TiRGN: Time-Guided Recurrent Graph Network with Local-Global Historical Patterns for Temporal Knowledge Graph Reasoning

Yujia Li , **Shiliang Sun** and **Jing Zhao***
School of Computer Science of Technology, East China Normal University, Shanghai 200062, China
jzhao@cs.ecnu.edu.cn

Abstract

Temporal knowledge graphs (TKGs) have been widely used in various fields that model the dynamics of facts along the timeline. In the extrapolation setting of TKG reasoning, since facts happening in the future are entirely unknowable, insight into history is the key to predicting future facts. However, it is still a great challenge for existing models as they hardly learn the characteristics of historical events adequately. From the perspective of historical development laws, comprehensively considering the sequential, repetitive, and cyclical patterns of historical facts is conducive to predicting future facts. To this end, we propose a novel representation learning model for TKG reasoning, namely TiRGN, a time-guided recurrent graph network with local-global historical patterns. Specifically, TiRGN uses a local recurrent graph encoder network to model the historical dependency of events at adjacent timestamps and uses the global history encoder network to collect repeated historical facts. After the trade-off between the two encoders, the final inference is performed by a decoder with periodicity. We use six benchmark datasets to evaluate the proposed method. The experimental results show that TiRGN outperforms the state-of-the-art TKG reasoning methods in most cases.

1 Introduction

Knowledge graphs (KGs) have been widely used in intelligent applications, such as question answering, recommendation systems, and information retrieval. However, the incompleteness of KGs limits their performance in downstream tasks. In this regard, there have been many studies focusing on the reasoning and completion of static KGs. In reality, many data often have complex dynamics, and it is a challenge to characterize the dynamics and reason over temporal knowledge graphs (TKGs).

Each fact in TKGs is represented by a quadruple containing a timestamp, which indicates that the fact occurred at a specific time. For example, `(Barack Obama, impose sanctions, Sudan, 2013-11-7)` indicates the fact that Obama imposed sanctions on Sudan in 2013-11-7. In this paper, we mainly solve the extrapolation problem of TKG reasoning, including entity prediction and relation prediction. This is significant for many practical applications, such as event process induction, social relation prediction, disaster relief, and financial analysis.

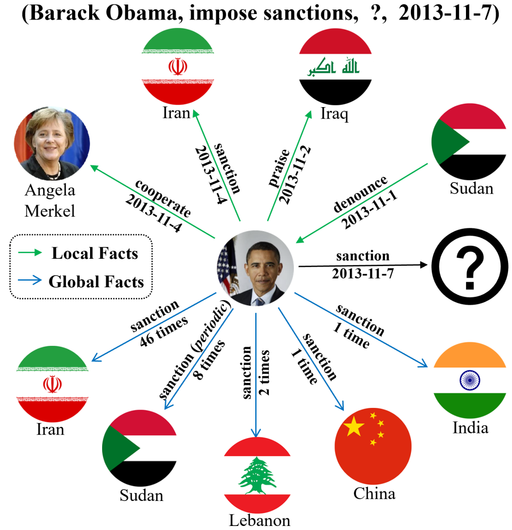

Figure 1: Example of different historical facts related to the query from ICEWS05-15. Different arrows indicate facts at adjacent timestamps and facts that are repeated and periodic at global timestamps.

Making accurate predictions about future facts needs to learn more about historical facts based on the laws of historical development. According to historic recurrence [Trompf, 1979] and social cycle theory [Schlesinger, 1999], historical facts may have a repetitive or even cyclical pattern. Besides, according to human cognition, the historical facts at adjacent timestamps have a sequential pattern, in which the evolution law of adjacent events can be captured to predict what will happen next. For example, as shown in Figure 1, we predict the query `(Barack Obama, impose sanctions, ?, 2013-11-7)`. We can get a series of historical adjacent facts and global facts related to the query. The countries in the relevant local facts include Iran which has recently been sanctioned, and Sudan which has recently

*Corresponding author.

criticized Obama. In the relevant global facts, Obama imposed sanctions on eight countries, including sanctions on Iran 46 times and Sudan eight times. If both local and global facts are considered, we can narrow the scope of the prediction results of the example query to Iran and Sudan. In addition, we find that Obama imposed sanctions on Sudan around November annually from 2009 to 2012. Considering periodicity and the other two historical patterns simultaneously, it is more likely that the predicted result will be Sudan, which is indeed the case. Therefore, capturing different historical patterns can constrain the range of the final prediction results and improve accuracy.

Recently, some methods have tried to extract relevant historical information for different queries in a heuristic way. However, they did not comprehensively consider the different historical characteristics. RE-Net [Jin *et al.*, 2020] and CyGNet [Zhu *et al.*, 2021] only consider the entities or entity-relation pairs in the global history. Among them, RE-Net can only capture global facts within a limited time. Otherwise, it will cause huge time complexity. CyGNet describes global facts by occurrence frequency, which leads to the narrow results that the prediction always tends to the facts with the highest frequency. RE-GCN [Li *et al.*, 2021b] attempts to model local historical dependency but lacks the capture of global historical information. Furthermore, although these methods extract historical information, none of them models the periodicity of historical facts.

