Learn from Relational Correlations and Periodic Events for Temporal Knowledge Graph Reasoning

Ke Liang
National University of Defense Technology
Changsha, Hunan, China
liangke200694@126.com

Lingyuan Meng
National University of Defense Technology
Changsha, Hunan, China
mly_edu@163.com

Meng Liu
National University of Defense Technology
Changsha, Hunan, China
mengliuedu@163.com

Yue Liu
National University of Defense Technology
Changsha, Hunan, China
yueliu19990731@163.com

Wenxuan Tu
National University of Defense Technology
Changsha, Hunan, China
wenxuantu@163.com

Siwei Wang
National University of Defense Technology
Changsha, Hunan, China
wangsiwei13@nudt.edu.cn

Sihang Zhou
National University of Defense Technology
Changsha, Hunan, China
sihangjoe@gmail.com

Xinwang Liu*
National University of Defense Technology
Changsha, Hunan, China
xinwangliu@nudt.edu.cn

ABSTRACT

Reasoning on temporal knowledge graphs (TKGR), aiming to infer missing events along the timeline, has been widely studied to alleviate incompleteness issues in TKG, which is composed of a series of KG snapshots at different timestamps. Two types of information, *i.e.,* intra-snapshot structural information and inter-snapshot temporal interactions, mainly contribute to the learned representations for reasoning in previous models. However, these models fail to leverage (1) semantic correlations between relationships for the former information and (2) the periodic temporal patterns along the timeline for the latter one. Thus, such insufficient mining manners hinder expressive ability, leading to sub-optimal performances. To address these limitations, we propose a novel reasoning model, termed RPC, which sufficiently mines the information underlying the Relational correlations and Periodic patterns via two novel Correspondence units, *i.e.,* relational correspondence unit (RCU) and periodic correspondence unit (PCU). Concretely, relational graph convolutional network (RGCN) and RCU are used to encode the intra-snapshot graph structural information for entities and relations, respectively. Besides, the gated recurrent units (GRU) and PCU are designed for sequential and periodic inter-snapshot temporal interactions, separately. Moreover, the model-agnostic time vectors are generated by time2vector encoders to guide the time-dependent decoder for fact scoring. Extensive experiments on six benchmark datasets show that RPC outperforms the state-of-the-art TKGR models, and also demonstrate the effectiveness of two novel strategies in our model.

*Corresponding Author

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591711

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; **Logical and relational learning**; **Temporal reasoning**.

KEYWORDS

Temporal Knowledge Graph Reasoning; Extrapolation Relation Reasoning; Graph Learning

ACM Reference Format:
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 (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591711

1 INTRODUCTION

Knowledge graphs (KGs), as the representation of real-world facts and events, have facilitated countless knowledge-driven tasks, *e.g.,* information retrieval [23, 37, 56], natural language understanding [2, 5, 54], recommendation systems [1, 48, 49], healthcare AI [24, 32, 55], etc. However, since facts and events usually keep evolving over time, conventional KGs (static KGs) cannot meet the requirements to model such complex dynamics. Thus, the temporal knowledge graph (TKG) is designed, which is composed of a series of KG snapshots at different timestamps. With the temporal information, the facts in TKGs are usually in the format of quadruple, *i.e., (head entity, relation, tail entity, timestamp)*, where the timestamp is used

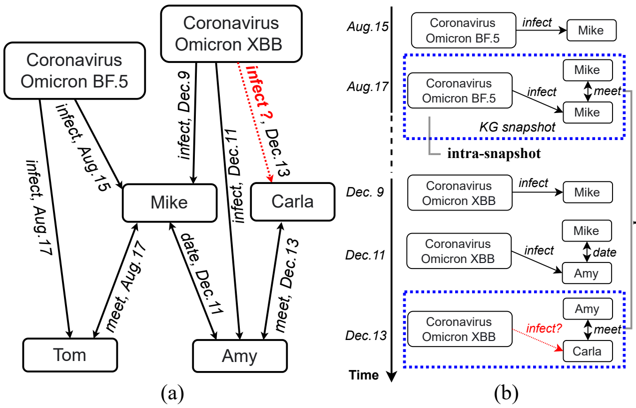


Figure 1: Illustration of temporal knowledge graph reasoning (TKGR). Sub-graphs (a) and (b) are two different views of the TKGs. By mining the logical patterns underlying the TKGs, TKGR models aim to infer the missing event, which is represented in red dotted edges. Besides, blue boxes indicate different KG snapshots and two gray lines demonstrate that "intra-" and "inter-" are used to describe the interactions "within" and "between" snapshots, respectively.

to indicate the specific time for the occurrence of facts. For example, *(Mike, date, Amy, Dec. 11)* represents that Mike has a date with Amy on December 11, as shown in Fig. 1 (a).

However, similar to conventional static KGs, TKGs also suffer from incompleteness issues [31]. To alleviate the problem, temporal knowledge graph (TKGR) models, aiming to infer missing events at specific timestamps, have been increasingly studied these years. According to the scope of the reasoning timestamp, there are two settings for TKGR, *i.e.,* interpolation reasoning, and extrapolation reasoning [28]. Given the timestamp scope $[0, T]$, the reasoning is performed within 0 to T in the former setting, while the reasoning timestamps are beyond the T in the latter one. In other words, extrapolation reasoning aims to infer future facts, which is more challenging and practical, and our work also falls into this type.

The key to inferring future facts and events is to fully mine the information underlying the TKGs. We find that two types of information, *i.e.,* intra-snapshot structural information and inter-snapshot temporal interactions, mainly contribute to the learned representations for reasoning in previous models. The former information describes the logical patterns within the graph structure for each KG snapshot, while the latter one describes the temporal interactions between KG snapshots at different timestamps. Recent TKGR models [19, 25, 29, 38], generally leverage the intra-snapshot structural information by only learning entity representations via graph neural network (GNN) models. Besides, inter-timestamp temporal interactions are commonly mined via recurrent neural network (RNN) models. Although proven effective, these methods can still be improved by leveraging the information in more sufficient manners.

Two main limitations exist in most of the previous models, which are illustrated as follows: **(1)** as for the intra-snapshot information, these models fail to leverage the **semantic correlations between relations**. The interactions between entities can be modeled as the edges in graphs, and such structural information can be easily get via most GNN models. While, the rich information underlying commonly seen relational correlations in TKGs still needs to be exploited, which will benefit the expressive ability of the models. For example, the relation *infect* will have a strong correlation with relations, such as *meet*, *date*, as *Coronavirus Omicron XBB* are highly contagious between two people contacting in person. **(2)** As for the inter-snapshot interactions, these models ignore mining the **periodic temporal information** along the timeline. The periodic events can be easily found in our world, such as the Olympic Games, and National Leader Election. We believe that the relevant information around such events in the past will play an important role in reasoning when similar events occur in the future since they will leave similar impacts. For example, there will exist a tourism boom in the venue per four years during Olympic Games.

