Utilizing Everything in History: Modeling Relation Inference Path and Entity Structure for Temporal Knowledge Graph Reasoning

Anonymous ACL submission

Abstract

Temporal Knowledge Graph (TKG) extrapolation fundamentally involves selecting the correct answer from all entities based on historical information. Current methods can easily eliminate most incorrect answers, narrowing the candidate pool to a tiny area called the candidate zone. However, these methods often fail to find the correct answer within this zone, primarily because the entities within the candidate zone are similar in subgraph structure or relational connectivity, causing significant interference. These methods, which either model the graph structure of entities or the paths of relationships, can only address one type of similarity. To address this issue, we propose a model called the **R**elation Causal Logic **I**nference and **E**ntity **S**tructure Learning (RIES), which consists of two modules: relation inference and entity structure. These two modules model the causal logic of relations over time and the temporal evolution of entities' subgraph structure, respectively, allowing for the differentiation of candidates similar in subgraph structure and relational connectivity. When evaluated on five commonly used public datasets, the performance of RIES surpasses that of other state-of-the-art baselines.

1 Introduction

Predicting future facts accurately requires a comprehensive analysis of historical data. Each timestamp links entities through a variety of relations, constructing a knowledge graph characterized by intricate structural and causal logic. Methods like CyGNet (Zhu et al. (2021)), CENET (Xu et al. (2023)), HGLS (Zhang et al. (2023)), and EvoExplore (Zhang et al. (2022)) typically model historical facts based on repetitive patterns, primarily making predictions from these recurrences. In contrast, some methods are entirely independent of entities, such as DaeMon (Dong et al. (2023)) and TiPNN (Dong et al. (2024)), which search for relation paths that have occurred in history and learn entity-agnostic inference rules. The main issues with these methods include:

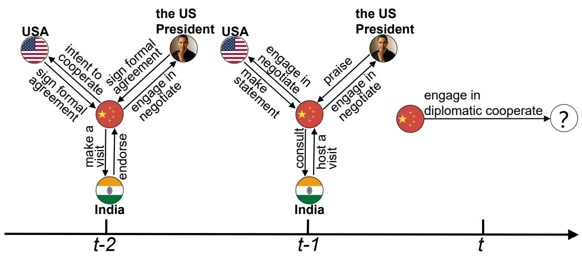

Figure 1: An illustration of temporal reasoning over a TKG.

Issue 1: The causal logic in the temporal order of relations between pairs of entities is not captured. Some graph-structured TKGR methods like CyGNet, CENET, HGLS, and EvoExplore do not focus on the changes in relations of the same entity pair across different timestamps, ignoring the causal logic of these relations over time. In the example of Figure 1, the variety of historical relations between the entities China and the US President do not contribute equally to answering queries. Focusing more on relations that are highly relevant to the query can reduce semantic noise during the reasoning process.

Issue 2: The aforementioned approaches consider only entities, or only relations, which have limitations in some specific cases. If we focus solely on relations, independent of the entities, it becomes difficult to distinguish between entities that share very similar historical relations with the query subject s. For instance, in Figure 1, the entities USA and the US President would be hard to differentiate. If we only consider the subgraph structure of the entities, such as USA and India, we find that the neighboring entities connected in the subgraphs for these two countries at different timestamps are all other country entities. The subgraph structures represented by these two entities are very similar, making it difficult to distinguish between them in the final prediction. To summarize, existing models focus on only one type of information in entities and relations and ignore the other, which limits their performance in TKGR.

To address the aforementioned issues, we model relations and entities information in a unified framework that allows these two types of information to be complementary in the reasoning process.

To solve issue 1, we propose a relation inference module, which consists of two parts: RCL (Relation Causal Logic) and PCA (Path Confidence Aggregation). (1) RCL: This part focuses on learning the temporal causal logic between historical relations and the query relation r_q. (2) PCA: This part involves aggregating the confidence scores of all relation inference paths between query subject s and candidate entities. It calculates the probability score that the query relation r_q will occur between the query subject s and the candidate entities at the timestamp t_q, based solely on relation data.

In order to tackle issue 2, we first propose an entity structure module, which models the structural dependencies between entities and concurrent facts. This enables us to generate a dynamic structural encoding of the query subject s and each candidate entity. We then decode this information to determine the probability of interaction between the query subject s and each candidate entity at the query timestamp t_q and under the query relation r_q. Subsequently, we combine the predictive probability scores from both the relation level and the entity level for each candidate entity to arrive at a final predictive probability score. By leveraging both relation and entity information, we can significantly improve the accuracy of our predictions.

In summary, our work makes the following contributions:

1) We have developed a relation inference module that explores the causal logic of relations in their temporal sequence by collecting information about the interactions between query entities and candidate entities from historical data.

2) To our knowledge, we are the pioneers in integrating modeling of relations and entities within a unified framework, effectively leveraging both relation and entity information.

3) Extensive experiments indicate that our model substantially outperforms existing methods.

2 Related Work

Depending on the type of historical information that a model focuses on, existing models can be divided into two categories: models based on historical entity information and models based on historical relation information.