To this end, we propose a model that captures multiple historical characteristics with local-global historical information, called Time-Guided Recurrent Graph Network (TiRGN). The main ideas of TiRGN are (1) to combine the local and global historical information to capture sequential, repetitive, and cyclical patterns of historical facts, (2) to regard the temporal subgraphs at adjacent timestamps as a sequence and regard the subgraphs at global timestamps as a constraint, and (3) to design a time-guided periodic decoder by using model-independent time vectors. Specifically, in order to capture the sequential pattern of historical facts, we design a graph neural network-based encoder with a double recurrent mechanism to simultaneously evolve the entity and relation representations at adjacent timestamps. In addition, we use the global one-hop or multi-hop repeated history information to capture repetitive patterns of historical facts. Furthermore, the time vectors involved in the model can capture the periodicity of facts. Finally, the scores calculated by the decoder are used to realize entity and relation predictions.

In general, this work presents the following contributions:

- We propose a representation learning model TiRGN for TKG reasoning, which simultaneously considers the sequential, repetitive and cyclical patterns of historical facts. As far as we know, this is the first time to integrate these historical characteristics for TKG reasoning.
- We design a double recurrent mechanism with long-distance dependencies to encode the adjacent subgraph sequences and use a low-complexity global history encoder to collect repeated facts. We use periodic time vectors to guide the decoder and realize the trade-off between local and global historical information.
- Experiments on six public TKG datasets demonstrate that TiRGN is consistently effective on both entity prediction and relation prediction.

2 Related Work

Recently, some studies have tried to incorporate temporal information into KG reasoning, which can be classified into two settings: interpolation and extrapolation [Jin *et al.*, 2020]. Interpolation is used to predict missing historical facts. For this setting, several embedding-based methods associate time with facts and map them to a low-dimensional space [García-Durán *et al.*, 2018; Ma *et al.*, 2019; Wu *et al.*, 2020]. TTransE [Jiang *et al.*, 2016] is a variant of TransE [Bordes *et al.*, 2013], which treats relation and time as translation between entities. DE-SimplE [Goel *et al.*, 2020] and ChronoR [Sadeghian *et al.*, 2021] characterize temporal information by learning the embeddings of different timestamps. However, these models cannot predict future facts.

This paper focuses on the extrapolation setting of predicting future facts according to history, which has been studied recently. Know-Evolve [Trivedi *et al.*, 2017] and DyREP [Trivedi *et al.*, 2019] use the temporal point process to model the occurrence of facts in TKGs. Glean [Deng *et al.*, 2020] incorporates relational and word contexts to enrich the features of facts for reasoning. RE-NET [Jin *et al.*, 2020] employs a neighborhood aggregator and recurrent event encoder to model the historical facts as subgraph sequences. TANGO [Han *et al.*, 2021b] explores the neural ordinary differential equation to build a continuous-time reasoning model. xERTE [Han *et al.*, 2021a] uses a subgraph sampling technique to construct interpretable reasoning graphs. CluSTeR [Li *et al.*, 2021a] and TITer [Sun *et al.*, 2021] both use reinforcement learning to search a series of historical facts for reasoning. CyGNet [Zhu *et al.*, 2021] and RE-GCN [Li *et al.*, 2021b] are the most relevant work to us. CyGNet uses a copy-generation mechanism to capture the global repetition frequency of facts. RE-GCN learns the evolutional representations of entities and relations at each timestamp through the structures that capture local historical dependency. However, none of the above methods simultaneously consider the sequential, repetitive, and cyclical historical facts.

3 The Proposed Method

3.1 Notations

In this paper, we formalize a TKG $\mathcal{G}$ as a sequence of subgraphs, i.e., $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$. The subgraph $\mathcal{G}_t = (\mathcal{E}, \mathcal{R}, \mathcal{F}_t)$ at t is a directed multi-relational graph, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{F}_t$ is the set of facts at t. A fact in $\mathcal{F}_t$ can be formalized as a quadruple (s, r, o, t), where $s, o \in \mathcal{E}$ and $r \in \mathcal{R}$. It represents that there is a relation r between the subject entity s and the object entity o at t. For each quadruple (s, r, o, t), an inverse relation quadruple (o, r^{-1}, s, t) is often added to the dataset. TKG reasoning can be classified into entity prediction $(s, r, ?, t)$ and relation prediction $(s, ?, o, t)$ given the set of historical facts before t. For each prediction at t, we formalize the subgraph sequence of its previous m timestamps as $\mathcal{G}_{t-m:t-1}$.