To this end, we propose a novel encoder-decoder extrapolation temporal relation reasoning model, termed RPC, which sufficiently mines the information underlying the Relational correlations and Periodic patterns via two novel Correspondence units, *i.e.,* relational correspondence unit (RCU) and periodic correspondence unit (PCU). As for the intra-snapshot structural information, the relational graph convolutional network (RGCN) and RCU are used to encode the entities and relations, respectively. Specifically, in RCU, the relation is encoded based on the relational correspondence graph, which is reconstructed from the original graph according to the designed construction policy. Besides, as for the inter-snapshot temporal interactions, the gated recurrent units (GRU) are adopted for sequential temporal information, and the novel periodic correspondence units (PCU) are proposed to capture periodic temporal information within the specific scope, *i.e.,* k-Historical Windows, by calculating the correspondence scores between different KG snapshots. Moreover, the model-agnostic time vectors are generated by time2vector encoders to guide the time-dependent decoder for fact scoring. Extensive experiments on six benchmark datasets demonstrate the promising capacity of our RPC from four aspects, including superiority, effectiveness, transferability, and sensitivity. The main contributions are summarized as follows:

- We propose a novel encoder-decoder extrapolation temporal relation reasoning model, termed RPC, which sufficiently mines the information underlying relational correlations and periodic patterns in TKGs.
- We are the first to leverage the relational correlations intra-snapshot structural information for temporal knowledge graph reasoning via the relational correspondence unit (RCU). The relation representation is learned on the relational correspondence graphs reconstructed from original graphs based on the designed construction policy.
- Both sequential and periodic inter-snapshot temporal information is leveraged in our model. In particular, the novel periodic correspondence unit (PCU) is proposed for periodic temporal interactions within specific scope by calculating the correspondence scores between different KG snapshots.
- Extensive experiments on six datasets demonstrate the promising capacity of our RPC from four aspects, *i.e.,* superiority, effectiveness, transferability, and sensitivity.

2 RELATED WORK

2.1 Temporal Knowledge Graph Reasoning

More and more studies have focused on temporal knowledge graph reasoning in recent years, which can be roughly divided into two types according to the scope of queried timestamps, including interpolation reasoning and extrapolation reasoning [19, 28].

2.1.1 Interpolation Reasoning. Generally speaking, interpolation reasoning aims to infer the facts in the past. Most of the early attempts for TKGR are proposed in this setting [8–10, 21, 30, 39, 50–52]. For example, TTransE [10] is a variant of TransE [3], which treats both relation and time as translation between entities. Besides, TComplEx and TNTComplEx [21] are both developed based on ComplEx, which models the facts with additional time information as the fourth-order tensor. However, these models cannot scale well to extrapolation reasoning tasks.

2.1.2 Extrapolation Reasoning. Our work focuses on a more challenging and practical setting, *i.e.,* extrapolation reasoning, which aims to infer future facts based on historical facts. Know-Evolve [45] and DyREP [46] leverage the temporal point process to model the occurrence of facts. Based on them, GHNN [17] further integrates the subgraph information around the queried entity to assist reasoning. Later on, CyGNet [57] tends to capture repetitive patterns by modeling repetitive facts with the same head entity and relation. In addition, the specific neural ordinary differential equation is designed in TANGO [16] for continuous-time reasoning. Besides, xERTE [15] uses a subgraph sampling technique to construct interpretable reasoning graphs. Moreover, reinforcement techniques are used in TKGR to search for the most possible facts, such as CluSTeR [27], and TITer [43]. Moreover, there exists a majority quantity of models that employ graph neural network (GNN), and recurrent neural network (RNN) models to learn the representations for the graph structure in each KG snapshot and the temporal interactions between different KG snapshots separately, including RE-NET [19], RE-GCN [29], CEN [26] EvoKG [38], TiRGN [25], HIP [18]. Our model can also be classified into this type.

2.2 Structural Information Learning

The inter-snapshot interaction is usually modeled as graph structural information of the knowledge graph snapshot at the specific timestamp. Graph neural networks (GNNs) are regarded as effective models to mine the graph structural information [13, 22, 33–36, 47]. Thus, inspired by them, different graph neural network models are also adopted for the structural information in TKGR. The most commonly used are RGCN [40], and its variants [4, 44]. These models focus more on learning the representation of entities but omit the rich correlation semantic between different relations. Two recent attempts [6, 12] for static knowledge graph reasoning tend to mine such information based on different relational correspondence graphs, *i.e.,* the graph to describe the correspondence between relations. Inspired by them, we are the first work to extend the idea to the TKGR setting with specific relation-correspondence graphs, which provide more complete structural information for encoding and reasoning.

Table 1: The Summary of important notations

Notations	Descriptions
$\mathcal{G} = \{\mathcal{E}, \mathcal{R}, S_{\mathcal{G}}, \mathcal{T}\}$	temporal knowledge graph
$\mathcal{E}$	entity set
$\mathcal{R}$	relation set
$S_{\mathcal{G}}$	KG snapshot set
$\mathcal{T}$	timestamp set
$\mathcal{G}_t$	KG snapshot at timestamp t
e, r, t	entity, relation, timestamp
$\mathbf{E}, \mathbf{R}$	embedding matrices of entity and relation.
$\mathbf{T^p}, \mathbf{T^{np}}$	periodic and non-periodic time embeddings
$\mathbf{H_p}$	periodic historical embedding
W_p	periodic correspondence weight vector
k	scope size for historical window
α	trade-off hyperparameter
m	number of most corresponding events

2.3 Temporal Interaction Modeling

Compared to static KGR models, TKGR models need extra mechanisms to model the temporal interactions between different KG snapshots, which is also called intra-snapshot temporal information in this work. However, existing models focus more on the sequential temporal information between continuous KG snapshots. Researchers usually leverage recurrent neural network (RNN) models for it, and most of these models [19, 29, 38] can well mine such temporal information. However, we can easily find some cyclical and periodic events along the timeline, such as the Sport World Cup, Olympic Games, and National Leader Election. However, previous models will leave out periodic temporal information underlying the periodic events. Some recent works try to solve it based on the repetition frequency of concrete events, such as CyGNet [57], TiRGN [25], which is not intelligent and requires human efforts to high-frequency event selection. To conquer the limitation, the novel periodic correspondence unit (PCU) is designed in our work to capture the most corresponding timestamps in a heuristic manner.

3 METHOD

The details of our RPC will be introduced in this section. Before that, some important preliminaries are first declared.