Models based on historical entity information focus on modeling information about the entity (Park et al. (2022);Yang et al. (2023);Wu et al. (2020);Jin et al. (2020);Xiao et al. (2024);Zhang et al. (2023)). For instance, CyGNet (Zhu et al. (2021)) counts the frequency of entities occurring repeatedly in history and uses a copy mechanism to select prediction results from the entities that appear frequently. CENET (Xu et al. (2023)) adopts a comparative learning approach to capture the dependency of queries on both historical and non-historical entities. EvoExplore (Zhang et al. (2022)) implements a hierarchical attention mechanism to model the intricate local and global structures of entities.

Models based on historical relation information are completely independent of entities and focus on modeling the temporal path of relations (Sun et al. (2021);Lin et al. (2023)). For instance, CluSTeR (Li et al. (2021)) utilizes reinforcement learning to develop cluster search strategies that identify explicit and reliable relation clues for predicting future facts. DaeMon (Dong et al. (2023)) introduces a novel architecture that leverages timeline relations to adaptively capture temporal path information between query topics and candidate objects. ALRE-IR (Mei et al. (2022)) extracts relation paths from historical subgraphs, aligns these paths with current events to formulate rules, and then uses these rules to predict missing entities.

3 Method

3.1 Preliminaries

Let ε,R,T denote the finite set of entities, relations, and timestamps, respectively. In the temporal knowledge graph, each fact is represented by a quaternion (s,r,o,t), where $s \in \varepsilon$ is the subject entity, $o \in \varepsilon$ is the object entity, and $r \in R$ is the relation between s and o that occurs at timestamp $t \in T$. Specifically, given a query $q = (s, r_q, ?, t_q)$, we take the candidate object $o_i \in \varepsilon_c$ as an example, where the subscript c of ε_c is the initial letter of candidate, and ε_c is denoted as the set of all entities connected in the history of the query subject s,which we take as the set of candidate entities.

3.2 Model Overview

For predicting queries, we can consider two levels: On the one hand, from the relation, for a specific relation r_j between a subject s and a candidate object o_i under the historical timestamp t_τ denoted as $r_j^{t_\tau}$, a relation inference path $path\left(r_j^{t_\tau}\right) = (r_j, t_\tau) \rightarrow (r_q, t_q)$ is formed between it and the relation r_q under the query timestamp t_q. This relation inference path suggests that any pair of entities that have a relation r_j under timestamp t_τ, that pair will have a relation r_q under timestamp t_q. We explore the potential causal logic between (r_j, t_τ) and (r_q, t_q) to assess the confidence level that the relation inference path $path(r_j^{t_\tau})$ holds, and use it as a basis for reasoning that the query $q = (s, r_q, o_i, t_q)$ holds. After obtaining confidence scores for all relation inference paths between the subject s and the candidate object o_i, we aggregate these scores to finally obtain the likelihood score for reasoning that the query $q = (s, r_q, o_i, t_q)$ holds from the relation level.

On the other hand, focusing on entities, we examine the changes in the connectivity of the candidate object o_i with neighboring entities across various historical timestamps. We achieve the dynamic structural encoding of o_i by capturing the structural changes in the subgraphs where o_i is situated, which reflects the evolution of o_i's structural semantics over time. Similarly, we can obtain the dynamic structural encoding for the subject s. Subsequently, we decode the dynamic structural encodings of s and o_i using the ConvTransE (Shang et al. (2019)) decoder to determine the probability of interaction between s and o_i at the given query timestamp t_q and query relation r_q.

Ultimately, by integrating the scores from both the relation level and the entity structure level, we utilize this composite score as the final probability score for predicting the validity of the query $q = (s, r_q, o_i, t_q)$. The overall flow of our proposed model is shown in Figure 2. In the following, we elaborate on each part of the model.

3.3 Relation Inference

We denote the set of relations connected to the subject s of a query q at timestamp t_τ as $R_{s\rightarrow\varepsilon}^{t_\tau} \in \mathbb{R}^{|\varepsilon_c|\times|R|\times d}$, where $|\varepsilon_c|$ is the base of the set of candidate objects, $|R|$ is the base of the set of relations, and d is the dimension of the relation embedding. Specifically, given a query $q = (s, r_q, ?, t_q)$, we consider all the connected relations between the subject s and the candidate entity o_i. Since our goal is to capture the causal logic of the relations between s and o_i entity pairs across time, we need to obtain all relations information $R_{s\rightarrow o_i}^{t_\tau} \in R_{s\rightarrow\varepsilon}^{t_\tau}$ within the historical timestamp range of $[t_{q-len}, t_{q-1}]$, $\tau = q-len, \ldots, q-1$, where the parameter len is the length of the timestamp range of the historical information under consideration. Specifically for a single relation $r_j^{t_\tau} \in R_{s\rightarrow o_i}^{t_\tau} (j = 1, \ldots, |R_{s\rightarrow o_i}^{t_\tau}|)$ at timestamp t_τ, the confidence score of the relation inference path $path(r_j^{t_\tau})$ corresponding to relation $r_j^{t_\tau}$ is computed as follows:

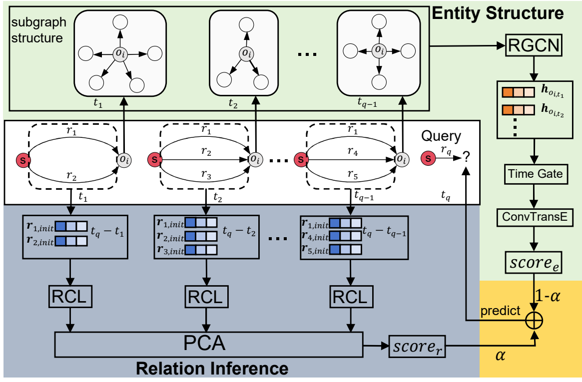

Figure 2: Architecture of RIES Framework. The gray shaded area in the bottom left explores the causal logic over time in the connecting relations between the subject entity s and the candidate entity o_i; the green shaded area in the upper right models each temporal subgraph of o_i to capture its dynamic structural semantics.

$$con(path(r_j^{t_\tau})) = RCL(r_j, r_q, (t_\tau, t_q)) \quad (1)$$

Where $RCL(\cdot)$ is a relation causal logic module, which aims to mine the potential causal logic between the query relation r_q and relation r_j in terms of temporal order.

We then aggregate the confidence scores of these relation inference paths to obtain the total confidence score of all relation inference paths between entity pairs s and o_i under timestamp t_τ:

$$con(path(R_{s\rightarrow o_i}^{t_\tau})) = \sum_{j=1}^{|R_{s\rightarrow o_i}^{t_\tau}|} con(path(r_j^{t_\tau})) \quad (2)$$

Upon calculating the total confidence scores for the relation inference paths between entities s and o_i across the time horizon $[t_{q-len}, t_{q-1}]$, we utilize path confidence aggregation (PCA) to aggregate these total confidence scores. This aggregation provides the historical relation inference path scores for $s \rightarrow o_i$:

$$score_r = PCA(con(path(R_{s\rightarrow o_i}^{[t_{q-len}, t_{q-1}]}))) \quad (3)$$

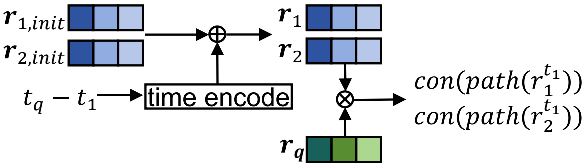

Figure 3: The architecture of RCL module. Exploring the causal logic of the relations r_1 and r_2 at timestamp t_1 on the relation r_q at timestamp t_q in a temporal order.

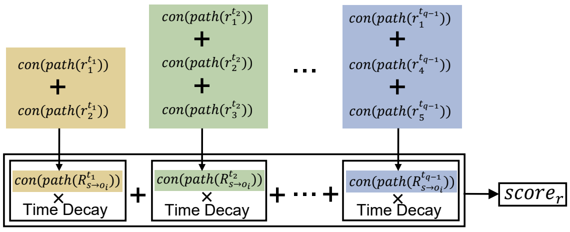

Figure 4: The architecture of PCA module. Aggregating the confidence scores of all relation inference paths between query subject s and candidate entity o_i.

In the following section, we provide a detailed description of the RCL module and the PCA module, respectively.

3.3.1 Relation Causal Logic

The workflow of relation causal logic (RCL) is shown in Figure 3. We first encode the temporal information as follows: At a specific historical timestamp t_τ, the relation $r_j^{t_\tau}$ occurring between the entity pairs s and o_i may lead to a query relation r_q occurring at timestamp $t_\tau + \Delta t$. Therefore, we encode the time interval Δt between the query time t_q and the historical time t_τ. For a relation $r_j^{t_\tau} \in R_{s \to o_i}^{t_\tau} (j = 1, \ldots, |R_{s \to o_i}^{t_\tau}|)$ at timestamp t_τ, where the time interval from the query q is $\Delta t = t_q - t_\tau$, the time interval is encoded as a d-dimensional time-encoded vector using the following equation:

$$T_{(\Delta t, 2\tau)} = sin(\Delta t / 10000^{2\tau/d}) \tag{4}$$

$$T_{(\Delta t, 2\tau+1)} = cos(\Delta t / 10000^{2\tau/d}) \tag{5}$$

After encoding the timing information, we add the time encoding to the initialized relation encoding $\mathbf{r}_{j,init}$ so that we obtain an embedding of the relation $r_j^{t_\tau}$:

$$\mathbf{r}_j = \mathbf{r}_{j,init} + T_{\Delta t} \tag{6}$$

Next, we obtain the relation inference path $path\left(r_j^{t_\tau}\right) = (r_j, t_\tau) \to (r_q, t_q)$ from the relation $r_j^{t_\tau}$ between the entity pairs s and o_i to the relation r_q at the query time t_q. We consider $r_j^{t_\tau}$ as the cause and r_q at t_q as the effect. Finally, we assess the confidence that the relation inference path $path(r_j^{t_\tau})$ holds by capturing the association between $r_j^{t_\tau}$ and r_q at the query time t_q. To compute this, we directly use the dot product method:

$$con(path(r_j^{t_\tau})) = \mathbf{r}_j * \mathbf{r}_q \tag{7}$$

Where $\mathbf{r}_j$ is the relation $r_j^{t_\tau}$ embedding that contains the time encoding and $\mathbf{r}_q$ is the initial relation embedding of the query q that does not contain the time encoding.