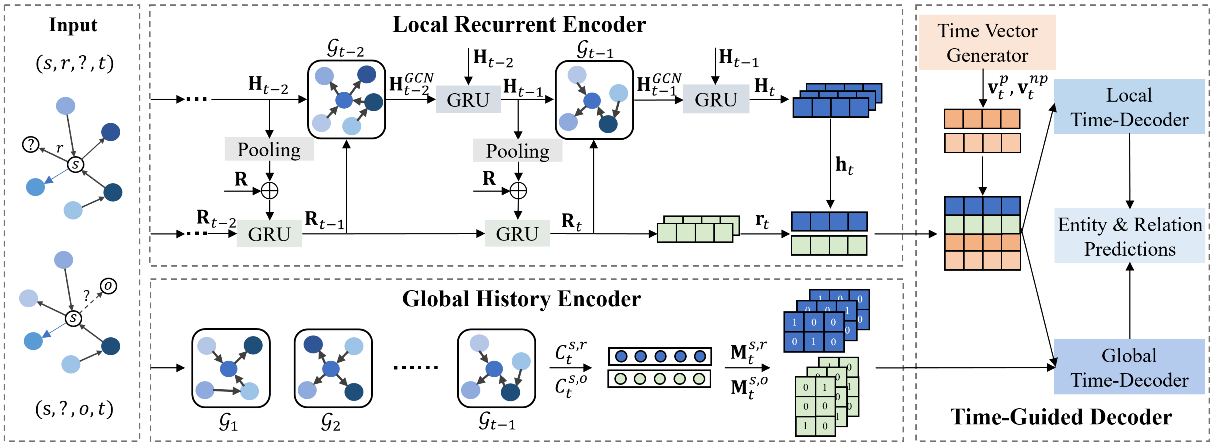

Figure 2: Illustration of the proposed TiRGN model. The local recurrent encoder encodes the evolutionary representations of entities and relations at adjacent timestamps. The global encoder collects all the repeated facts before the timestamp of the query. Periodic and non-periodic time vectors, together with the representations of entities and relations, are sent to the local and global decoders to obtain the final scores for prediction.

For all timestamps before t, the set of all candidate object entities for each entity prediction $(s, r, ?, t)$ is denoted as $\mathcal{C}_t^{s,r}$, and the set of all candidate relations for each relation prediction $(s, ?, o, t)$ is denoted as $\mathcal{C}_t^{s,o}$.

3.2 Model Overview

The overall framework of TiRGN is shown in Figure 2. According to the laws of historical development, TiRGN consists of three components, which are used to capture the sequential, repetitive, and cyclical patterns of historical facts, respectively. Local recurrent encoder is used to explore structural features and historical dependency. If there are facts in the subgraphs of adjacent timestamps containing the same semantic information as the query, the probability of predicting the entity in these facts will increase. According to common sense, TiRGN assumes that the closer the time of the facts, the more significant the impact on the final results. Global history encoder takes the relevant facts at all previous timestamps into account to avoid missing entities or relations that have not appeared at the adjacent timestamps. Periodicity is introduced into the decoder by the time vector generator to find possible periodic facts. Finally, TiRGN combines the periodic local and global decoders to realize the trade-off between the importance of local and global historical facts.

3.3 Local Recurrent Encoder

The local recurrent encoder focuses on the adjacent histories. For each query $(s, r, ?, t)$, we consider the subgraphs $\mathcal{G}_{m-1:t-1}$ of m adjacent timestamps. We aggregate and transfer the KG information from spatial and temporal views, respectively. Specifically, the graph convolutional network (GCN) is used to make single-step aggregation and the gated recurrent unit (GRU) is used among multiple timestamps to perform multi-step evolution.

Single-Step Aggregation

At each timestamp, we want to cover the facts related to the entity in the query as many as possible. Therefore, we design a one-dimensional convolution-based GCN, a multi-relation aggregator, to merge multiple relations and multi-hop neighbor information at a single timestamp. Compared with RE-GCN which sums relation embedding to entity embedding in GCN to make single-step aggregation, our TiRGN uses the one-dimensional convolution on entity embedding and relation embedding and thus can merge them better. The aggregator is defined as:

$$\mathbf{h}_{o,t}^{l+1} = \sigma\left(\sum\nolimits_{(s,r,o)\in\mathcal{F}_t} \frac{1}{c_o}\mathbf{W}_r^l\left(\psi(\mathbf{h}_{s,t}^l, \mathbf{r}_t)\right) + \mathbf{W}_o^l\mathbf{h}_{o,t}^l\right) \tag{1}$$

where $\mathbf{h}_{s,t}^l$, $\mathbf{h}_{o,t}^l$ denote the l^{th} layer embeddings of entities s, o at t, $\mathbf{W}_r^l$, $\mathbf{W}_o^l$ denote learnable weights and $\mathbf{W}_r^l$ is relation-specific, c_o is a normalizing factor equal to the in-degree of o, σ is the RReLu activation function, and ψ is the one-dimensional convolution operator. Note that when an entity does not have any relation with other entities in the subgraph, there will still be a self-loop edge to update it.