3.1 Preliminary

A temporal knowledge graph (TKG) is defined as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, S_{\mathcal{G}}, \mathcal{T}\}$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ represent the set of entity, relation, and timestamp, respectively. Besides, we can also regard it as a series of static knowledge graphs, *i.e.,* snapshots, denoted as $S_{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \cdots, \mathcal{G}_T\}$. The snapshot at timestamp $t \in [0, T]$ is the set of fact quadruples, *i.e.,* $\mathcal{G}_t = \{(e_h, r_{h,t}, e_t, t)\}$. The main goal of our task, *i.e.,* the extrapolation relation reasoning on TKG is to predict the likelihood of the queried facts $(e^q_h, r^q_{h,t}, e^q_t, t^q)$, where $t^q > T$, based on the facts and events before this timestamp. Specifically, our model aims to score the queried fact $(e^q_h, r^q_{h,t}, ?, t^q)$ for the most convincible tail entity e^q_t, where the superscript q represents the queried facts. Note that the notations are summarized in Tab. 1. Besides, we observe the

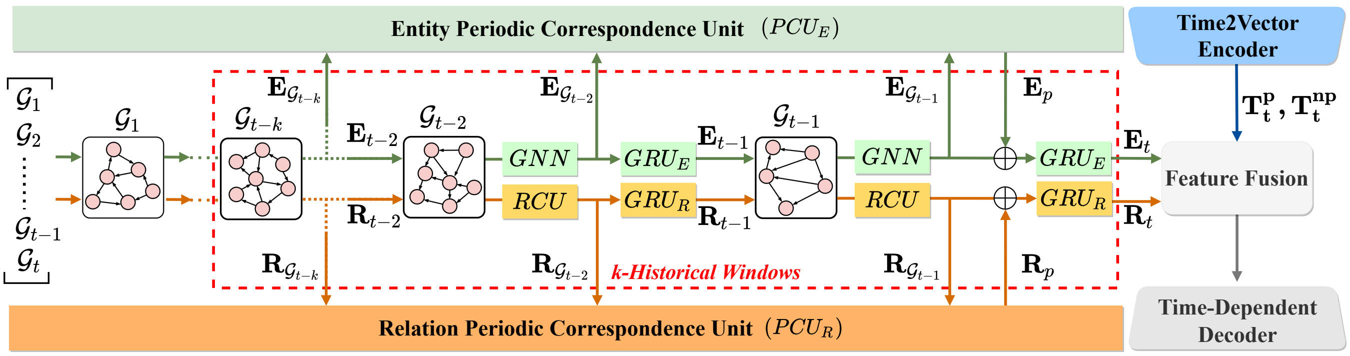


Figure 2: The framework of the proposed RPC, which sufficiently mines the information underlying the relational correlations and periodic patterns via two novel and simpel correspondence units, *i.e.,* relational correspondence unit (RCU) and periodic correspondence unit (PCU). Concretely, GNN and RCU are used to encode the intra-snapshot graph structural information for entities and relations, respectively. Besides, the GRU and PCU are designed for sequential and periodic inter-snapshot temporal interactions within the scope of k-Historical Windows, separately. Moreover, the model-agnostic time vectors are generated by time2vector encoders to guide the time-dependent decoder for fact scoring.

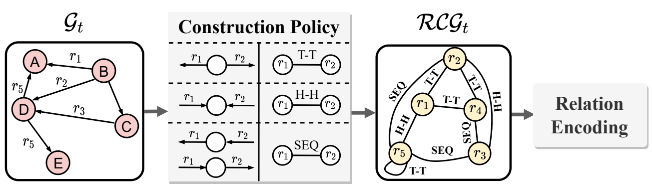


Figure 3: Illustration of the relational correspondence unit (RCU). RCU contains two main procedures, *i.e.,* relational correlation graph ($\mathcal{RCG}$) construction and relation encoding. The construction policy is provided above, together with an example for $\mathcal{RCG}$ construction.

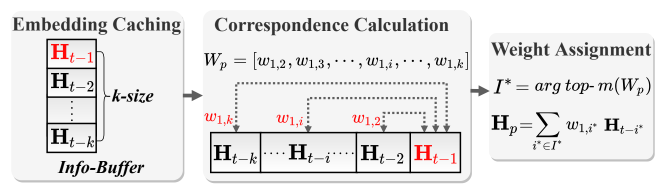


Figure 4: Illustration for the periodic correspondence unit (PCU), which contains three procedures, *i.e.,* embedding caching, correspondence calculation, and weight assignment. H represents the embedding, and it can be substituted as the entity and relation embedding (E, R) for PCU$_R$ and PCU$_E$.

existing TKGR models mainly leverage two types of information for reasoning, *i.e.,* intra-snapshot structural information and inter-snapshot temporal information as shown in Fig. 1 (b), which are defined below for a better understanding of our model.

Intra-snapshot Structural Information. *Given a TKG, the information within each KG snapshot is called intra-snapshot information. Since this information is usually mined as graph structural representations, it is also named intra-snapshot structural information, which can be further learned from two sources, i.e., entities and relations in KG snapshots.*

Inter-snapshot Temporal Information. *Given a TKG, the information between different KG snapshots is called inter-snapshot information. This information is usually modeled as temporal interactions, which can be divided into two types, i.e., sequential temporal information between continuous KG snapshots and periodic temporal information between similar periodic KG snapshots at different timestamps.*

3.2 Model Overview

To solve the extrapolation relation reasoning task, RPC is proposed in this work. In general, it is an encoder-decoder model, but in a more fine-grained encoding and decoding manner compared to previous TKGR models. As for the intra-snapshot structural information, the relational graph convolutional network (RGCN) and RCU are used to encode the entities and relations, respectively. Besides, as for the inter-snapshot temporal information, the gated recurrent units (GRU) are designed for sequential temporal information, and novel periodic correspondence units (PCU) are proposed to capture periodic temporal information. Moreover, the model-agnostic time vectors are generated by time2vector encoders to guide the time-dependent decoder for fact scoring. The overall framework of RPC is shown in Fig. 2, and more details will be illustrated as follow.

3.3 Intra-snapshot Structural Representation Learning

The graph structural information underlying the specific snapshot, *i.e.,* static knowledge graph, can be mined from two different sources, including entities and relations. Specifically, the graph neural network (GNN) is adopted for entity representation learning, while a novel relational correspondence unit (RCU) is designed for relation representation learning. More details for these two modules will be introduced as follow.

One-Dimensional Convolution-based RGCN Encoder. Inspired by [25, 40], we leverage a one-dimensional convolution-based RGCN encoder for entity encoding at each timestamp. The aggregation function is defined in Eq. 4.

$$\mathbf{h}_{e_t,t}^{l+1} = \sigma\left(\sum_{(e_h,r_{h,t},e_t,t)\in\mathcal{G}_t} \frac{1}{c_{e_h,r_h,t}}\mathbf{h}_{e_t,t}^{l}W_{r,t}^{l} + \mathbf{h}_{e_h,t}^{l}W_0^{l}\right), \quad (1)$$

where $\boldsymbol{h}_{e_h,h}^{l}$ and $\boldsymbol{h}_{e_t,t}^{l}$ is the embedding of entity e_h and e_t of the l^{th} layer at t^{th} KG snapshot. $\mathcal{N}_{i,r,t}$ denotes the set of neighbor indices of entity e_i under the relation r at t^{th} timestamp. $c_{e_h,r,t} = |\mathcal{N}_{e_h,r,t}|$ is a normalization constant. $\boldsymbol{W}_0^l$ and $\boldsymbol{W}_r^l$ are the wight parameters. $\sigma(\cdot)$ is a activation function.