3.3.2 Path Confidence Aggregation

The workflow of path confidence aggregation (PCA) is shown in Figure 4. Calculation by means of Equation 2, we obtain the total confidence level score $con(path(R_{s \to o_i}^{t_{q-1}})), \ldots, con(path(R_{s \to o_i}^{t_{q-len}}))$ for the relation inference path for $s \to o_i$ at each timestamp within the time range $[t_{q-len}, t_{q-1}]$. In special cases, when two inference paths, $path(r_j^{t_{q-len}})$ and $path(r_j^{t_{q-1}})$, under different historical timestamps have the same relation r_j, we should assign different weights to these paths to distinguish between them. Due to the stability and simplicity of power functions, we define a power function-based time decay coefficient:

$$W_d(t_q, t_\tau) = (t_q - t_\tau)^{-\gamma} \tag{8}$$

The larger the value of γ in the above equation, the faster the rate at which W_d decays over time. The time decay coefficient W_d ensures that relation inference paths closer in time to the query time t_q are assigned higher weights. We weight the relation inference path confidence scores at each timestamp as follows:

$$PCA(con(path(R_{s \to o_i}^{[t_{q-len}, t_{q-1}]}))) = \sum_{\tau = q-len}^{q-1} W_d(t_q, t_\tau) con(path(R_{s \to o_i}^{t_\tau})) \tag{9}$$

3.4 Entity Structure

This module explores the association between the subject s of a query q and a candidate object o_i in terms of dynamic structural semantics, determining the probability that the subject s of the query interacts with candidate object o_i under the query timestamp t_q and the query relation r_q. The entire process is divided into two parts: encoding and decoding.

3.4.1 Entity Dynamic Structural Encoding

For simultaneous facts, entities usually have strong semantic correlations with their neighboring entities. To capture these semantics, we model them using the ω-layer R-GCN (Schlichtkrull et al. (2018)) as a structural encoder:

$$\mathbf{h}_{s,t}^{l} = f\Big(\frac{1}{|N_{s,t}|}\sum_{e_o^t \in N_{s,t}} W_1^l(\mathbf{h}_{o,t}^{l} + \mathbf{r}) + W_2^l \mathbf{h}_{s,t}^{l-1}\Big) \quad (10)$$

Where $N_{s,t}$ is the set of neighbors of entity s in the static subgraph at timestamp t, $f(\cdot)$ is the reflection modified linear unit (RReLU (Xu et al. (2015))) activation function, $W_1^l \in \mathbb{R}^{d\times d}$ is a relation-specific parameter used for aggregating structural features based on different edges, $W_2^l \in \mathbb{R}^{d\times d}$ denotes the parameter that aggregates the self-loop features of all entities, $\mathbf{h}_{o,t}^{l}$ and $\mathbf{r}$ denote the embedding of the neighboring entity e_o^t in the l-th layer of the R-GCN and the embedding of the connected relation, respectively. After ω layers of R-GCN, we can obtain a representation $\mathbf{h}_{s,t}^{\omega}$ that only considers semantic dependencies with neighboring nodes of entity s at timestamp t.

To capture the dynamic structural semantic changes of an entity s over a short period, the model needs to consider all temporally neighboring facts. Therefore, we use the structural semantic output of the entity from the previous timestamped subgraph as input to the R-GCN model for the next timestamp:

$$\mathbf{h}_{s,t+1}^{1} = \mathbf{h}_{s,t}^{\omega} \quad (11)$$

We use the time-gate loop component to further model the temporal dependence of the entity structure. The dynamic structural semantic embedding $\mathbf{e}_{s,t+1}$ of the final entity s is determined by two components: the output of the last layer of the R-GCN, $\mathbf{h}_{s,t+1}^{\omega}$, and the $\mathbf{e}_{s,t}$ from the previous timestamp. The specific expressions are as follows:

$$\mathbf{e}_{s,t+1} = U_{t+1} \otimes \mathbf{h}_{s,t+1}^{\omega} + (1 - U_{t+1}) \otimes \mathbf{e}_{s,t} \quad (12)$$

The expression $\otimes$ denotes the dot product operation. The time gate $U_{t+1} \in \mathbb{R}^{d\times d}$ undergoes a nonlinear transformation as:

$$U_{t+1} = \sigma(W_u \mathbf{e}_{s,t} + b) \quad (13)$$

Where $\sigma(\cdot)$ is the sigmoid function and $W_u \in \mathbb{R}^{d\times d}$ is the weight matrix of the time gate.

3.4.2 Entity Dynamic Structure Decoding

We choose ConvTransE (Shang et al. (2019)) as the decoder to compute the degree of association between the subject s of the query q and the candidate object o_i at the dynamic structural-semantic level under the query timestamp t_q, represented as follows:

$$score_e = \sigma(\mathbf{e}_{o_i,t_q} ConvTransE(\mathbf{e}_{s,t_q}, \mathbf{r}_q)) \quad (14)$$

Where $\mathbf{r}_q$ is the initial relation embedding of query q. This function yields the probability that the subject s interacts with a candidate object o_i at time t_q and relation r_q. In other words, it represents the probability that the query $q = (s, r_q, o_i, t_q)$ holds from the perspective of the entity structure.