Multi-Step Evolution

For each query, in order to include the sequential dependencies of subgraphs at the previous timestamps, we use a double recurrent mechanism to update the representations of entities and relations progressively, i.e., entity-oriented GRU and relation-oriented GRU, so that it can obtain information at more distant timestamps. Entity-oriented GRU is used to update embeddings of entities in the sequence of subgraphs:

$$\mathbf{H}_t = \text{GRU}\left(\mathbf{H}_{t-1}, \mathbf{H}_{t-1}^{GCN}\right) \tag{2}$$

where $\mathbf{H}_t, \mathbf{H}_{t-1} \in \mathbb{R}^{|\mathcal{E}|\times d}$ are the d-dimensional entity embedding matrices at t and $t-1$, and $\mathbf{H}_{t-1}^{GCN} \in \mathbb{R}^{|\mathcal{E}|\times d}$ is the entity embedding matrix after singe-step aggeration at $t-1$. For relations, in order to maintain the consistency with the update of the entity embedding in the subgraph sequence, relation-oriented GRU is also used for update:

$$\mathbf{r}_t' = [\text{pooling}\left(\mathbf{H}_{t-1}, \mathcal{H}_{r,t}\right); \mathbf{r}] \tag{3}$$

$$\mathbf{R}_t = \text{GRU}\left(\mathbf{R}_{t-1}, \mathbf{R}_t'\right) \tag{4}$$

where $\mathcal{H}_{r,t}$ is all entities connected to $\mathbf{r}$ at t, $\mathbf{r}'_t$ is obtained by $\mathbf{H}_{t-1}$ and $\mathcal{H}_{r,t}$ using mean pooling operation, $\mathbf{R}'_t$ consists of $\mathbf{r}_t$ of all relations, and $\mathbf{R}_t, \mathbf{R}_{t-1} \in \mathbb{R}^{|\mathcal{R}|\times d}$ are relation embedding matrices at t and $t-1$. $\mathbf{R}_t$ is finally updated by $\mathbf{R}_{t-1}$ and $\mathbf{R}'_t$ through relation-oriented GRU.

3.4 Global History Encoder

The global history encoder is designed to get the repetitive global candidate facts, so as to provide global constraints for scoring in the decoder. For each query $(s, r, ?, t)$ or $(s, ?, o, t)$, we use this encoder to obtain candidate one-hop or multi-hop entities and relations. It is worth noting that, unlike CyGNet, we only consider whether the entity or relation has appeared before, without considering the frequency of its occurrence. We believe that directly using frequency as features may mislead prediction as the fact happened long ago does not necessarily occur in the future. The local recurrent encoder have been able to capture the impact of frequency of recent facts. This module is more to narrow the scope of prediction and avoid omissions rather than directly determining the final result. Specifically, we traverse all the subgraphs $\mathcal{G}_{0:t-1}$ before t and get the query results $\{c_0^{s,r}, c_1^{s,r}, \dots, c_{t-1}^{s,r}\}$, $\{c_0^{s,o}, c_1^{s,o}, \dots, c_{t-1}^{s,o}\}$. Then we take the union of the set of candidate entities at timestamp t:

$$\mathcal{C}_t^{s,r} = c_0^{s,r} \cup c_1^{s,r} \cup \dots \cup c_{t-1}^{s,r} \quad (5)$$

Therefore, for the query $(s, r, ?, t)$, the candidate entity matrix $\mathbf{M}_t^{s,r} \in \mathbb{Z}^{|\mathcal{E}|\times|\mathcal{R}|\times|\mathcal{E}|}$ assigns the values of positions existent in $\mathcal{C}_t^{s,r}$ to 1 and the non-existent to 0. The same operations and settings are also used for the candidate relation set $\mathcal{C}_t^{s,o}$ and candidate relation matrix $\mathbf{M}_t^{s,o} \in \mathbb{Z}^{|\mathcal{E}|\times|\mathcal{E}|\times|\mathcal{R}|}$. Although these two matrices have large dimensions, they are both sparse $(0, 1)$-matrices, so they have a low space complexity and time complexity during access. In addition, the global history encoder can be used for different levels of candidate fact records, including one-hop and multi-hop. At present, we only consider the one-hop candidate set.

3.5 Time Guided Decoder

After getting the embeddings of the local entities and relations, as well as the candidate sets of global entities and relations, we use the local and global decoders to score the facts. Periodic and non-periodic time vectors guide the decoders to incorporate the periodicity of facts into the model when calculating local and global scores.