Based on it, we can further get the entity representation for the KG snapshot at t timestamp $\mathbf{E}_{\mathcal{G}_t}$ by catenating all the entity representations based on Eq. 2.

$$\mathbf{E}_{\mathcal{G}_t} = \bigoplus_{e\in\mathcal{E}_t}\mathbf{h}_{e,t}, \quad (2)$$

Relational Correspondence Unit. The specific relational correspondence unit is designed for mining the semantic correlations between relations in RPC (See Fig. 3). It contains two main procedures, *i.e.,* relational correlation graph ($\mathcal{RCG}$) construction and relation encoding.

Inspired by recent attempts [6, 12] on static KGR, we design the task-specific construction policy to guide the construction procedure of relational correspondence graph $\mathcal{RCG}_t$ from the original KG snapshot $\mathcal{G}_t$. The $\mathcal{RCG}$ is an undirected graph, where each node represents one type of relation in the corresponding $\mathcal{G}$. Three types of edges are enrolled in our $\mathcal{RCG}$, including *T-T*, *H-H*, and *SEQ*. Concretely, the *T-T*, *i.e.,* tail to tail, refers to the situation that two edges in $\mathcal{G}$ are outgoing from the same entity. Similar to it, the *H-H*, *i.e.,* head to head, refers to the situation that two edges in $\mathcal{G}$ are in-going to the same entity. Besides, *SEQ*, *i.e.,* sequential, refers to the situation that two edges are connected end to end. Taking Fig. 3 as an example, we can get (r_1, *T-T*, r_2) in $\mathcal{RCG}$ according to (B, r_1, A) and (B, r_2, D) in $\mathcal{G}$. Besides, (r_1, *H-H*, r_5) in $\mathcal{RCG}$ is generated based on (B, r_1, A) and (D, r_5, A) in $\mathcal{G}$.

After building the $\mathcal{RCG}_t$, the relation representation at t^{th} KG snapshot is aggregated on $\mathcal{RCG}_t$ based on Eq. 3.

$$\mathbf{h}_{r,t} = \sigma\left(W\sum_{r_n\in\mathcal{N}_r}\frac{1}{N_r}\mathbf{h}_{n,t}\right), \quad (3)$$

where $\mathcal{N}_r$ denotes the set of connected nodes in $\mathcal{RCG}_t$ for relation r, and $\mathbf{h}_{r,t}$ is a learned embedding of the target relation.

Similar to entity representation learning, we can further get the relation representation of the KG snapshot at timestamp t $\mathbf{R}_{\mathcal{G}_t}$ by catenating all the relation representations as follows.

$$\mathbf{R}_{\mathcal{G}_t} = \bigoplus_{r\in\mathcal{R}_t}\mathbf{h}_{r,t}, \quad (4)$$

3.4 Inter-snapshot Temporal Interaction Modeling

The Inter-snapshot temporal interactions among different snapshots along the timeline can be roughly divided into two types, including sequential temporal interactions, and periodic temporal interactions. As for the type of interaction, it is widely studied by multiple works [29, 38]. Similar to them, we model such interactions by using the gated recurrent unit (GRU), *i.e.,* a lightweight recurrent neural network (RNN). As for the periodic temporal interactions, a periodic correspondence unit (PCU) is designed to capture the most corresponding and similar KG snapshots from history, and further, integrate them to refine the final embedding for scoring. The details for each module will be illustrated below.

Gated Recurrent Units (GRU). Double-gated recurrent mechanisms are adopted to capture the sequential temporal information for entity and relation, separately. Two GRU models, *i.e.,* GRU_E and GRU_R, update the representations of entities and relations progressively based on the Eq. 5 and Eq. 6.

$$\mathbf{E}_t = \begin{cases} GRU_E(\mathbf{E}_{\mathcal{G}_{t-1}}, \mathbf{E}_{t-1}) & t\in[0, T-1] \\ GRU_E(\mathbf{E}_{\mathcal{G}_{t-1}}\oplus\alpha\cdot\mathbf{E}_p, \mathbf{E}_{t-1}) & t = T \end{cases}, \quad (5)$$

$$\mathbf{R}_t = \begin{cases} GRU_R(\mathbf{R}_{\mathcal{G}_{t-1}}, \mathbf{R}_{t-1}) & t\in[0, T-1] \\ GRU_R(\mathbf{R}_{\mathcal{G}_{t-1}}\oplus\alpha\cdot\mathbf{R}_p, \mathbf{R}_{t-1}) & t = T \end{cases}, \quad (6)$$

where $\mathbf{E}_t$ and $\mathbf{R}_t$ are the embedding of entity and relation. The subscript letter t indicates the timestamp of the embeddings, *i.e.,* t represents the t^{th} timestamp. $\mathbf{E}_{\mathrm{G}_{t-1}}$ and $\mathbf{R}_{\mathrm{G}_{t-1}}$ are the embeddings after encoding on the KG snapshots at the specific timestamp. $\mathbf{E}_p$ and $\mathbf{R}_p$ are the periodic historical embeddings generated by periodic correspondence units (PCU), and α is the trade-off weight of the periodic historical embeddings.

Periodic Correspondence Unit. Periodic correspondence unit (PCU) is a novel mechanism for mining the information underlying the periodic temporal interactions in history, which is not based on either frequency of specific events. The main idea of it is to capture the most similar snapshots and refine the final representation using the representations of these selected snapshots. Concretely, three steps are designed in PCU as shown in Fig. 4, including embedding caching, correspondence calculation, and weight assignment.

Embedding caching aims to store the embeddings of the snapshots at the latest k timestamps cropped by *k-Historical Window* (See Fig. 2). The k-size information buffer, *i.e.,* info-buffer, is used for caching, where k is a size for the user-defined scope.

Then, the correspondence weights between the representation of each cropped snapshot with the latest snapshot, *i.e.,* H_{t-1}, are calculated based on the similarity function in the correspondence calculation procedure. Different similarity functions can be used, but we select cosine similarity as the baseline in RPC (See Eq. 8).

$$W_p = \{w_{1,i}|i\in[t-k, t-1]\}, \quad (7)$$

$$w_{1,i} = cosine(readout(\mathbf{H}_{t-1}), readout(\mathbf{H}_i)), \quad (8)$$

where W is the weight vector constituted by various $w_{1,i}$, and $w_{1,i}$ represents the correspondence weight (See Fig. 4). *readout(·)* aims to get the representation vector based on the matrix, which is more suitable for cosine similarity calculation.

