; Dumančić, S.; Niepert, M. Learning sequence encoders for temporal knowledge graph completion. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), Brussels, Belgium, 31 October–4 November 2018. [[CrossRef](#)]
30. Goel, R.; Kazemi, S.M.; Brubaker, M.; Poupart, P. Diachronic embedding for temporal knowledge graph completion. In Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020), New York, NY, USA, 7–12 February 2020; pp. 3988–3995. [[CrossRef](#)]
31. Lacroix, T.; Obozinski, G.; Usunier, N. Tensor decompositions for temporal knowledge base completion. In Proceedings of the 8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia, 30 April 2020. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.