Periodic and Non-Periodic Time Vectors

Some facts happen periodically throughout the timeline, such as presidential elections, while some facts are more likely to happen within a certain period of time, e.g., the elected president will participate in more activities in a certain period of time. Therefore, it makes sense to consider both the periodicity and non-periodicity of the historical facts. We design the periodic and non-periodic time vectors, respectively:

$$\mathbf{v}_t^p = f(\boldsymbol{\omega}_p t + \boldsymbol{\varphi}_p) \quad (6)$$

$$\mathbf{v}_t^{np} = \boldsymbol{\omega}_{np} t + \boldsymbol{\varphi}_{np} \quad (7)$$

where $\mathbf{v}_t^p$ and $\mathbf{v}_t^{np}$ are d-dimensional periodic and non-periodic time vectors, respectively, $\boldsymbol{\omega}_p$, $\boldsymbol{\varphi}_p$, $\boldsymbol{\omega}_{np}$ and $\boldsymbol{\varphi}_{np}$ are learnable parameters, and f is a periodic activation function. We chose the sine function as the periodic function because the sine function is expected to work well when extrapolated to future and out-of-sample data [Vaswani *et al.*, 2017]. When $f = \sin$, $\boldsymbol{\omega}_p$ and $\boldsymbol{\varphi}_p$ are the frequency and the phase-shift of the sine function. The period of $\mathbf{v}_t^p$ is $2\pi/\boldsymbol{\omega}_p$, so it has the same value at t and $t + 2\pi/\boldsymbol{\omega}_p$. In addition, referring to Time2Vec [Kazemi *et al.*, 2019], we can also prove that the both time vectors are invariant to the scaling of the time interval and adapt to datasets with different time intervals.

Time-ConvTransE/Time-ConvTransR

After obtaining time vectors, in order to perform entity prediction and relation prediction simultaneously, we design Time-ConvTransE and Time-ConvTransR inspired by ConvTransE [Shang *et al.*, 2019] for the two tasks, respectively. Specifically, the decoder performs one-dimensional convolution on concatation of four embeddings (entity embedding $\mathbf{h}_t^s$, realtion embedding $\mathbf{r}_t$, two time embedding $\mathbf{v}_t^p, \mathbf{v}_t^{np}$) and scores the resulting representation. Formally, the convolution operator is computed as follows:

$$\begin{aligned} m_c^n =& m_c\left(\mathbf{h}_t^s, \mathbf{r}_t, \mathbf{v}_t^p, \mathbf{v}_t^{np}, n\right) \\ =& \sum\nolimits_{\tau=0}^{K-1} \mathbf{w}_c(\tau, 0)\hat{\mathbf{h}}_t^s(n+\tau) + \mathbf{w}_c(\tau, 1)\hat{\mathbf{r}}_t(n+\tau) \\ &+ \mathbf{w}_c(\tau, 2)\hat{\mathbf{v}}_t^p(n+\tau) + \mathbf{w}_c(\tau, 3)\hat{\mathbf{v}}_t^{np}(n+\tau) \end{aligned} \quad (8)$$

where c is the number of convolution kernels, K is the kernel width, $n \in [0, d]$ indicates the entries in the output vector, and $\mathbf{w}_c$ are learnable kernel parameters. $\hat{\mathbf{h}}_t^s$, $\hat{\mathbf{r}}_t$, $\hat{\mathbf{v}}_t^p$ and $\hat{\mathbf{v}}_t^{np}$ are padding versions of $\mathbf{h}_t^s$, $\mathbf{r}_t$, $\mathbf{v}_t^p$ and $\mathbf{v}_t^{np}$, respectively. Similar to the improvement from TransE to TTransE, the convolution operation integrates time information while maintaining the translational property of embeddings. Thus, each convolution kernel forms an output vector $\mathbf{m}_c\left(\mathbf{h}_t^s, \mathbf{r}_t, \mathbf{v}_t^p, \mathbf{v}_t^{np}\right) = [m_c^0, m_c^1, \dots, m_c^{d-1}]$ can be aligned to get the matrix $\mathbf{M}_{conv} \in \mathbb{R}^{c\times d}$.

After nonlinear one-dimensional convolution, the final output of Time-ConvTransE is defined as follows:

$$\psi\left(\mathbf{h}_t^s, \mathbf{r}_t, \mathbf{v}_t^p, \mathbf{v}_t^{np}\right) = \mathrm{ReLu}\left(\mathrm{vec}\left(\mathbf{M}_{conv}\right)\mathbf{W}\right)\mathbf{H}_t^o \quad (9)$$

where vec is a feature map operator, and $\mathbf{W} \in \mathbb{R}^{cd\times d}$ is a matrix for linear transformation. Time-ConvTransR calculates the scores in the same way, only replacing $\mathbf{r}_t$ with $\mathbf{h}_t^o$.