After that, within the weight assignment procedure, we pick up the *top-m* similar KG snapshots based on the weights (*i.e.,* similarity scores), and then generate the periodic historical embedding with the weighted concatenation. Eq. 9 and Eq. 10 illustrate the above

Table 2: Typical benchmark datasets for TKGR. # represents the quantity.

Dataset	# Entities	# Relations	# Timestamps	Time Interval	# Train Facts	# Validation Facts	# Test Facts
GDELT [53]	7,691	240	8,925	15 mins	1,033,270	238,765	305,241
ICEWS05-15 [11]	10,488	251	4,017	24 hours	386,962	46,092	46275
ICEWS14 [15]	7,128	230	365	24 hours	63,685	13,823	13,222
ICEWS18 [15]	23,033	256	7,272	24 hours	373,018	45,995	49,545
YOGA [8]	10,623	10	189	1 year	161,540	19,523	20,026
WIKI [8]	12,554	24	232	1 year	2,735,685	341,961	341,961

procedure.

$$I^* = arg\ top_m(W_p), \tag{9}$$

$$\mathbf{H}_p = \sum_{i^* \in I^*} w_{1,i^*} \cdot \mathbf{H}_{t-i^*}, \tag{10}$$

where I^* is composed by the indexes of the top m value in W. The corresponding weights with the chosen index in i^* contribute to the final periodic embedding $\mathbf{H}_p$. Note that $\mathbf{H}$ can be substituted by $\mathbf{E}$ and $\mathbf{R}$ for entity and relation, respectively.

3.5 Time-Dependent Decoding

Inspired by TiRGN [25], the time2vector [20] is adopted encoder to generate the time vectors, *i.e.,* periodic and non-periodic time vectors, to guide the decoders to incorporate the periodicity attributes within facts after getting the representations from intra-snapshot and inter-snapshot information.

Time2Vector Encoder. The periodic and non-periodic time-dependent vectors, *i.e.,* $\mathbf{T}_t^p$ and $\mathbf{T}_t^{np}$, are generated based on Eq. 11 and Eq. 12 by leveraging Time2Vec encoder [20].

$$\mathbf{T}_t^p = sin(\omega_p\, t + \phi_p), \tag{11}$$

$$\mathbf{T}_t^{np} = \omega_{np}\, t + \phi_{np}, \tag{12}$$

where ω_{np}, ω_p, ϕ_{np}, and ϕ_p are learnable parameters.

Feature Fusion. The fused representation $\mathbf{O}$ is generated by convolution operation used in [25] for four representations, *i.e.,* $\mathbf{E}_t$, $\mathbf{R}_t$, $\mathbf{T}_t^p$ and $\mathbf{T}_t^{np}$.

$$\mathbf{M}_c = \{m_c^i | i \in [0, d-1]\} \tag{13}$$

$$\begin{aligned} m_c^n = \sum_{\tau=0}^{K-1} & \mathbf{w}_c(\tau,0)\hat{\mathbf{E}}_t^s(n+\tau) + \mathbf{w}_c(\tau,1)\hat{\mathbf{R}}_t(n+\tau) \\ & + \mathbf{w}_c(\tau,2)\hat{\mathbf{T}}_t^p(n+\tau) + \mathbf{w}_c(\tau,3)\hat{\mathbf{T}}_t^{np}(n+\tau), \end{aligned} \tag{14}$$

where c, K, and n represent the number of convolutional kernels, the kernel width, and the entries in the output vector ranging from 0 to $d-1$, respectively. Meanwhile, $\mathbf{w}_c$ are learnable kernel parameters. Besides, we padding $\mathbf{E}_t^s$, $\mathbf{R}_t$, $\mathbf{T}_t^p$ and $\mathbf{T}_t^{np}$ to get $\hat{\mathbf{E}}_t^s$, $\hat{\mathbf{R}}_t$, $\hat{\mathbf{T}}_t^p$ and $\hat{\mathbf{T}}_t^{np}$, respectively. Each convolution kernel forms a vector $\mathbf{M}_c$, which can be further aligned to get the matrix $\mathbf{O}_t$.

Time-Dependent Decoder. The Time-ConvTransE in [25, 42] is adopted as the time-dependent decoder in RPC for fact scoring, which is defined as follows:

$$\mathbf{score} = softmax\left(ReLu\left(map\left(\mathbf{O}_t\right)\mathbf{W}\right)\mathbf{H}_t^o\right), \tag{15}$$

where *map* and $\mathbf{W}$ are the feature mapping operation and matrix for linear transformation, separately. Besides, $\mathbf{H}_t^o$ represents the initial embedding, which can be substituted as either entity embedding $\mathbf{E}_t^o$ or relation embedding $\mathbf{R}_t^o$.

3.6 Training Objective

The loss function L is designed for entity prediction, which is formalized below:

$$L = \sum_{(e_h, r_{h,t}, e_t, t) \in \mathcal{G}} \mathbf{y}_t^e log\ \mathbf{score}(e_t | e_h, r_{h,t}, t), \tag{16}$$

where $\mathbf{score}(e_t|e_h, r_{h,t}, t)$ is the probabilistic score for the facts calculated by Eq. 15. $\mathbf{y}_t^e$ is the label vectors, where the element is 1 if the fact occurs; otherwise, 0.

4 EXPERIMENT

In this section, the experimental settings are first introduced from four aspects, including datasets, evaluation metrics, implementation, and compared baselines. Then, we comprehensively analyze the proposed RPC also from four aspects by answering the following questions.

- **Q1: Superiority.** Does RPC outperforms the existing state-of-the-art existing temporal knowledge graph reasoning models?
- **Q2: Effectiveness.** Are the proposed relational correspondence units (RCU) and periodic correspondence units (PCU) effective in capturing the intra-snapshot and inter-snapshot information? Can they boost the performance?
- **Q3: Transferability.** Can the designed relational correspondence units (RCU) and periodic temporal units (PCU) be easily and effectively extended to other temporal knowledge graph reasoning models?
- **Q4: Sensitivity.** How does the performance fluctuation of RPC with different hyper-parameters?

We conduct experiments to answer the above questions. Specifically, answers of **Q1** to **Q4** are offered in Sec. 4.2 to 4.5.

4.1 Experiment Setting

4.1.1 Datasets and Evaluation Metrics. Six TKGR benchmark datasets are leveraged to evaluate our RPC, including GDELT [53], ICEWS05-15 [11], ICEWS14 [15], ICEWS18 [15], YOGA [8] and WIKI [8]. The statistics of these datasets are shown in Tab. 2. Two commonly used metrics are adopted to evaluate the performance, *i.e.,* mean reciprocal rank (MRR) and Hits@k. MRR indicates the ranks of

Table 3: Performance (in percentage) for entity prediction task on GDELT, ICEWS14, and ICEWS05-15. The best results are marked as boldfaced.