3.5 Inference

To ensure that we can maximize the use of relation and entity information, we introduce the coefficient α to adjust the weight between the relation inference score and the entity structure score. The final prediction that the missing object entity in $q = (s, r_q, ?, t_q)$ will be the highest combined probability entity $\hat{o}$ for both aspects:

$$P(o|s, r_q, t_q) = \alpha * score_r + (1-\alpha) * score_e \quad (15)$$

$$\hat{o} = argmax_{o\in\varepsilon_c} P(o|s, r_q, t_q) \quad (16)$$

Where $P(o|s, r_q, t_q)$ denotes the predicted probability of all candidate object entities $o \in \varepsilon_c$.

3.6 Train

In the relation inference process, we compute the similarity between the embedding $\mathbf{r}_j$ of $r_j^{t_\tau}$ and the relation embedding $\mathbf{r}_q$ of the query q in the embedding space by using the dot product to obtain the confidence score for the relation inference path $path\left(r_j^{t_\tau}\right) = (r_j, t_\tau) \rightarrow (r_q, t_q)$. The challenge lies in determining the correct inference path and assigning it a higher confidence score. To address this, we design a positive and negative sample comparison training method. This method learns the $r_j^{t_\tau}$ relation embedding $\mathbf{r}_j$ in the relation inference path $path\left(r_j^{t_\tau}\right)$ so that when the relation inference path is correct, the historical relation embedding $\mathbf{r}_j$ is spatially close to the relation embedding $\mathbf{r}_q$ of the query q. Conversely, when the relation inference path is incorrect, $\mathbf{r}_j$ is spatially distant from $\mathbf{r}_q$.

First, we negatively sample and generate the error quaternion. Specifically, given a correct quaternion $pos = (s, r, o, t)$, we randomly

Datasets	Entities	Relations	Training	Validation	Test	Time Granules
ICEWS14	7128	230	63685	13823	13222	365
ICEWS0515	10488	251	322958	69224	69147	4017
ICEWS18	23033	256	373018	45995	49545	304
WIKI	12554	24	539286	67538	63110	232
YAGO	10623	10	161540	19523	20026	189

Table 1: Statistical data for the datasets.

Model	ICEWS14				ICEWS18				ICEWS0515			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
ComplEX	30.84	21.51	34.48	49.58	21.01	11.87	23.47	39.87	31.69	21.44	35.74	52.04
R-GCN	28.03	19.42	31.95	44.83	15.05	8.13	16.49	29.00	27.13	18.83	30.41	43.16
DE-SimplE	32.67	24.43	35.69	49.11	19.30	11.53	21.86	34.80	35.02	25.91	38.99	52.75
CyGNet	32.73	23.69	36.31	50.67	24.93	15.90	28.28	42.61	34.97	25.67	39.09	52.94
xERTE	40.79	32.70	45.67	57.30	29.31	21.03	33.51	46.48	46.62	37.84	52.31	63.92
CEN	42.40	32.08	47.46	61.31	31.05	21.70	35.44	50.59	-	-	-	-
TECHS	43.88	34.59	49.36	61.95	30.85	21.81	35.39	49.82	48.38	38.34	54.69	68.92
DaeMon	-	-	-	-	31.85	22.67	35.92	49.80	-	-	-	-
HGLS	47.00	35.06	-	70.41	29.32	19.21	-	49.83	46.21	35.32	-	67.12
RPC	44.55	34.87	49.80	65.08	34.91	24.34	38.74	55.89	51.14	39.47	57.11	71.75
TiPNN	-	-	-	-	32.17	22.74	36.24	50.72	-	-	-	-
DLGR	46.72	36.67	51.61	-	35.48	25.11	40.03	-	-	-	-	-
RIES	**54.34**	**41.88**	**61.49**	**77.84**	**39.12**	**26.28**	**45.02**	**64.69**	**56.52**	**44.50**	**63.47**	**79.03**
Absolute Boost	7.34	5.21	9.88	7.43	3.64	1.17	4.99	8.80	5.38	5.03	6.36	7.28
Relative Boost	15.62	14.21	19.14	10.55	10.26	4.66	12.47	15.75	10.52	12.74	11.14	10.15

Table 2: Performance (in percentage) on ICEWS14, ICEWS18, ICEWS0515. Best results are bolded, sub-optimal results are underlined.

sample an object entity from historical events and disrupt the quaternion to generate an incorrect quaternion neg that satisfies the condition $neg = \{(s,r,o',t)|o' \in \varepsilon - o\}$. We ensure that the correct quaternions (positive samples) receive higher scores and the incorrect quaternions (negative samples) receive lower scores by using the $SoftMarginLoss$ function, expressed as follows:

$$L = \sum_{(s,r,o,t) \in P \bigcup N} log(1 + exp(-y \cdot score_r(s,r,o,t))) \tag{17}$$

$$y = \begin{cases} 1, & (s,r,o,t) \in P \\ -1, & (s,r,o,t) \in N \end{cases} \tag{18}$$

In Equation 18, P is the set of correct quaternions and N is the set of incorrect quaternions.