3.6 Scoring Function and Training Objective

Since the impact of local and global historical facts may be different, we weight their importance through a variable factor. If $\mathbf{h}_t$ and $\mathbf{r}_t$ in Eq. (9) are the entity and relation embeddings obtained by the local recurrent encoder, then we get the output of local time-guided decoder. The output of the global decoder needs to mask the position where the value of the candidate matrix is equal to 0. After calculating the local probability decoding score $\mathbf{p}^{local}$ and the global probability decoding score $\mathbf{p}^{global}$ by softmax activation function, the final score is obtained by summing in proportion:

$$\mathbf{p}^{score} = \mathrm{softmax}(\psi^{score}) \quad (10)$$

$$\mathbf{p}^{final} = \alpha \times \mathbf{p}^{global} + (1-\alpha) \times \mathbf{p}^{local} \quad (11)$$

Model	ICE18				ICE14				ICE05-15			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RGCRN [2018]	28.02	18.62	31.59	46.44	38.48	28.52	42.85	58.10	44.56	34.16	50.06	64.51
RE-NET [2020]	29.78	19.73	32.55	48.46	39.86	30.11	44.02	58.21	43.67	33.55	48.83	62.72
CyGNet [2021]	27.12	17.21	30.97	46.85	37.65	27.43	42.63	57.90	40.42	29.44	46.06	61.60
TANGO [2021b]	28.97	19.51	32.61	47.51	–	–	–	–	42.86	32.72	48.14	62.34
xERTE [2021a]	29.31	21.03	33.51	46.48	40.79	32.70	45.67	57.30	46.62	37.84	52.31	63.92
RE-GCN [2021b]	32.62	22.39	36.79	52.68	42.00	31.63	47.20	61.65	48.03	37.33	53.90	68.51
TITer [2021]	29.98	22.05	33.46	44.83	41.73	32.74	46.46	58.44	47.60	38.29	52.74	64.86
TiRGN	**33.66**	**23.19**	**37.99**	**54.22**	**44.04**	**33.83**	**48.95**	**63.84**	**50.04**	**39.25**	**56.13**	**70.71**

Table 1: Performance (in percentage) for entity prediction task on ICEWS18, ICESW14 and ICEWS05-15 with time-aware metrics.

Model	WIKI				YAGO				GDELT			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RGCRN [2018]	65.79	61.66	68.17	72.99	65.76	62.25	67.56	71.69	19.37	12.24	20.57	33.32
RE-NET [2020]	58.32	50.01	61.23	73.57	66.93	58.59	71.48	86.84	19.55	12.38	20.80	34.00
CyGNet [2021]	58.78	47.89	66.44	78.70	68.98	58.97	76.80	86.98	20.22	12.35	21.66	35.82
TANGO [2021b]	53.04	51.52	53.84	55.46	63.34	60.04	65.19	68.79	19.66	12.50	20.93	33.55
xERTE [2021a]	73.60	69.05	78.03	79.73	84.19	80.09	88.02	89.78	19.45	11.92	20.84	34.18
RE-GCN [2021b]	78.53	74.50	81.59	84.70	82.30	78.83	84.27	88.58	19.69	12.46	20.93	33.81
TITer [2021]	73.91	71.70	75.41	76.96	87.47	80.09	89.96	90.27	18.19	11.52	19.20	31.00
TiRGN	**81.65**	**77.77**	**85.12**	**87.08**	**87.95**	**84.34**	**91.37**	**92.92**	**21.67**	**13.63**	**23.27**	**37.60**

Table 2: Performance (in percentage) for entity prediction task on WIKI, YAGO and GDELT with time-aware metrics.

Model	ICE18	ICE14	ICE05-15	WIKI	YAGO	GDELT
RGCRN	42.19	41.92	44.18	96.47	95.30	20.99
RE-GCN	45.66	45.29	47.07	98.90	98.80	21.47
TiRGN	**46.64**	**47.71**	**48.17**	**99.04**	**99.30**	**24.91**

Table 3: Performance (in percentage) for relation prediction task with time-aware MRR.

where variable factor $\alpha \in [0, 1]$. Entity prediction and relation prediction calculate the score in the same way, but can take different values of α.

We regard both entity and relation predictions as multi-label learning problems and train them together. Therefore, the total loss that contains the loss of entity prediction L^e and the loss of relation prediction L^r is formalized as:

$$
\begin{aligned}
L = L^e + L^r = &\sum\nolimits_{(s,r,o,t)\in\mathcal{G}} \mathbf{y}_t^e \log \mathbf{p}\left(o \mid s, r, t\right) \\
&+ \sum\nolimits_{(s,r,o,t)\in\mathcal{G}} \mathbf{y}_t^r \log \mathbf{p}\left(r \mid s, o, t\right)
\end{aligned} \tag{12}
$$

where $\mathbf{p}\left(o \mid s, r, t\right)$ and $\mathbf{p}\left(r \mid s, o, t\right)$ are the final probabilistic scores of entity and relation predictions. $\mathbf{y}_t^e \in \mathbb{R}^{|\mathcal{E}|}$ and $\mathbf{y}_t^r \in \mathbb{R}^{|\mathcal{R}|}$ are the label vectors for the two tasks, of which the element is 1 if the fact occurs, otherwise 0.

4 Experiments

4.1 Setup

Datasets

We use six TKG datasets to evaluate TiRGN on entity prediction and relation prediction tasks, including ICEWS14 [García-Durán *et al.*, 2018], ICEWS18 [Jin *et al.*, 2020], ICEWS05-15 [García-Durán *et al.*, 2018], GDELT [Jin *et al.*, 2020], WIKI [Leblay and Chekol, 2018], and YAGO [Mahdisoltani *et al.*, 2015]. We follow the preprocessing strategy for datasets in RE-NET.