Model	GDELT				ICEWS14				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
RGCRN (2018)	19.37	12.24	20.57	33.32	38.48	28.52	42.85	58.10	44.56	34.16	50.06	64.51
RE-NET (2020)	19.55	12.38	20.80	34.00	39.86	30.11	44.02	58.21	43.67	33.55	48.83	62.72
xERTE (2020)	19.45	11.92	20.84	34.18	40.79	32.70	45.67	57.30	46.62	37.84	52.31	63.92
CyGNet (2021)	20.22	12.35	21.66	35.82	37.65	27.43	42.63	57.90	40.42	29.44	46.06	61.60
TANGO (2021)	19.66	12.50	20.93	33.55	–	–	–	–	42.86	32.72	48.14	62.34
RE-GCN (2021)	19.69	12.46	20.93	33.81	42.00	31.63	47.20	61.65	48.03	37.33	53.90	68.51
TITer (2021)	18.19	11.52	19.20	31.00	41.73	32.74	46.46	58.44	47.60	38.29	52.74	64.86
CEN (2022)	-	-	-	-	42.20	32.08	47.46	61.31	-	-	-	-
TiRGN (2022)	21.67	13.63	23.27	37.60	43.81	33.49	48.90	63.50	49.84	39.07	55.75	70.11
EvoKG (2022)	19.28	-	20.55	34.44	27.18	-	30.84	47.67	-	-	-	-
RPC (Ours)	**22.41**	**14.42**	**24.36**	**38.33**	**44.55**	**34.87**	**49.80**	**65.08**	**51.14**	**39.47**	**57.11**	**71.75**

Table 4: Performance (in percentage) for entity prediction task on ICEWS18, WIKI, and YAGO. The best results are marked as boldfaced.

Model	ICEWS18				WIKI				YAGO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
RGCRN (2018)	28.02	18.62	31.59	46.44	65.79	61.66	68.17	72.99	65.76	62.25	67.56	71.69
RE-NET (2020)	29.78	19.73	32.55	48.46	58.32	50.01	61.23	73.57	66.93	58.59	71.48	86.84
xERTE (2020)	29.31	21.03	33.51	46.48	73.60	69.05	78.03	79.73	84.19	80.09	88.02	89.78
CyGNet (2021)	27.12	17.21	30.97	46.85	58.78	47.89	66.44	78.70	68.98	58.97	76.80	86.98
TANGO (2021)	28.97	19.51	32.61	47.51	53.04	51.52	53.84	55.46	63.34	60.04	65.19	68.79
RE-GCN (2021)	32.62	22.39	36.79	52.68	78.53	74.50	81.59	84.70	82.30	78.83	84.27	88.58
TITer (2021)	29.98	22.05	33.46	44.83	73.91	71.70	75.41	76.96	87.47	80.09	89.96	90.27
CEN (2022)	31.50	21.70	35.44	50.59	78.93	75.05	81.90	84.90	-	-	-	-
TiRGN (2022)	33.58	23.10	37.90	54.20	80.05	75.15	84.35	87.56	87.95	84.34	91.37	92.92
EvoKG (2022)	29.28	-	33.94	50.09	68.03	-	79.60	85.91	68.59	-	81.13	92.73
RPC (Ours)	**34.91**	**24.34**	**38.74**	**55.89**	**81.18**	**76.28**	**85.43**	**88.71**	**88.87**	**85.10**	**92.57**	**94.04**

the inferred true entity within the queried candidates, and Hits@k represents the proportion of times that the true entity appears in the top k ranking candidates.

4.1.2 Implementation. All experiments are conducted based on a single NVIDIA 3090Ti. The dimension of the embedding is set to 200. The layer number of the RGCN encoder is set to 2, along with the dropout rate fixed as 0.2. The scope size k is set to 18, and the number of most similar snapshots m is set to 2. To fair comparison, static graph constraints are added for ICEWS14, ICEWS18, and ICEWS05-15 as previous works [25, 29]. The channel number for decoding is set to 50, and the kernel size is set to 4×3. The α values are selected as 0.01 for periodic historical embeddings. Adam is used for parameter learning with a learning rate of 0.001. We report the mean results of three times experiments as previous works do.

4.1.3 Compared Baselines. Eleven typical exploration TKGR models are selected as the compared baselines, including RGCRN [41], RE-NET [19], xERTE [15], CyGNet [57], TANGO [16], RE-GCN [29], TITer [43], CEN [26], EvoKG [38] and TiRGN [25]. Note that the results for other models are all obtained from previous papers.

4.2 Performance Comparison for Superiority (RQ1)

We compare our model with eleven other state-of-the-art models ranging from 2018 to 2022 on six benchmark datasets. Tab. 3 and Tab. 4 show that our RPC significantly outperforms other TKGR models for both the MRR and Hit@k (*i.e.,* Hits@1, Hits@3, and Hits@10) evaluation metrics. Compared to the second best performances, RPC on average makes about 2.4%, 2.97%, 1.9%, and 2.1%

Table 5: Ablation Study of RPC on ICEW14

Model	ICEWS14			
	MRR	Hits@1	Hits@3	Hits@10
RPC (baseline)	44.55	34.87	49.80	65.08
- RCU	43.82	34.11	48.27	64.26
- (PCU_E & PCU_R)	43.17	33.84	46.97	63.21
- PCU_E	43.76	33.54	47.98	63.92
- PCU_R	43.65	33.85	47.23	63.78
- (RCU & PCU_E & PCU_R)	42.17	32.17	46.32	62.73

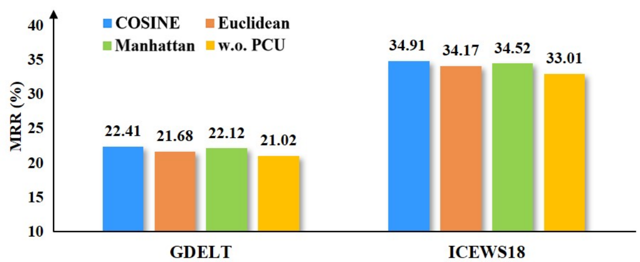


Figure 5: Performance comparison with different correspondence scoring functions.

performance improvements on MRR, Hits@1, Hits@3, and Hits@10, respectively. In particular, RPC improves Hits@1 values by about 5.9% and 5.4% on GDELT and ICEWS18 datasets. We can make such improvements on Hits@1 metrics, which means that our model indeed can infer the accurate entities within entity candidates. It further reveals the better expressive ability of our RPC compared to other models with the proposed relational correspondence and periodic correspondence units. Thus, based on the above results and analyses, we can assert that our model has good superiority.

