Joint Embedding of Local Structures and Evolutionary Patterns for Temporal Link Prediction

Tingxuan Chen[1], Jun Long[2](✉), Liu Yang[1](✉), Guohui Li[1], Shuai Luo[1], and Meihong Xiao[1]

[1] School of Computer Science and Engineering, Central South University, Changsha, China
{chentingxuan,yangliu,guohuili,luoshuai06,xiaomeihong,junlong}@csu.edu.cn
[2] Big Data Institute, Central South University, Changsha, China

Abstract. Link prediction tackles the prediction of missing facts in an incomplete knowledge graph (KG) and has been widely explored in reasoning and information retrieval. The vast majority of existing methods perform link prediction on static KGs, with the assumption that the relational facts are generally correct. However, some facts may not be universally valid, as they tend to evolve. Despite the prevalence of temporal knowledge graphs (TKGs) with evolving facts, the studies on such data for temporal link prediction are still far from resolved. In this paper, we propose SiepNet, a novel graph neural network for temporal link prediction, driven by local **S**tructural **I**nformation and **E**volutionary **P**atterns. Specifically, SiepNet captures the local structural information based on a relation-aware GNN architecture, and incorporates temporal attention to model long- and short-range historical dependencies hidden in TKGs. Moreover, SiepNet integrates local structures and evolutionary patterns to enhance the semantic representation of evolving facts in TKGs. The extensive experiments on five real-world TKG datasets demonstrate the effectiveness of our approach SiepNet in temporal link prediction, compared with the state-of-the-art methods.

Keywords: Temporal knowledge graph · Graph embedding · Temporal link prediction · Representation learn · Evolutionary patterns

1 Introduction

Knowledge graphs (KGs) organize and store real-world facts, enabling multifarious downstream applications, such as knowledge retrieval, question answering, and recommender systems [12]. KGs encode factual knowledge in the form of triple (s, r, o) as directed graphs, where nodes correspond to the subject entity s or object entity o, and edges represent the relation r among them. Owing to the high cost of knowledge fusion and dynamics of facts, most KGs often suffer from incompleteness [31]. Thus, link prediction becomes a crucial task, which intends

X. Yang et al. (Eds.): ADMA 2023, LNAI 14177, pp. 107–121, 2023.
https://doi.org/10.1007/978-3-031-46664-9_8

to recover the most probable missing facts. Since real-world KGs contain millions of multi-relational facts, traditional symbolic and logic-based approaches cannot be extended to large-scale KGs for link prediction.

Recently, KG embedding has emerged as a promising method for link prediction. It attempts to learn multi-dimensional vectorial representations of entities and relations in KGs, while using a scoring function to evaluate the plausibility of a triplet. Represented by TransE [1], these translation-based approaches achieve a good trade-off between model complexity and link prediction performance by modelling relations as translation operations on entity embeddings. However, the vast majority of existing embedding methods perform link prediction on static KGs, with the assumption that the relational facts in KGs are generally correct.

Actually, facts always evolve over a specific period of time [3]. Therefore, researchers construct temporal knowledge graphs (TKGs) to store ever-growing temporal information either explicitly or implicitly, such as YAGO [24] and ICEWS [16]. Figure 1 shows an example of a temporal knowledge graph (TKG), where the fact (*Donald Trump*, *president of*, *USA*) was accurate only from 2017 to 2020. However, traditional KG embedding methods cannot address the issue of TKGs, where facts often show temporal dynamics. For example, they often confuse entities such as *Trump* and *Biden* when predicting (?, *president of*, *USA*, *2021*). Additionally, TKG embeddings carrying temporal information are challenging due to the sparsity and irregularities of temporal expressions [5].

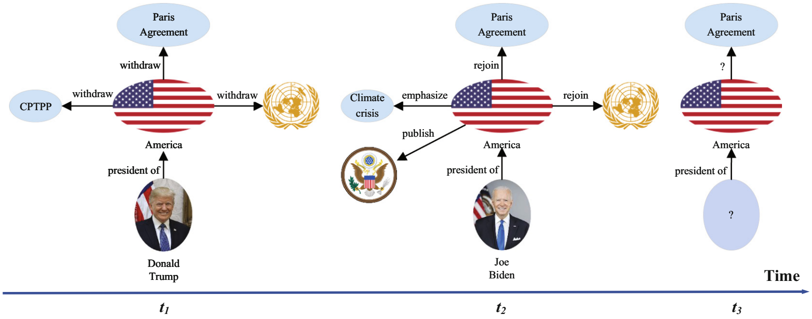


Fig. 1. Example of temporal knowledge subgraphs.

To solve the challenges, Know-Evolve [27] and its extension DyRep [28] predict future events based on ground truths of preceding events at inference time. As a result, these methods cannot predict missing events in future time-stamps without ground truths. To capture more information based on past facts, Jin proposed a novel autoregressive architecture RE-NET [14], which models facts as probability distributions over TKGs. However, RE-NET learns representations of entities and relations by implicitly exploiting temporal information without distinguishing dynamic dependencies across facts.