Evaluation Metrics

We adopt two widely used metrics to evaluate the model performance on TKG reasoning, mean reciprocal rank (MRR) and Hits@k. MRR is the average reciprocal values of the ranks of the true entity candidates for all queries, and Hits@k represents the proportion of times that the true entity candidates appear in the top k of the ranked candidates. Some recent works mention that the filtered setting [Bordes *et al.*, 2013] is not suitable for extrapolation on TKG reasoning [Han *et al.*, 2021a; Han *et al.*, 2021b]. Therefore, we use the time-aware filtered setting to report the experimental results.

Implementation Details

For all the datasets, the embedding dimension d is set to 200. The number of one-dimensional convolution-based GCN layers is set to 2 and the dropout rate for each layer is set to 0.2. The optimal local history lengths m are set to 10, 9, 15, 2, 1, and 7 for ICEWS18, ICEWS14, ICEWS05-15, WIKI, YAGO, and GDELT, respectively. Similar to RE-GCN, static graph constraints are added for ICEWS14, ICEWS18, and ICEWS05-15. For time-guided decoders, the number of channels is set to 50 and the kernel size is set to 4×3. We tried multiple α values from 0.1 to 0.9 and finally selected 0.3 as the global weight for all the datasets. Adam is used for parameter learning, and the learning rate is set to 0.001. The code is available in https://github.com/Liyyy2122/TiRGN.

Model	ICE18	ICE14	ICE05-15	WIKI	YAGO	GDELT
le	32.65	42.10	48.54	78.64	83.21	20.08
ge	29.87	39.24	40.57	54.46	60.05	21.13
ge+fre	29.21	38.48	40.36	54.23	58.02	19.90
td	28.76	38.16	38.22	47.42	56.22	19.64
le+td	32.75	42.67	48.67	78.88	83.31	21.20
ge+td	30.03	39.81	40.41	54.47	61.36	21.19
le+ge	33.47	43.82	49.37	81.40	86.18	21.43
TiRGN	**33.66**	**44.04**	**50.04**	**81.65**	**87.95**	**21.67**

Table 4: Results (in percentage) of ablation studies with time-aware MRR. **le** is for local recurrent encoder, **ge** is for global history encoder, **td** is for periodic time decoder, **fre** is for frequency.

4.2 Results

Results on Entity Prediction

The results of the entity prediction task are shown in Tables 1 and 2. On the six benchmark datasets, TiRGN continuously outperforms all baselines. Specifically, the performance of TiRGN is better than CyGNet because it not only considers the global historical repetitive facts but also pays attention to the facts at the adjacent timestamps and the periodicity of facts. RGCRN, RE-NET, TANGO, xERTE and RE-GCN consider the facts of adjacent timestamps and show strong performance in the experiment. Nevertheless, TiRGN uses a one-dimensional convolution-based GCN and double recurrence mechanism to capture more comprehensive structural features and historical dependency. Therefore, TiRGN is superior to these models that capture a single historical characteristics. Due to the historical facts search strategy based on reinforcement learning, TITer performs well on YAGO. The historical facts search strategy of TITer is suitable for the datasets with fewer timestamps and facts. However, once the dataset has too many timestamps, TITer will cause performance degradation due to its inability to model the ample search space, such as on the GDELT. In contrast, TiRGN only obtains the set of candidate entities and relations that record the occurrence of global facts without worrying about excessive complexity. Therefore, TiRGN has more substantial applicability for different datasets.

As shown in Tables 1 and 2, TiRGN has significantly improved the performance on ICEWS05-15 and GDELT with a large number of timestamps. It proves the effectiveness of capturing longer local historical dependencies through a double recurrent network. Besides, TiRGN has the most obvious effect on the datasets with more facts such as ICEWS05-15, WIKI, and GDELT, which also demonstrates that it is necessary to consider different historical characteristics when historical information is sufficient.

Results on Relation Prediction

Since some models are not designed for relation prediction, we select temporal models that can be used for relation prediction. As shown in Table 3, TiRGN performs better than all baselines. TiRGN achieves limited improvement on datasets that is easy to achieve high performance due to few relations. For the datasets with a larger number of relations, the performance of TiRGN improves significantly, which verifies the observation results mentioned in the entity prediction again.

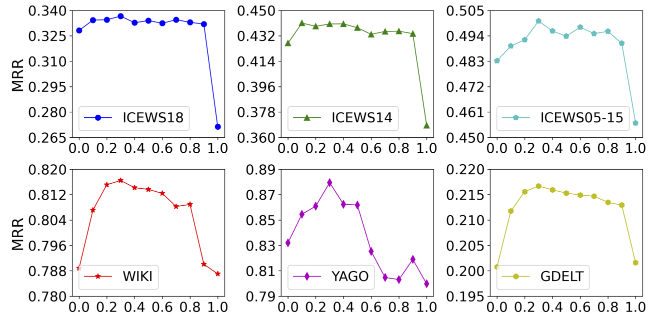

Figure 3: Sensitivity analysis results of hyperparameter α.