4.3 Ablation Study for Effectiveness (RQ2)

The ablation studies are performed on ICEWS14 with all four evaluation metrics to investigate the effectiveness of two designed strategies, *i.e.,* relational correspondence unit (RCU) and periodic correspondence unit (PCU). The performances are shown in Tab. 5, where six sub-models are compared, including (1) the original RPC model, (2) RPC without RCU denoted as "- RCU", (3) RPC without any PCUs, denoted as "- PCU_R & PCU_R", (4) the model without PCU for entities denoted as "- PCU_E", (5) RPC without PCU_R, denoted as "- PCU_R", and (6) RPC without RCU and two types of PCUs, denoted as "- (RCU & PCU_R & PCU_R). Tab. 5 shows that the average MRR, Hits@1, Hits@3, and Hits@10 values are decreased by 1.3%, 1.4%. 2.5% and 1.5%, respectively. More concretely, we can get more observations when carefully analyzing the results in Tab. 5. Referring to the performance comparison of "- RCU" and "- PCU_R & PCU_R", we can find that PCU makes more performance improvements, which suggests that PCU is more effective for temporal knowledge graph reasoning (TKGR). Besides, the results of "- PCU_E" and "- PCU_R" indicates that the effectivenesses of periodic correspondence units for entities and relations are relatively equivalent to TKGR. Moreover, we also replace the $cosine(\cdot)$ correspondence scoring function

Table 6: Transfer Experiments of RCU and PCU on RE-GCN

Model	ICEWS14		WIKI	
	MRR	Hits@10	MRR	Hits@10
RE-GCN	42.00	61.65	78.53	84.70
+ RCU	42.65	62.16	79.01	85.93
+ PCU	42.81	62.74	79.97	86.02
+ (RCU & PCU)	43.92	63.27	80.82	86.81

as two other functions, *i.e.,* Euclidean distance $Euclidean(\cdot)$ [14], and Manhattan distance $Manhattan(\cdot)$ [7], and conduct the experiments on WIKI and YAGO. The MRR results are shown in Fig. 5. It indicates that all of the evaluated functions can contribute to the effective PCUs, where $cosine(\cdot)$ is the most effective one.

Based on promising results and the above analyses, we can assert that the proposed relational correspondence units (RCU) and periodic temporal units (PCU) can effectively capture the intra-snapshot relational structural information and inter-snapshot periodic temporal information, separately.

4.4 Transferability Analysis on RE-GCN (RQ3)

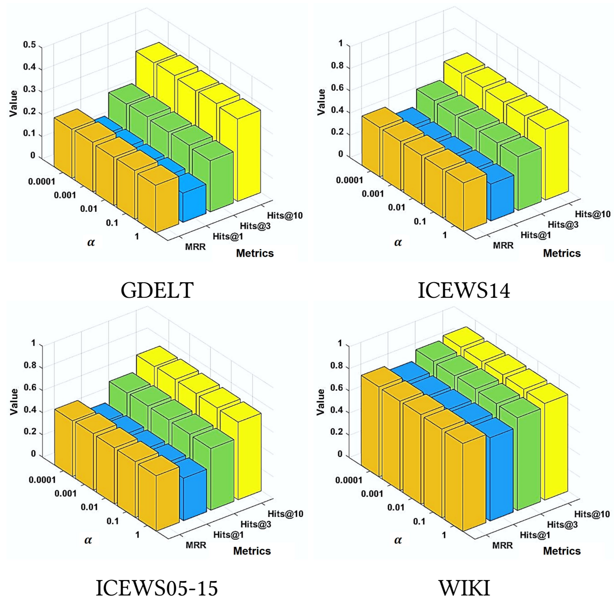
To demonstrate the transferability of our model, we extend the main ideas of RPC, *i.e.,* relational correspondence unit (RCU) and periodic correspondence unit (PCU), to RE-GCN [29]. Four sub-models are designed for the performance comparison, including (1) the original RE-GCN model, (2) RE-GCN combined with RCU denoted as "+ RCU", (3) RE-GCN combined with PCUs, denoted as "+ PCU", and (4) RE-GCN combined with both RCU and PCUs, denoted as "+ (RCU & PCU)". According to Tab. 6, we can find that similar to the conclusions in the previous section, both RCU and PCU can still benefit other TKGR models. For example, it is observed that 1.6% improvements for "+RCU", 1.9% improvements for "+PCU" and 4.6% improvements for "+ (RCU & PCU)" ON MRR on ICEWS14. *i.e.,* It demonstrates that the proposed mechanisms are kind of model-agnostic, which means that they can be easily transferred to other models. Besides the transferability, it also proves the effectiveness of our module from different aspects.

4.5 Hyper-parameter Analysis on α (RQ4)

We investigate the influence of the hyper-parameter trade-off weight α on four datasets, *i.e.,* GDELT, ICEWS14, ICEWS05-15, WIKI, for all four evaluation metrics, *i.e.,* MRR, Hits@1, Hits@3, Hits@10. As for the scope of the hyper-parameter, α is selected in $\{0.0001, 0.001, 0.01, 0.1, 1\}$ for all four datasets. We observe that the performance for all the evaluation metrics will not fluctuate greatly when α is varying in Fig. 6. It demonstrates that in general RPC is insensitive to α, but it is more possible for us to find the best α from 0.0001 to 0.001. For example, The best performances are reached when $\alpha = 0.001$ for GDELT and ICEWS05-15, and $\alpha = 0.0001$ for ICEWS14 and WIKI.

5 CONCLUSION

In this paper, we propose a novel encoder-decoder TKGR model, termed RPC, for extrapolation reasoning. It can sufficiently mine the information underlying the Relational correlations and Periodic



GDELT ICEWS14

ICEWS05-15 WIKI

Figure 6: Sensitivity analysis of hyper-parameter α on GDELT, ICEWS14, ICEWS05-15, and WIKI.

patterns via two novel Correspondence units, *i.e.,* relational correspondence unit (RCU) and periodic correspondence unit (PCU). As the first TKGR model to leverage the relational correlations intra-snapshot structural information, we design the relational correspondence unit (RCU) for learning the relation representations on the relational correspondence graphs reconstructed from original graphs based on the construction policy. Moreover, both sequential and periodic inter-snapshot temporal information is leveraged in our model. In particular, the novel periodic correspondence unit (PCU) is proposed for periodic temporal interactions within specific scope by calculating the correspondence scores between different KG snapshots. Extensive experiments on six benchmark datasets demonstrate the promising capacity of our RPC from four aspects, including superiority, effectiveness, transferability, and sensitivity. In the future, we plan to design specific strategies to adaptive select the size k of historical windows for different datasets, thus making the model more intelligent.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by the National Key R&D Program of China (Grant No. 2020AAA0107703).

REFERENCES

[1] Wendong Bi, Bingbing Xu, Xiaoqian Sun, Zidong Wang, Huawei Shen, and Xueqi Cheng. 2022. Company-as-Tribe: Company Financial Risk Assessment on Tribe-Style Graph with Hierarchical Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2712–2720.