In this work, we observe that TKGs are dynamically heterogeneous graphs with multiple relationships, i.e., the local structures of graphs are always diverse under different time windows, and the facts evolve across time windows. As an example in Fig. 1, the local structure information of the entity *America* comes from 4 entities and 2 relations at t_1. While at t_2, the local structure of the entity *America* changes significantly, resulting in not only the emergence of new entities and relations but also the absence of some entities and relations at t_1. Moreover, the fact (*Donald Trump*, *president of*, *America*) at t_1 evolves into (*Joe Biden*, *president of*, *America*) at t_2.

To this end, we propose SiepNet, a novel graph neural network for temporal link prediction, driven by local **S**tructural **I**nformation and **E**volutionary **P**atterns. The main ideas of SiepNet are (1) capturing graph structure dependencies based on a relation-aware GNN architecture, (2) learning long-range and short-range evolutionary patterns of TKGs using an attention-based recurrent network, and (3) integrating local structures and evolutionary patterns to strengthen the representation learning of facts, which improves the performance of temporal link prediction. We summarize our main contributions as follows:

- We propose a representation learning model SiepNet for temporal link prediction, which simultaneously considers local structures and evolutionary patterns hidden in TKGs.
- We design an attention-based recurrent network to tackle dynamic dependencies across entities over time, which helps to distinguish the impact of different historical facts on future facts inference.
- To validate the effectiveness of our model, we conduct extensive experiments on five real-world TKGs containing millions of multi-relational facts with different time intervals, where our model consistently outperforms other baselines in terms of temporal link prediction.

2 Related Work

Towards temporal link prediction, we restrict our focus to recent works on TKG embedding methods, including geometric models and neural network models.

Geometric Models. These models attempt to minimize the distance between two entity vectors translated by geometric transformations of relations. TTransE [17] extends TransE [1] for static KGs to TKGs by adding temporal constraints. TA-TransE [5] embeds temporal information into relation types, which can be used with existing scoring functions for temporal link prediction in TKGs. HyTE [3] utilizes time-specific normal vectors directly to generate representations of entities and relations over different time-stamps. Nevertheless, these geometric models cannot infer future facts according to past facts and cannot be further extended to extrapolate settings.

Neural Network Models. These models use deep neural networks to learn underlying features of time-stamps for link prediction. RE-NET [14] combines a recurrent neural network and a neighborhood aggregator to model event sequences. CyGNet [34] predicts future facts by modelling observed facts with a copy-generation network. TITer [25] continuously transfers query nodes to new nodes through relevant temporal facts based on time-aware reinforcement learning strategies, and generates representation vectors of unseen entities using an IM module. CluSTeR [19] performs temporal reasoning on TKGs by joint reinforcement learning and a graph convolution network. RE-GCN [20] learns evolutionary representations of facts at each timestamp, by modelling KG sequences recurrently using a recurrent evolutionary network. However, the performance of these neural network models is limited by repetitive patterns.

3 Problem Definition

We consider a temporal knowledge graph as a sequence of graph snapshots, ordered ascending based on time-stamps, namely $G = \{G_1, G_2, \cdots, G_\tau\}$, where $G_t = (V_t, E_t)$ represents the snapshot at a particular time slice t ($t \in 1,\ 2, \cdots, \tau$) with an entity set V_t and a relation set E_t. V_t corresponds to the subject entity s or object entity o at a time slice t, and E_t represents the relation r between them. Thus, a fact in G_t is denoted by a quadruple (s, r, o, t) with a time slice t, in which $s \in V_t$, $o \in V_t$ and $r \in E_t$.

Given the preceding observed facts in G, the temporal link prediction aims to predict the missing facts of the current time slice t, i.e., to predict the unseen subject entity s given $(?, r, o, t)$ (object entity o given $(s, r, ?, t)$, and relation r given $(s, ?, o, t)$) at a particular time slice t.

4 Methodology

4.1 The Model Architecture

The proposed model SiepNet depicted in Fig. 2 consists of two main components: (1) Local Structural Information Aggregation, and (2) Evolutionary Patterns Aggregation. First of all, we design a relation-aware GNN to capture the local structural information from multi-relational and multi-hop neighbors of each single graph snapshot. Then, we explore long-range and short-range evolutionary patterns of TKGs using an attention-based recurrent network. In addition, we integrate local structures and evolutionary patterns to strengthen the representation learning of facts, which in turn improves the performance of temporal link prediction.