The training task based on the $SoftMarginLoss$ function is to assign higher scores to correct quaternions and lower scores to incorrect quaternions, with these scores derived from the confidence level of the relation inference paths. From the perspective of the embedding space, this task brings the historical relation embeddings of the positive examples closer to the query relation embedding, while moving the historical relation embeddings of the negative examples further away from the query relation embedding.

In short, this training task is to enable correct relation inference paths to achieve higher confidence scores.

4 Experiment

4.1 Experimental Setup

4.1.1 Datasets

We use five benchmark datasets (ICEWS14 (Li et al. (2022b)), ICEWS0515 (Ren et al. (2023)), ICEWS18 (Boschee et al. (2015)), WIKI (Vrandečić and Krötzsch (2014)), and YAGO (Suchanek et al. (2007))) to evaluate the performance of the model on the temporal knowledge graph reasoning task. To ensure a fair comparison, we follow the data partition provided in the reference TECHS (Lin et al. (2023)) to divide each dataset into training, validation, and test sets. Table 1 provides statistics for these data sets.

To assess the validity of our proposed model, we have thoroughly compared the experimental results with various static and temporal models.

4.1.2 Assessment Indicators and Training Settings

In our experiments, we used MRR and Hits@1,3,10 as evaluation indicators. For the configuration of the model, we use random initialization to generate relation embeddings of dimension 200. To optimize all model parameters, we used the Adam (Kingma (2014)) optimizer and set the initialized learning rate to 0.001. For the entity structure module, we set the number of layers ω of R-GCN to 2. For each R-GCN layer, the dropout rate is set to 0.2 and the history length is set to 10. For Con-

Model	WIKI				YAGO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
ComplEX	24.47	19.69	27.28	34.83	44.38	25.78	48.2	59.01
R-GCN	13.96	-	15.75	22.05	20.25	-	24.01	37.30
DE-SimplE	45.43	42.60	47.71	-	54.91	51.64	57.30	-
CyGNet	58.78	47.89	66.44	78.70	68.98	58.97	76.80	86.98
xERTE	73.60	69.05	78.03	79.73	84.19	80.09	88.02	89.78
CEN	78.93	75.05	81.90	84.90	-	-	-	-
TECHS	75.98	-	-	82.39	89.24	-	-	92.39
DaeMon	82.38	78.26	86.03	88.01	91.59	90.03	93.00	93.34
HGLS	82.04	78.07	84.04	-	87.48	83.17	89.76	-
RPC	81.18	76.28	85.43	88.71	88.87	85.10	92.57	94.04
TiPNN	83.04	79.04	86.45	88.54	92.06	90.79	93.15	93.58
DLGR	82.98	80.14	80.14	-	88.87	84.60	92.35	-
RIES	**89.46**	**87.34**	**91.82**	**93.12**	**94.73**	**92.83**	**95.25**	**96.63**
Absolute Boost	6.42	7.20	5.37	4.41	2.67	2.04	2.10	2.59
Relative Boost	7.73	8.98	6.21	4.97	2.90	2.25	2.25	2.75

Table 3: Performance (in percentage) on WIKI, YAGO. Best results are bolded, sub-optimal results are underlined.

vTransE, the kernel size is set to 2×3 and the dropout rate is set to 0.2. Specifically, we trained the model for 100 epochs, with early stopping if the validation loss did not decrease for 10 consecutive epochs. All experiments were conducted on a single Tesla T4 GPU with 16GB of memory. The model has approximately 9 million parameters. The time required to run one epoch on the ICEWS14, ICEWS18, ICEWS0515, YAGO, and WIKI datasets is approximately 10, 60, 110, 10, and 20 minutes, respectively.

4.2 Experimental Results

The experimental results of RIES and all the baselines on TKG reasoning are presented in Tables 2 and 3. The results are from the average of the experiments. We chose ComplEX (Trouillon et al. (2016)) and R-GCN (Schlichtkrull et al. (2018)) as static models for comparison. DE-SimplE (Goel et al. (2020)), CyGNet (Zhu et al. (2021)), xERTE (Han et al. (2020)), CEN (Li et al. (2022a)), TECHS (Lin et al. (2023)), DaeMon (Dong et al. (2023)), HGLS (Zhang et al. (2023)), RPC (Liang et al. (2023)), TiPNN (Dong et al. (2024)), and DLGR (Xiao et al. (2024)) as comparative temporal models.

Static models such as ComplEX and R-GCN underperform compared to temporal models because they fail to consider temporal information and dependencies across different snapshots. Similarly, the interpolation model DE-SimplE also performs poorly because such models struggle to handle events occurring in future timestamps. Among the extrapolation models, CyGNet, xERTE, CEN, HGLS, and DLGR focus on entity information and overlook the dynamic changes in relations between entity pairs over time. TECHS, DaeMon, RPC, and TiPNN start from relations, utilizing path-based searches to extract potential logical rules within the graph. These methods are limited by the existing paths, which restrict their search range and impair their performance. Our proposed model operates within a unified framework that models relations and entities, exploring the causal logic between relations over time and the dynamic structural changes of entities. By fully leveraging information on relations and entities for prediction, our model outperforms the state-of-the-art across all metrics on five datasets.