4.3 Ablation Studies

To better understand the effectiveness of different model components that capture the corresponding historical characteristics, we conduct ablation studies. As shown in Table 4, the local recurrent encoder **(le)** has the greatest impact on performance, which indicates that adjacent historical facts are crucial for the prediction. The global history encoder **(ge)** has a consistent impact on all the datasets, which shows the necessity of avoiding omissions for prediction. Besides, a single global history encoder that directly uses frequency **(ge+fre)** as features performs worse than that without frequency, which is in line with our assumptions. Since periodic facts are special cases in the datasets, the periodic time decoder **(td)** does not greatly improve the performance, but it will not reduce the performance even for non-periodic facts. Therefore, these results further show that different historical characteristics are all helpful to the prediction.

4.4 Sensitivity Analysis

To explore the importance of global and local historical facts to the prediction results, we conduct a sensitivity analysis. α is a variable trade-off factor between global and local historical facts. As shown in Figure 3, neither ignoring the facts at adjacent timestamps nor ignoring the global repetitive facts can make effective prediction, which further demonstrates the necessity of combining local and global historical patterns in TiRGN. Besides, the results show that the performance is better when the value of α is from 0.2 to 0.5. This phenomenon shows that the facts of adjacent timestamps are more important than global repeated facts. By adjusting the value of α on the validation set, TiRGN can obtain the best trade-off between local and global historical facts.

5 Conclusion

In this paper, we propose a model named TiRGN for TKG reasoning, which learns the representations of entities and relations by capturing multiple characteristics of historical facts. We combine a local encoder that captures the structural dependency of local histories with a global encoder that captures the pattern of global repetitive histories, so as to conduct reasoning through a time-guided decoder with periodicity. The experimental results on six benchmark datasets demonstrate the significant advantages and effectiveness of TiRGN on temporal entity and relation predictions. In addition, ablation experiments show that these characteristics of historical facts play positive roles in TKG reasoning.

Acknowledgements

This work was supported by the Shanghai Municipal Project 20511100900, the NSFC Projects 62076096 and 62006078, Shanghai Knowledge Service Platform Project ZF1213, STCSM Project 22ZR1421700 and the Fundamental Research Funds for the Central Universities.

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, pages 2787–2795, 2013.

[Deng *et al.*, 2020] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Dynamic knowledge graph based multi-event forecasting. In *KDD*, pages 1585–1595, 2020.

[García-Durán *et al.*, 2018] Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. In *EMNLP*, pages 4816–4821, 2018.

[Goel *et al.*, 2020] Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. Diachronic embedding for temporal knowledge graph completion. In *AAAI*, pages 3988–3995, 2020.

[Han *et al.*, 2021a] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *ICLR*, pages 1–24, 2021.

[Han *et al.*, 2021b] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *EMNLP*, pages 8352–8364, 2021.

[Jiang *et al.*, 2016] Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. Towards time-aware knowledge graph completion. In *COLING*, pages 1715–1724, 2016.

[Jin *et al.*, 2020] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *EMNLP*, pages 6669–6683, 2020.

[Kazemi *et al.*, 2019] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[Leblay and Chekol, 2018] Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *WWW*, pages 1771–1776, 2018.

[Li *et al.*, 2021a] Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *ACL*, pages 4732–4743, 2021.

[Li *et al.*, 2021b] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *SIGIR*, pages 408–417, 2021.

[Ma *et al.*, 2019] Yunpu Ma, Volker Tresp, and Erik A. Daxberger. Embedding models for episodic knowledge graphs. *J. Web Semant.*, 59:100490, 2019.

[Mahdisoltani *et al.*, 2015] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*, pages 1–11, 2015.

[Sadeghian *et al.*, 2021] Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. Chronor: Rotation based temporal knowledge graph embedding. In *AAAI*, pages 6471–6479, 2021.

[Schlesinger, 1999] Arthur M Schlesinger. *The cycles of American history*. Houghton Mifflin Harcourt, 1999.

[Seo *et al.*, 2018] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *ICONIP*, pages 362–373, 2018.

[Shang *et al.*, 2019] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *AAAI*, pages 3060–3067, 2019.

[Sun *et al.*, 2021] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *EMNLP*, pages 8306–8319, 2021.

[Trivedi *et al.*, 2017] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *ICML*, pages 3462–3471, 2017.

[Trivedi *et al.*, 2019] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *ICLR*, pages 1–25, 2019.

[Trompf, 1979] Garry Winston Trompf. *The idea of historical recurrence in Western thought: From antiquity to the Reformation*. Univ of California Press, 1979.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Wu *et al.*, 2020] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L. Hamilton. Temp: Temporal message passing for temporal knowledge graph completion. In *EMNLP*, pages 5730–5746, 2020.

[Zhu *et al.*, 2021] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. In *AAAI*, pages 4732–4740, 2021.