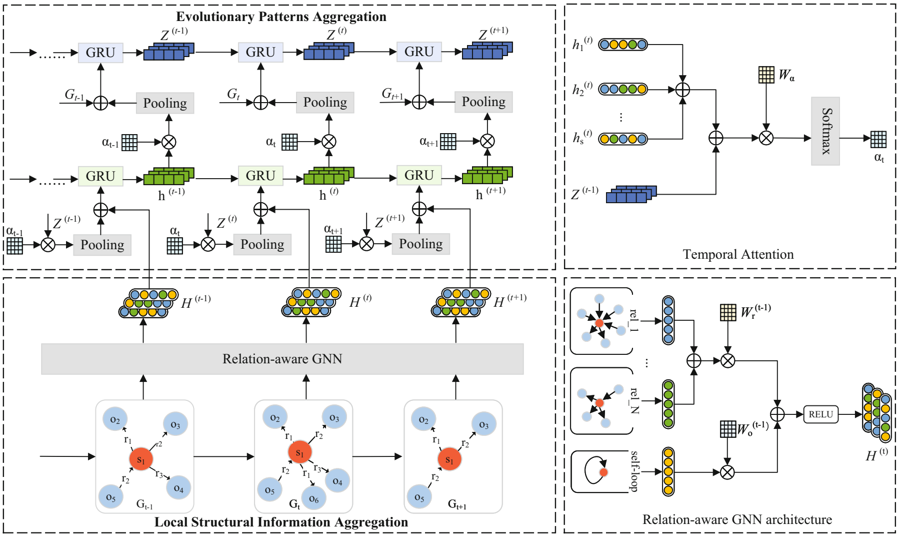

Evolutionary Patterns Aggregation
GRU
Pooling
Pooling
Relation-aware GNN
Local Structural Information Aggregation
Temporal Attention
Softmax
RELU
Relation-aware GNN architecture
rel_1
rel_N
self-loop

Fig. 2. The architecture of the SiepNet temporal link prediction model.

4.2 Local Structural Information

To aggregate local structural information from multi-relational and multi-hop neighbors in each graph snapshot G_t, SiepNet seeks to make two linked nodes share similar representations. To achieve this, we let each node representation $h_o^{(t)}$ in G_t aggregates neighbors and past messages, and then calculate its new representation. Initially, $h_o^{(0)}$ is set to trainable embedding vector for each node. SiepNet calculates the forward-pass update of an entity denoted by v_o in a multi-relational graph, based on the following message-passing neural network:

$$h_o^{(t)} = \sigma(\sum_{s \in N_{o,r}^t} \mathcal{F}_{str}(h_s^{(t-1)}, r^{(t-1)}) + W_o^{(t-1)} h_o^{(t-1)}) \tag{1}$$

where $h_o^{(t)}$ is the intermediate representation of node v_o at time slice t, combining local structural messages $h_s^{(t-1)}$ from all neighbors $N_{o,r}^t$ under relation $r \in E_t$ and its past messages $h_o^{(t-1)}$. $W_o^{(t-1)}$ is a learnable parameter, indicating the past weight. To comprehensively aggregate the local structural messages of node v_o, we implement the message function $\mathcal{F}_{str}(.,.)$ by

$$\mathcal{F}_{str}(h_s^{(t-1)}, r^{(t-1)}) = \frac{1}{c_{o,r}^t} W_r^{(t-1)} [h_s^{(t-1)} \times r^{(t-1)}] + b_{str} \tag{2}$$

where $h_s^{(t-1)} \times r^{(t-1)}$ is the local structural messages, while $W_r^{(t-1)}$ and b_{str} are the learnable parameters, indicating the local weight and bias. $c_{s,r}$ is a normalizing factor that can either be learned or chosen in advance (e.g., $c_{o,r}^t = |N_{o,r}^t|$).

Unlike traditional GCNs, SiepNet accumulates and encodes features of entities from local structural neighborhoods, i.e., $\frac{1}{c_{s,r}} W_r^{(t-1)}[h_s^{(t-1)} \times r^{(t-1)}]$. Intuitively, relations with different types and directions can derive various local graph structures between entities. Therefore, SiepNet accumulates the overall features of each entity by relation-specific transformations, i.e., $\sum_{s \in N_{o,r}^t} \mathcal{F}_{str}(h_s^{(t-1)}, r^{(t-1)})$. To calculate the past messages of an entity, SiepNet introduces a single self-connection to each node, i.e., $W_o^{(t-1)} h_o^{(t-1)}$. Finally, SiepNet combines both the overall features and information from past steps, and outputs a sequence of representations notated as $\left\{H^{(1)}, \cdots, H^{(t)}\right\}$, where $H^{(t)} = \left\{h_1^{(t)}, \cdots, h_n^{(t)}\right\}$ denotes the representations of entities in each single graph snapshot G_t.

4.3 Evolutionary Patterns

Besides aggregating local structural information, previous facts also influence current representations. Moreover, facts are always evolving over adjacent time windows, further changing the local structural information of the current graph snapshot. Intuitively, we should capture these two evolutionary patterns, i.e., long-range historical dependence and short-range structural dependence. To achieve this, we design an attention-based recurrent block in SiepNet to capture evolutionary patterns in TKGs. Formally, SiepNet combines the local structural representation $h_o^{(t)}$ and the historical representation $(\mathrm{h}_o^{(t-1)}, Z^{(t-1)})$:

$$\mathrm{h}_o^{(t)}, Z^{(t)} := \mathcal{F}_{evo}(h_o^{(t)}, \mathrm{h}_o^{(t-1)}, Z^{(t-1)}) \tag{3}$$

where $\mathcal{F}_{evo}$ is a recurrent operator, which allows SiepNet to learn long-range dependencies of sequence data and explore the evolving patterns of temporal knowledge graphs to update current representations. When there are few structural dependencies from neighbor nodes (i.e., $h_o^{(t)} \longrightarrow 0$), current representations $(Z^{(t)}, \mathrm{h}_o^{(t)})$ will be greatly influenced by long- and short-range historical dependencies $(Z^{(t-1)}, \mathrm{h}_o^{(t)})$. Otherwise, local structural dependences $h_o^{(t)}$ will have a greater impact on current representations.