	ICEWS14	ICEWS18	YAGO
RIES	54.34	39.12	94.73
RIES w/o R	46.16(-8.2)	35.26(-3.9)	89.32(-5.4)
RIES w/o E	50.81(-3.5)	36.05(-3.1)	82.65(-12.1)
RIES w/o (E&R-TE)	43.79(-10.6)	31.55(-7.6)	79.53(-15.2)
RIES w/o (E&R-TD)	46.39(-8.0)	33.83(-5.3)	81.14(-13.6)

Table 4: Results (in percentage) by different variants of our model on three datasets.

4.3 Ablation Study

To test the contribution of each component in the model, we performed ablation experiments.

To further analyze the contribution that each part of the model makes to the final prediction results, we report in Table 4 above the results of the MRR metrics for the five sub-models on the test sets of the three datasets. The five sub-models compared are: 1. RIES, the full model. 2. RIES w/o R, representing RIES without the relation inference module. 3. RIES w/o E, representing RIES without the entity structure module. 4. RIES w/o (E&R-TE), representing RIES without the entity structure module and without using time encoding in the relation inference module. 5. RIES w/o (E&R-TD), representing RIES without the entity structure module and without using the time decay coefficient in the relation inference module.

query	relation	score-r		score-e	Target entity
(China, engage in diplomatic cooperate, ?, t)	engage in negotiate,t-1 make statement,t-1 intent to cooperate,t-2 sign formal agreement,t-2	0.575 0.516 0.351 0.332	⇒**1.774**	0.703	USA(√)
	engage in negotiate,t-1 praise,t-1 engage in negotiate,t-2 sign formal agreement,t-2	0.575 0.604 0.287 0.332	⇒**1.798**	0.372	the US President
	host a visit,t-1 consult,t-1 make a visit,t-2 endorse,t-2	0.316 0.287 0.158 0.208	⇒**0.969**	0.768	India

Table 5: A case demonstrating that entity and relation information can effectively complement each other in the reasoning process.

From the results in Table 4, we draw the following findings:

Effectiveness of combined use of relation and entity information. The full model RIES outperforms RIES w/o R and RIES w/o E on all datasets, which confirms that relation and entity information complement each other well for future prediction.

Validity of time encoding in relation inference modules. The experimental results of RIES w/o (E&R-TE) have a substantial decrease compared to RIES w/o E. This is because RIES w/o (E&R-TE) does not consider the dynamic change of causal logic between relations, and ignores the absolute temporal numerical information. What is learned in this case is a static relation inference path independent of temporal order, which is unsuitable for reasoning on temporal knowledge graphs.

Validity of time decay coefficient in relation inference modules. The experimental results for RIES w/o (E&R-TD) have also decreased compared to RIES w/o E. This confirms the necessity of considering the relative temporal distance of the inference paths from the query. The value of historical relation information decreases progressively as this relative temporal distance increases.

4.4 Case Study

Considering the limited length of the paper, it is necessary to limit the number of relations between the subject entity and candidate entities. Therefore, we set the parameter len of the history time horizon to 2. For the query in the ICEWS14 test set (China, engage in diplomatic cooperate, ?, t), we selected the top three scoring entities among the candidates and presented them in Table 5.

From the perspective of relation inference alone, relations such as engage in negotiate, make statement, and praise provide high scores for the candidate entities the US President and USA. The scores for USA (1.774) and the US President (1.798) are very similar, but the incorrect answer, the US President, scores higher than the correct, USA.

From the perspective of entity structure alone, the subgraph structures of the candidate entities USA and India are quite similar, with neighboring nodes mostly being other national entities. However, the incorrect answer, India (0.768), scores higher than the correct, USA (0.703). This is primarily because India has a closer relationship with China compared to USA, as both are Asian countries and their connected neighboring entities are predominantly from Asia.

The correct answer, USA, can only be determined by combining scores from both relation inference and entity structure. This shows that considering only relation or entity information alone is not enough to distinguish similar candidate entities. Optimal reasoning results can only be achieved by effectively utilizing both types of information.

5 Conclusion

In this paper, we consider two types of information in graphs: entity information and relation information. For the first time, we model these two types of information within a unified framework. We further propose the RIES model, divided into two components: relation inference and entity structure, to handle relation and entity information. At the relation level, the relation inference component explores the causal logic of different relations over time and constructs reasonable inference paths. At the entity level, the entity structure component encodes the dynamic structure of entities and discovers their associations within subgraph structures. Experiments on five benchmark datasets demonstrate the effectiveness of our model in temporal knowledge graph extrapolation tasks.

Limitations

The timestamp range for historical information modeled by RIES is determined by the parameter *len*. Currently, selecting the *len* value requires manual intervention, with different datasets needing to be manually set to different values. This makes it challenging to determine the optimal parameter value. Future work could explore the automatic optimization of this parameter to further enhance the model's predictive capability.