[2] Wendong Bi, Bingbing Xu, Xiaoqian Sun, Li Xu, Huawei Shen, and Xueqi Cheng. 2023. Predicting the Silent Majority on Graphs: Knowledge Transferable Graph Neural Network. *arXiv preprint arXiv:2302.00873* (2023).

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Proc. of NeurIPS* (2013).

[4] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. 2019. Relational Graph Attention Networks. (2019).

[5] Jiaao Chen, Jianshu Chen, and Zhou Yu. 2019. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6244–6251.

[6] Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. 2021. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6271–6278.

[7] Wei-Yu Chiu, Gary G Yen, and Teng-Kuei Juan. 2016. Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 20, 6 (2016), 972–985.

[8] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proc. of EMNLP*.

[9] Alberto Garca-Duran, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *EMNLP*.

[10] Alberto Garcia-Duran, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *EMNLP*.

[11] Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *EMNLP*.

[12] Yuxia Geng, Jiaoyan Chen, Wen Zhang, Jeff Z Pan, Mingyang Chen, Huajun Chen, and Song Jiang. 2022. Relational Message Passing for Fully Inductive Knowledge Graph Completion. *arXiv preprint arXiv:2210.03994* (2022).

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[14] John Clifford Gower. 1985. Properties of Euclidean and non-Euclidean distance matrices. *Linear algebra and its applications* 67 (1985), 81–97.

[15] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *Proc. of ICLR*.

[16] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 8352–8364.

[17] Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020. Graph hawkes neural network for forecasting on temporal knowledge graphs. *arXiv preprint arXiv:2003.13432* (2020).

[18] Yongquan He, Peng Zhang, Luchen Liu, Qi Liang, Wenyuan Zhang, and Chuang Zhang. 2021. HIP Network: Historical Information Passing Network for Extrapolation Reasoning on Temporal Knowledge Graph.. In *Proc. of IJCAI*.

[19] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2019. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530* (2019).

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

[21] Timothee Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2019. Tensor Decompositions for Temporal Knowledge Base Completion. In *Proc. of ICLR*.

[22] Liang Li, Siwei Wang, Xinwang Liu, En Zhu, Li Shen, Kenli Li, and Keqin Li. 2022. Local Sample-Weighted Multiple Kernel Clustering With Consensus Discriminative Graph. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–14. https://doi.org/10.1109/TNNLS.2022.3184970

[23] Qian Li, Shu Guo, Yangyifei Luo, Cheng Ji, Lihong Wang, Jiawei Sheng, and Jianxin Li. 2023. Attribute-Consistent Knowledge Graph Representation Learning for Multi-Modal Entity Alignment. *arXiv preprint arXiv:2304.01563* (2023).

[24] Xingwang Li, Yijia Zhang, Jian Wang, Mingyu Lu, and Hongfei Lin. 2022. Knowledge-Enhanced Dual Graph Neural Network for Robust Medicine Recommendation. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 477–482.

[25] Yujia Li, Shiliang Sun, and Jing Zhao. 2022. TiRGN Time-Guided Recurrent Graph Network with Local-Global Historical Patterns for Temporal Knowledge Graph Reasoning. *ijcai22* (2022).

[26] Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022. Complex Evolutional Pattern Learning for Temporal Knowledge Graph Reasoning. In *Proc. of ACL*.

[27] 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 *ACL*.

[28] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 408–417.

[29] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proc. of SIGIR*.

[30] Ke Liang, Yue Liu, Sihang Zhou, Xinwang Liu, and Wenxuan Tu. 2022. Relational Symmetry based Knowledge Graph Contrastive Learning.

[31] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over Different Types of Knowledge Graphs: Static, Temporal and Multi-Modal. *arXiv preprint arXiv:2212.05767* (2022).

[32] Ke Liang, Sifan Wu, and Jiayi Gu. 2021. MKA: A Scalable Medical Knowledge-Assisted Mechanism for Generative Models on Medical Conversation Tasks. *Computational and Mathematical Methods in Medicine* 2021.

[33] Meng Liu and Yong Liu. 2021. Inductive representation learning in temporal networks via mining neighborhood and community influences. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2202–2206.

[34] Meng Liu, Jiaming Wu, and Yong Liu. 2022. Embedding global and local influences for dynamic graphs. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 4249–4253.

[35] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. 2022. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7603–7611.

[36] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2022. Hard Sample Aware Network for Contrastive Deep Graph Clustering. *arXiv preprint arXiv:2212.08665* (2022).

[37] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. *arXiv preprint arXiv:1805.07591* (2018).

[38] 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 *Proc. of WSDM*.

[39] Ali Sadeghian, Miguel Rodriguez, Daisy Zhe Wang, and Anthony Colas. 2016. Temporal reasoning over event knowledge graphs. In *Workshop on Knowledge Base Construction, Reasoning and Mining*.

[40] 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*. Springer, 593–607.

[41] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International conference on neural information processing*. Springer, 362–373.

[42] 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*, Vol. 33. 3060–3067.

[43] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting. In *Proc. of EMNLP*.

[44] Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proc. of ICML*.

[45] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*. PMLR, 3462–3471.

[46] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*.

[47] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. 2021. Deep Fusion Clustering Network. In *Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence*. 9978–9987.

[48] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 417–426.

[49] Lei Wang, Ee-Peng Lim, Zhiwei Liu, and Tianxiang Zhao. 2022. Explanation guided contrastive learning for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2017–2027.

[50] Jiapeng Wu, Yishi Xu, Yingxue Zhang, Chen Ma, Mark Coates, and Jackie Chi Kit Cheung. 2021. Tie: A framework for embedding-based incremental temporal knowledge graph completion. In *Proc. of SIGIR*.

[51] Chengjin Xu, Yung-Yu Chen, Mojtaba Nayyeri, and Jens Lehmann. 2021. Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings. In *Proc. of NAACL*.

[52] Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, and Jens Lehmann. 2020. Temporal knowledge graph completion based on time series gaussian embedding. In *Proc. of ISWC*.

[53] Fuwei Zhang, Zhao Zhang, Xiang Ao, Fuzhen Zhuang, Yongjun Xu, and Qing He. 2022. Along the Time: Timeline-traced Embedding for Temporal Knowledge Graph Completion. In *Proc. of CIKM*.

[54] Mingda Zhang, Keren Ye, Rebecca Hwa, and Adriana Kovashka. 2020. Story completion with explicit modeling of commonsense knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 376–377.

[55] Yingying Zhang, Xian Wu, Quan Fang, Shengsheng Qian, and Changsheng Xu. 2023. Knowledge-enhanced attributed multi-task learning for medicine recommendation. *ACM Transactions on Information Systems* 41, 1 (2023), 1–24.

[56] Shan Zhao, Minghao Hu, Zhiping Cai, and Fang Liu. 2020. Modeling Dense Cross-Modal Interactions for Joint Entity-Relation Extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 4032–4038.

[57] 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*, Vol. 35. 4732–4740.