Most existing works use simple recurrent neural networks to implement $\mathcal{F}_{evo}$ in message propagation, e.g., RE-NET [14] uses GRU [2], EvoNet [11] uses LSTM [10], etc. For historical snapshot propagation, these methods only summarize the current representations of nodes, i.e., $Z^{(t)} = \sum_{o \in V_t} \mathrm{h}_o^{(t)}$, ignoring dynamic interactions of nodes across time windows. However, both long-range historical dependence and short-range dynamic dependence present different temporal information, influencing the evolution of facts. To improve the ability of temporal link prediction, $\mathcal{F}_{evo}$ should consider historically long-range and short-range dependence of previous facts $G_{1:t}$ when modelling snapshot propagation, and thus influence current representations through local dynamic dependence of node interactions. Specifically, $\mathcal{F}_{evo}$ can be implemented by

$$\mathcal{F}_{evo}(h_o^{(t)}, \mathrm{h}_o^{(t-1)}, Z^{(t-1)}) = \begin{cases} Z^{(t)} = \mathrm{RNN}\left(Z^{(t-1)}, G_t \oplus g(\alpha_t \sum_{o \in V_t} \mathrm{h}_o^{(t)})\right) \\ \\ \mathrm{h}_o^{(t)} = \mathrm{RNN}\left((1-\alpha_t)\mathrm{h}_o^{(t-1)}, h_o^{(t)} \oplus g(\alpha_t Z^{(t-1)})\right) \end{cases} \tag{4}$$

where $\oplus$ denotes the concatenation operator and $g(*)$ is an element-wise max-pooling operator. We use a recurrent model RNN to update current representations $\mathrm{h}_o^{(t)}$ based on historical representation $(\mathrm{h}_o^{(t-1)}, Z^{(t-1)})$ and current local structural representation $h_o^{(t)}$, and capture evolutionary patterns $Z^{(t)}$ based on long-range and short-range dependencies $(Z^{(t-1)}, \mathrm{h}_o^{(t)})$ as well as current facts G_t.

Typically, the impact of long-range historical dependence and short-range structural dependence on current representations varies over time. Accordingly, we design the following temporal attention mechanism as follows to capture temporal information in node interactions, which in turn helps to model the long-range and short-range evolutionary patterns of facts.

$$\alpha_t = \mathrm{softmax}(W_\alpha(Z^{(t-1)} \oplus \sum_{s \in N_{o,r}^t} h_s^{(t)})) \tag{5}$$

where W_α is a independent parameter matrix, updated automatically by back-propagation. The attention score α_t re-weights the two evolutionary patterns, which is calculated based on long-range evolutionary dependencies and short-range structural dependencies.

The recurrent model RNN aims at smoothing two input vectors at each time step, which can be implemented using many existing methods. Here, we utilize GRU to update $\mathrm{h}_o^{(t)}$ as an example.

$$\mathrm{h}_o^{(t)} : \begin{cases} a^{(t)} = h_o^{(t)} \oplus g(\alpha_t Z^{(t-1)}) \\ i^{(t)} = \sigma(W_i a^{(t)} + U_z(1-\alpha_t)\mathrm{h}_o^{(t-1)}) \\ r^{(t)} = \sigma(W_r a^{(t)} + U_r(1-\alpha_t)\mathrm{h}_o^{(t-1)}) \\ \mathrm{h}_o^{(t)} = (1-i^{(t)}) \circ (1-\alpha_t)\mathrm{h}_o^{(t-1)} + i^{(t)} \circ \tanh(W_h a^{(t)} + U_h(r^{(t)} \circ \mathrm{h}_o^{(t-1)})) \end{cases} \tag{6}$$

where $i^{(t)}$ and $r^{(t)}$ are update gate and reset gate respectively, while $\circ$ is a Hadamard operator. The current node representations are updated by receiving their currently local structure dependencies and historical evolution dependencies, with a temporal attention score regulating the weight of long-range and short-range dependencies.

Consequently, both the representations $\mathrm{h}_o^{(t)}$ and $Z^{(t)}$ capture the evolutionary patterns and local structural dependencies up to the t-th time step, which in turn can be used to predict the facts G_{t+1} at the next time step. Then, we encode the current graph snapshot G_t as representation $\mathbf{H}_G^{(t)}$ with a fully connected layer, which can be formulated as

$$\mathbf{H}_G^{(t)} = \mathrm{FCL}_n(Z^{(t)} \oplus \sum_{o \in V_t} \mathrm{h}_o^{(t)}; \theta_n) \tag{7}$$

where the input are the concatenated features of all $\mathrm{h}_o^{(t)}$ and $Z^{(t)}$, while θ_n denotes the parameters of FCL_n. Then we use a classifier to estimate the probability of the next graph snapshot $\mathbf{P}(G_{t+1} \mid \mathbf{H}_G^{(t)})$.