References

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12:2.

Hao Dong, Zhiyuan Ning, Pengyang Wang, Ziyue Qiao, Pengfei Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Adaptive path-memory network for temporal knowledge graph reasoning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 2086–2094.

Hao Dong, Pengyang Wang, Meng Xiao, Zhiyuan Ning, Pengfei Wang, and Yuanchun Zhou. 2024. Temporal inductive path neural network for temporal knowledge graph reasoning. *Artificial Intelligence*, 329:104085.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3988–3995.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683.

DP Kingma. 2014. Adam: a method for stochastic optimization. In *Int Conf Learn Represent*.

Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022a. Complex evolutional pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 290–296.

Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022b. Hismatch: Historical structure matching based temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7328–7338.

Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2021. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4732–4743.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. 2023. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1559–1568.

Qika Lin, Jun Liu, Rui Mao, Fangzhi Xu, and Erik Cambria. 2023. Techs: Temporal logical graph networks for explainable extrapolation reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1281–1293.

Xin Mei, Libin Yang, Xiaoyan Cai, and Zuowei Jiang. 2022. An adaptive logical rule embedding model for inductive reasoning over temporal knowledge graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7304–7316.

Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. 2022. Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 794–803.

Xin Ren, Luyi Bai, Qianwen Xiao, and Xiangxi Meng. 2023. Hierarchical self-attention embedding for temporal knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pages 2539–2547.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI conference* on artificial intelligence*, volume 33, pages 3060–3067.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. 2020. Temp: Temporal message passing for temporal knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5730–5746.

Yao Xiao, Guangyou Zhou, Zhiwen Xie, Jin Liu, and Jimmy Xiangji Huang. 2024. Learning dual disentangled representation with self-supervision for temporal knowledge graph reasoning. *Information Processing & Management*, 61(3):103618.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023. Temporal knowledge graph reasoning with historical contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4765–4773.

Fan Yang, Jie Bai, Linjing Li, and Daniel Zeng. 2023. A continual learning framework for event prediction with temporal knowledge graphs. In *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 01–06. IEEE.

Jiasheng Zhang, Shuang Liang, Yongpan Sheng, and Jie Shao. 2022. Temporal knowledge graph representation learning with local and global evolutions. *Knowledge-Based Systems*, 251:109234.

Mengqi Zhang, Yuwei Xia, Qiang Liu, Shu Wu, and Liang Wang. 2023. Learning long-and short-term representations for temporal knowledge graph reasoning. In *Proceedings of the ACM Web Conference 2023*, pages 2412–2422.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4732–4740.

A Parameters Analysis

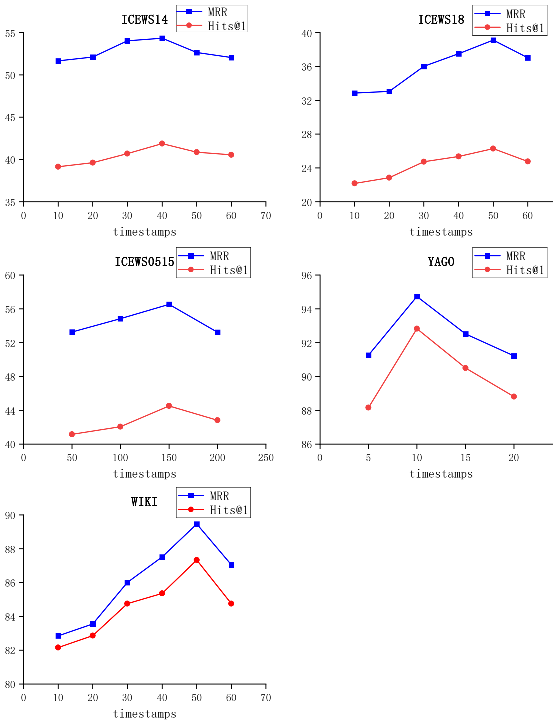

Figure 5: Result on five dataset with different *len*.

In the relational inference module, we acquired all relational information located within the historical timestamp range $[t_{q-len}, t_{q-1}]$, where the parameter *len* represents the length of this historical range. To determine the optimal value for *len*, we conducted a detailed parameter tuning experiment and tested the model's performance across different *len* values on the metrics MRR and Hits@1. The specific experimental results are shown in Figure 5.

The *len* values for the ICEWS14, ICEWS18, and WIKI datasets were set at 10, 20, 30, 40, 50, and 60. For the ICEWS0515 dataset, they were set at 50, 100, 150, and 200. On the YAGO dataset, they were set at 5, 10, 15, and 20. Across all five datasets, as the value of *len* increased, both metrics, MRR and Hits@1, initially improved and then declined. We analyzed the reasons as follows: When the value of *len* is too small, it considers too little historical information, failing to capture enough relational causal logic. Conversely, when *len* is too large, it introduces history that is too distant from the query time, which is of lower value and contains too much irrelevant information. Thus, both too-small and too-large values of len are detrimental to predicting future queries. Ultimately, the optimal values of len selected for the ICEWS14, ICEWS18, ICEWS0515, WIKI, and YAGO datasets were 40, 50, 150, 50, and 10, respectively.