4.4 Model Optimization

As the topology of TKGs changes over time, SiepNet model should continuously update its parameters to accommodate the evolutionary patterns of TKGs. Furthermore, note that the snapshots closer to the next time slice $(t+1)$ have more similar characteristics than those farther from the ground truth. Hence, we introduce the first l graph snapshots $G_{t-l+1}^{t+1} = \{G_{t-l+1},\ G_{t-l+2}, \cdots,\ G_{t+1}\}$ as the input, which is close to the next time slice $(t+1)$, based on minimizing the cross-entropy loss $\mathcal{L}$ for training.

$$\mathcal{L} = -\sum_{\tau=(t-l)}^{t} \hat{G}_{\tau+1} \log \mathbf{P}(G_{\tau+1} \mid \mathbf{H}_G^{(\tau)}) + (1-\hat{G}_{\tau+1}) \log(1-\mathbf{P}(G_{\tau+1} \mid \mathbf{H}_G^{(\tau)})) \quad (8)$$

where $\hat{G}_{\tau+1} \in \mathbb{R}^{|G_{\tau+1}|}$ is the label set of ground truths with elements of 1 if the fact occurs and 0 otherwise. SiepNet can fully aggregate the latest temporal information of the dynamic network, according to the sequence of previous snapshots G_{t-l+1}^{t+1}, which is considered as the most similar characteristics to the actual snapshots of G_{t+1}.

As in previous work on regularization, we employ dropout [9] to alleviate overfitting while capturing local structural information and evolutionary patterns.

5 Experiments

5.1 Experimental Setup

Datasets. In our experiments, we used five widely use TKG datasets, including three event-based TKGs (i.e., GDELT [18], ICEWS14 [27], and ICEWS18 [29]) and two public TKGs (i.e., WIKI [17] and YAGO [24]) specifically.

Evaluation Setting and Metrics. Following the prior work [34], we split each dataset except ICEWS14 into a training set, a validation set, and a test set at a ratio of 80%/10%/10%, respectively. For dataset ICEWS14, we directly utilize the splitting provided in [27]. We report a widely used filtered settings [8,14,34] of Mean Reciprocal Rank (MRR) and Hits at K (Hits@K), which are standard evaluation metrics for link prediction.

Baselines. We compare our proposed model SiepNet with a variety of static KG models and TKG models. Static KG models include DistMult [32], R-GCN [23], ConvE [4] and RotatE [26]. TKG models include TTransE [13], TA-DistMult [5], TA-TransE [5], HyTE [3], RE-NET [14], TeMP [30], RE-GCN [20], xERTE [6], TANGO-TuckER [7], TANGO-Distmult [7], CyGNet [34], EvoKG [22] and TLogic [21].

Model Configurations. Initially, we set the length of the history l to 10, which means that SiepNet saves the sequence of 10 previous snapshots. The dropout rate is set to 0.5, and the embedding size is set to 200 to match the baseline methods set in [34]. The model parameters are optimized using Adam optimizer [15] with a learning rate of 0.001. The training epoch is set to 20, which is sufficient for convergence in most cases. All experiments are conducted on GeForce GTX 3080 Ti. The baseline results are also adopted from [33].

5.2 Performance Evaluation

Overall Performance. Table 1 and Table 2 show the temporal link prediction performance of SiepNet and baselines on five real-world TKGs, where the best results are shown in **bold**. We use "–" instead of experimental results that are not run out within a day. Remarkably, SiepNet consistently outperforms the baselines in most cases, which convincingly validates its effectiveness.

Table 1. Performance (in percentage) for temporal link prediction on YAGO and WIKI datasets under the filtered settings

Method	YAGO			WIKI		
	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
DisMult [2015]	59.47	52.97	60.91	46.12	37.24	49.81
R-GCN [2018]	41.30	32.56	44.44	37.57	28.15	39.66
ConvE [2018]	62.32	56.19	63.97	47.57	38.76	50.10
RotatE [2018]	65.09	62.21	65.67	50.67	48.17	50.71
TTransE [2016]	32.57	27.94	43.39	31.74	22.57	36.25
TA-DisMult [2018]	61.72	50.57	65.32	48.09	45.97	49.51
TA-TransE [2018]	56.61	46.76	65.95	24.24	1.74	47.18
HyTE [2018]	23.16	10.78	45.74	43.02	28.81	45.74
RE-NET [2020]	65.16	63.29	65.63	51.97	48.01	52.07
TeMP [2020]	62.25	55.39	64.63	49.61	46.96	50.24
RE-GCN [2021]	65.29	59.98	68.70	44.86	39.82	46.75
xERTE [2021]	58.75	58.46	58.85	–	–	–
TANGO-TuckER [2021]	67.21	65.56	67.59	53.28	52.21	53.61
TANGO-Distmult [2021]	68.34	67.05	68.39	54.05	51.52	53.84
CyGNet [2021]	63.47	64.26	65.71	45.50	50.48	50.79
EvoKG [2022]	55.11	54.37	**81.38**	50.66	12.21	**63.84**
TLogic [2022]	1.29	0.49	0.85	51.07	50.13	51.18
SiepNet (ours)	**73.77**	**71.65**	74.65	**54.46**	**52.35**	62.73

Specifically, static KG methods usually show promising results, but lag behind the best-performing TKG method SiepNet to a large extent, as they

Table 2. Performance (in percentage) for temporal link prediction on ICEWS14, ICEWS18 and GDELT datasets under the filtered settings

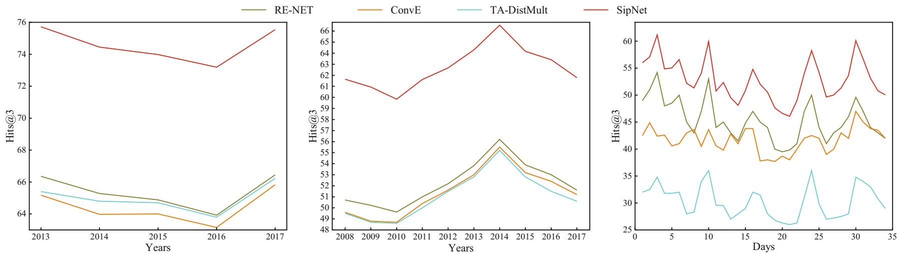
Method	ICEWS14			ICEWS18			GDELT		
	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
DisMult	19.06	10.09	22.00	22.16	12.13	26.00	18.71	11.59	20.05
R-GCN	26.31	18.23	30.43	23.19	16.36	25.34	23.31	17.24	24.96
ConvE	40.73	33.20	43.92	36.67	28.51	39.80	35.99	27.05	39.32
RotatE	29.56	22.14	32.92	23.10	14.33	27.61	22.33	16.68	23.89
TTransE	6.35	1.23	5.80	8.36	1.94	8.71	5.52	0.47	5.01
TA-DistMult	20.78	13.43	22.80	28.53	20.30	34.57	29.35	22.11	34.56
TA-TransE	15.99	0.00	26.39	17.69	0.01	30.14	19.18	0.00	33.20
HyTE	11.48	5.64	13.04	7.31	3.10	7.50	6.37	0.00	6.72
RE-NET	45.71	38.42	49.06	42.93	36.19	45.47	40.12	32.43	43.40
TeMP	43.13	35.67	45.79	40.48	33.97	42.63	37.56	29.82	40.15
RE-GCN	32.37	24.43	35.05	32.78	24.99	35.54	29.46	21.74	32.01
xERTE	32.92	26.44	36.58	36.95	30.71	40.38	–	–	–
TANGO-TuckER	46.42	38.94	50.25	44.56	37.87	47.46	38.00	28.02	43.91
TANGO-Distmult	46.68	41.20	48.64	44.00	38.64	45.78	41.16	35.11	43.02
CyGNet	48.63	41.77	52.50	46.69	40.58	49.82	50.29	44.53	**54.69**
EvoKG	18.30	6.30	19.43	29.67	12.92	33.08	11.29	2.93	10.84
TLogic	38.19	32.23	41.05	37.52	30.09	40.87	22.73	17.65	24.66
SiepNet	**49.97**	**42.65**	**53.28**	**47.93**	**43.41**	**52.36**	**50.79**	**45.10**	53.11

cannot capture the sequential patterns across time-stamps. Surprisingly, almost static KG methods normally perform better than two TKG methods (i.e., TTransE and HyTE) on five TKG datasets. It owes to the fact that TTransE and HyTE learn representations for each snapshot independently, instead of capturing long-range historical dependencies. Besides, the experimental results of TA-DistMult and DistMult validate the effectiveness of incorporating temporal information for temporal link prediction, where TA-DistMult is a temporal-aware version of static KG method DistMult.

In addition, SiepNet drastically outperforms other TKG methods, although they all consider dynamic features of facts. Especially on YAGO dataset with the most facts, SiepNet leads to improvements of 2.70% in MRR, 6.97% in Hits@1, and 5.10% in Hits@3 compared with the best baseline. We believe this is due to that SiepNet considers dynamic long-range and short-range historical dependencies using temporal attention, while other TKG models ignore the evolutionary patterns. The excellent performance of SiepNet and RE-NET validate the importance of long-range dependencies for link prediction. Although our performance in Hits@3 of YAGO, WIKI, and GDELT dataset are not the best, the remarkable performances in Hits@1 and MRR prove that our algorithm SiepNet is able to predict future facts more accurately. The main reason is that there is a large number of repetitive facts in these datasets. Thus, CyGNet and EvoKG perform well on Hits@3, but they cannot predict more accurate facts, resulting in Hits@1 much lower than ours. TeMP is designed to handle knowledge graph complementation tasks (graph interpolation) rather than predicting future events, so it does not perform as well as extrapolation models. Although xERTE supports a certain degree of predictive interpretation capability, it cannot efficiently handle large-scale datasets, such as GDELT and WIKI.

Note that static KG model and TKG model perform similarly well on YAGO and WIKI, but poorly on ICEWS14, ICEWS18 and GDELT. As discussed in [22], the time intervals of YAGO and WIKI datasets are much larger than other datasets. Therefore, each time-stamp in YAGO and WIKI has more local structural information than the other three datasets. Besides, ICEWS14 and ICEWS18 are extracted from the Integrated Crisis Early Warning System (ICEWS), which records many recurring political events with time stamps. Accordingly, only modelling repetitive patterns or 1-hop neighbors will lose a significant amount of evolutionary patterns and structural information. The experimental results show that SiepNet is able to better model these datasets, which contain complex dynamic dependencies over concurrent facts.

Performance over Time. To further evaluate the performance of SiepNet over time, we compared the performance in percentage of different timestamps, using filtered Hits@3 on YAGO, WIKI, and ICEWS18. As shown in Fig. 3, SiepNet consistently outperforms baselines over different timestamps. The performance of each method varies with the entities in the test set at each timestamp. In addition, the difference between our TKG model SiepNet and static KG model ConvE evolves slowly as time goes by, as shown in Fig. 3. We believe that further facts in the future are even harder to predict.



(a) YAGO (b) WIKI (c) ICEWS18

Fig. 3. Performance over specific timestamps with filtered Hits@3.

Specifically, each method shows a significant performance improvement at a particular timestamp in the future. We believe this is because facts from the past tend to reappear at the future timestamps. As shown in Fig. 3(a), all methods perform poorly in 2016, but in 2017 surpass their performance in 2013.

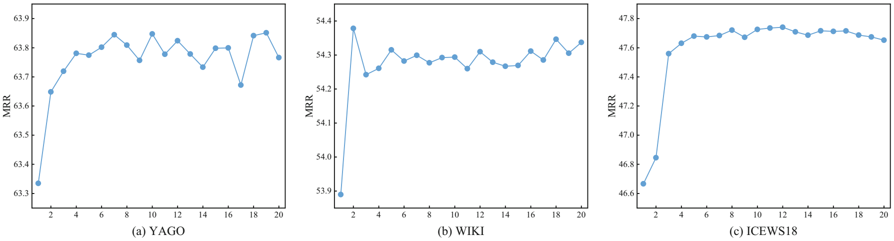
5.3 Ablation Study

To eliminate the effect of different model components of SiepNet, we create variants of SiepNet by adapting the use of model components and report the performances (in percentage) on YAGO dataset.

Table 3. Ablation study for temporal link prediction

Method	YAGO			
	MRR	Hits@1	Hits@3	Hits@10
SiepNet w. R	66.25	64.92	66.53	68.54
SiepNet w. B	73.53	71.66	74.66	76.92
SiepNet w/o TA	64.30	62.41	64.86	67.52
SiepNet	73.77	71.65	74.65	77.24

Evolutionary Patterns. To demonstrate how evolutionary patterns affect the final results of SiepNet, we conduct experiments using l random past graph snapshots rather than l snapshots closest to the current graph snapshot. The results denoted as SiepNet w. R are presented in Table 3. Obviously, SiepNet w. R hurts model quality, suggesting that modelling the snapshots closer to the current time slice can improve performance.



(a) YAGO

(b) WIKI

(c) ICEWS18

Fig. 4. Performance over different lengths of time slice with filtered MRR.

As described in Sect. 4.5, graph snapshots of adjacent time slices tend to have more similar characteristics. Thus, the length of previous time slice l affects the performance of our proposed model SiepNet. Figure 4 shows the performance of SiepNet on YAGO, WIKI and ICEWS18 datasets, with different lengths of time slices l for temporal link prediction. As the length of time slices increases, SiepNet performs better on MRR. Nevertheless, MRR tends to be stable when the length of time slices is over 6. As a result, longer time slices introduce more noise and lead to performance fluctuations of SiepNet.

Evolutionary Directions. SiepNet w. B in Table 3 indicates the variant of SiepNet using Bi-GRU instead of GRU to explore evolving patterns of TKGs. The experimental results of SiepNet w. B and SiepNet are similarly well on YAGO, as compared with other variants of SiepNet. Therefore, combining forward and backward snapshot information has less significant impacts on the performance of SiepNet and more computational overhead.

Temporal Attention. The results denoted as SiepNet w/o TA in Table 3 demonstrate the performance of SiepNet without temporal attention component. It can be seen that SiepNet w/o TA performs noticeably worse than SiepNet on YAGO datasets, which justifies the necessity of temporal attention component to model long-range and short-range dependencies.

6 Conclusion

In this paper, we propose a novel temporal link prediction model SiepNet, which adapts to the evolutionary process of dynamic facts by modelling temporal adjacency facts with associated semantic and informational patterns. Specifically, SiepNet explores the local structural information based on a relation-aware GNN architecture. In addition, SiepNet incorporates temporal attention to help with modelling long-range and short-range historical dependencies hidden in TKGs. The experimental results on seventeen baselines demonstrate the significant advantages and promising performance of SiepNet in temporal link prediction. In future work, we will explore the persistence modelling of facts, rather than just predicting missing facts at a certain time slice t.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No. U2003208 and No. 62172451), the Scientific and Technological Innovation 2030-Major software of New Generation Artificial Intelligence (No. 2020AAA0109601), and the Open Research software of Zhejiang Lab (No. 2022KG0AB01).

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), Lake Tahoe, America, pp. 1–9. ACM (2013)
2. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Gated feedback recurrent neural networks. In: International Conference on Machine Learning, pp. 2067–2075. PMLR (2015)
3. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 2001–2011. ACL (2018)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, USA, pp. 1811–1818. AAAI (2018)
5. García-Durán, A., Dumancic, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 4816–4821. ACL (2018)

6. Han, Z., Chen, P., Ma, Y., Tresp, V.: Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: International Conference on Learning Representations (ICLR) (2021)
7. Han, Z., Ding, Z., Ma, Y., Gu, Y., Tresp, V.: 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, pp. 8352–8364 (2021)
8. He, Y., Zhang, P., Liu, L., Liang, Q., Zhang, W., Zhang, C.: HIP network: historical information passing network for extrapolation reasoning on temporal knowledge graph. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, (IJCAI), pp. 1915–1921 (2021)
9. Hinton, G., NitishSrivastava, A., Salakhutdinov, I.R.R.: Improving neural networks by preventing co-adaptation of feature detectors. Comput. Sci. **3**(4), 212–223 (2012)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Hu, W., Yang, Y., Cheng, Z., Yang, C., Ren, X.: Time-series event prediction with evolutionary state graph. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 580–588 (2021)
12. Hui, B., Zhang, L., Zhou, X., Wen, X., Nian, Y.: Personalized recommendation system based on knowledge embedding and historical behavior. Appl. Intell. **52**(1), 954–966 (2022)
13. Jiang, T., et al.: Towards time-aware knowledge graph completion. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, pp. 1715–1724. ACL (2016)
14. Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: autoregressive structure inferenceover temporal knowledge graphs. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6669–6683. ACL, Virtual (2020)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (Poster), San Diego, California, USA. Openview (2015)
16. Lautenschlager, J., Shellman, S., Ward, M.: ICEWS event aggregations. Harvard Dataverse 3 (2015)
17. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion Proceedings of the the Web Conference 2018, Lyon, France, pp. 1771–1776. ACM (2018)
18. Leetaru, K., Schrodt, P.A.: GDELT: global data on events, location, and tone, 1979–2012. In: International Studies Association, San Francisco, California, USA, pp. 1–49 (2013)
19. Li, Z., et al.: Search from history and reason for future: two-stage reasoning on temporal knowledge graphs. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4732–4743, Berkeley Hotel, Bangkok, Thailand. ACL (2021)
20. Li, Z., et al.: 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, pp. 408–417. ACM, Virtual (2021)

21. Liu, Y., Ma, Y., Hildebrandt, M., Joblin, M., Tresp, V.: TLogic: temporal logical rules for explainable link forecasting on temporal knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 4120–4127 (2022)
22. Park, N., Liu, F., Mehta, P., Cristofor, D., Faloutsos, C., Dong, Y.: EvoKG: jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 794–803 (2022)
23. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
24. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, pp. 697–706. ACM (2007)
25. Sun, H., Zhong, J., Ma, Y., Han, Z., He, K.: Timetraveler: reinforcement learning for temporal knowledge graph forecasting. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Barceló Bávaro Convention Centre, Punta Cana, Dominican Republic, pp. 8306–8319. ACL (2021)
26. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations, Vancouver, Canada. OpenReview (2018)
27. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: International Conference on Machine Learning, Sydney, Australia, pp. 3462–3471. PMLR (2017)
28. Trivedi, R., Farajtabar, M., Biswal, P., Zha, H.: DyRep: learning representations over dynamic graphs. In: International Conference on Learning Representations, New Orleans, Louisiana, USA. OpenReview (2019)
29. Ward, M.D., Beger, A., Cutler, J., Dickenson, M., Dorff, C., Radford, B.: Comparing GDELT and ICEWS event data. Analysis **21**(1), 267–297 (2013)
30. Wu, J., Cao, M., Cheung, J.C.K., Hamilton, W.L.: TeMP: temporal message passing for temporal knowledge graph completion. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5730–5746 (2020)
31. Xiao, H., Chen, Y., Shi, X.: Knowledge graph embedding based on multi-view clustering framework. IEEE Trans. Knowl. Data Eng. **33**(2), 585–596 (2019)
32. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015, San Diego, California, USA. OpenReview (2015)
33. Xu, Y., Ou, J., Xu, H., Fu, L.: Temporal knowledge graph reasoning with historical contrastive learning. CoRR (2022)
34. Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y.: Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4732–4740. AAAI, Virtual (2021)