Knowledge-Based Systems 251 (2022) 109234

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Temporal knowledge graph representation learning with local and global evolutions

Jiasheng Zhang [a], Shuang Liang [a], Yongpan Sheng [b], Jie Shao [a,c,*]

[a] University of Electronic Science and Technology of China, Chengdu 611731, China
[b] Chongqing University, Chongqing 401331, China
[c] Shenzhen Institute for Advanced Study, UESTC, Shenzhen 518110, China

ARTICLE INFO

Article history:
Received 10 February 2022
Received in revised form 6 June 2022
Accepted 7 June 2022
Available online 16 June 2022

Keywords:
Knowledge graph
Temporal evolution
Representation learning

ABSTRACT

Temporal knowledge graph (TKG) representation learning aims to project entities and relations in TKGs to a low-dimensional vector space while preserving the evolutionary nature of TKGs. Most existing methods treat knowledge that happens at different times separately, which fails to explore how temporal knowledge graphs evolve over time. Actually, TKGs should evolve both on the local and global structures. The local structure evolution describes the formation process of the graph structure in a detailed manner, while the global structure evolution refers to the dynamic topology (e.g., community partition) of the graph, which is derived from the continuous formation process. Both are key factors for understanding the evolutionary nature of the TKGs. Unfortunately, little attention has been given to this area of research. In this paper, we propose a new TKG representation learning framework with local and global structure evolutions, named *EvoExplore*. Specifically, we define the local structure evolution as an establishment process of the relations between the entities, and propose a hierarchical-attention-based temporal point process to capture the formation process of the graph structure in a fine-grained manner. For global structure evolution, we propose a novel soft modularity parameterized by the entity representations to capture the dynamic community partition of the TKGs. Finally, we employ a multi-task loss function to jointly optimize the above two parts, which allows *EvoExplore* to learn the mutual influences of the local and global structure evolutions. Experimental results on three real-world datasets demonstrate the superiority of *EvoExplore* compared with the baseline methods. Code is available at https://github.com/zjs123/EvoExplore and https://github.com/zjs123/EvoExplore_MindSpore.

1. Introduction

Knowledge graphs (KGs) such as Freebase [1] and YAGO [2] are knowledge base systems that preserve human knowledge in a structured manner. They have been used in many domains and have proven to be highly valuable in various real-world applications, such as recommender systems [3,4] and question answering [5]. Each knowledge graph can be treated as a directed heterogeneous graph where the nodes denote the entities and labeled edges denote the relations between the entities. Each edge with its connected nodes can form a triple (s, r, o) that corresponds to one piece of knowledge. For example, (*Cat*, *Belongs To*, *Mammal*), where *Cat* and *Mammal* are subject and object entities respectively, and *Belongs To* is the relation between them. Because a KG is difficult to directly manipulate, KG representation learning methods are proposed to simplify the manipulation by projecting the entities and relations in KG to a vector space, and they have proven to be effective in many downstream tasks (e.g., link prediction [6]).

The existing methods mostly focus on learning representations for static knowledge graphs, in which the knowledge is eternally valid. However, with the enrichment of event-based interaction data [7], a large amount of knowledge exists that is only valid at a specific time. For example, (*Japan*, *Join In*, *APEC*, *1989/11/7*). As such, knowledge is constantly updated in the real world, and the entities and relations in the KG will also continuously appear and disappear, which makes the knowledge graph evolve over time. To this end, some researchers turn to learning representations for such time-evolving knowledge graphs, which are termed temporal knowledge graph (TKG) representation learning.

A TKG can be viewed as a natural extension of a KG in the time dimension, in which each knowledge representation is preserved as a quadruple (s, r, o, τ) and τ denotes the valid time period of the knowledge. TKG representation learning aims to project

* Corresponding author at: Shenzhen Institute for Advanced Study, UESTC, Shenzhen 518110, China.
E-mail address: shaojie@uestc.edu.cn (J. Shao).

https://doi.org/10.1016/j.knosys.2022.109234
0950-7051/

entities and relations to a low-dimensional vector space while preserving the evolutionary nature of a TKG and its semantics. Existing methods either learn the distinct representations on each snapshot, which fails to capture the detailed mechanism of the TKG formation (e.g., HyTE [8] and TNTComplEX [9]), or they simplify the evolution of a TKG as a diachronic entity representation, which is unable to leverage the TKG's structural dependencies (e.g., DE-DistMult [10] and ATiSE [11]). Specifically, the structural dependencies of a TKG refer to the relevance among the entities in the graph structure (e.g., adjacency and community members). Since the labeled edges among the nodes reflect the semantics of the entities and relations and their semantics evolve as the graph structure changes, learning the representations by modeling the structural dependencies of TKGs helps to understand how a TKG evolves over time. This is what we mean by leveraging structural dependencies. Although some previous methods, such as TeMP [12], employ graph neural networks (GNNs) to aggregate the neighbor information, they can only implicitly model the local structure via low-dimensional vectors, which fail to capture the structural dependencies in a fine-grained manner. Therefore, despite their effectiveness, none of the existing methods can explore how a TKG evolves over time.

Challenge and our approach: Considering that structural dependencies exist both on local and global structures, we argue that TKGs should also evolve both on the local and global structures. For a local structure, the evolution of TKGs is driven by the continuous establishment of relations. As shown in Fig. 1, from τ_i to τ_k, new relations are constantly established between entity *APEC* and the other entities, which reflects the connection strategy of Asia-Pacific Economic Cooperation to continuously attract new countries to join in. Meanwhile, the relation establishments are inevitably determined by the historical knowledge of the participating entities, e.g., entity *Japan* has relation *Member of* with entity *APEC* at time τ_j and τ_k because they used to have relation *Join In* in prior time τ_i. Therefore, considering the local structure's evolution by modeling the continuous establishment of relations helps to capture the formation process of the graph structure in a detailed manner. For a global structure, the evolution of a TKG can be reflected by its dynamic community partition over time. As shown in Fig. 1, with the continuous establishment of relations, three separate communities gradually merged into a larger community, which reflects the formation process of a new community (i.e., APEC member countries). Entities in the same community tend to have a closer relevance with each other, which reveals the inherent evolution pattern of a graph structure. Therefore, considering global structure evolution by modeling the dynamic community partition helps to capture the global structural dependencies (i.e., community members), which cannot be reflected in the individual relation establishments.

Furthermore, because the change of community partitions is derived from the continuous establishment of relations, and the establishment of a new relation is also influenced by the current community partition (e.g., entities in the same community are more likely to have relations), evolutions in local and global structures are interdependent and mutually drive the evolution of TKGs. Jointly modeling the local and global structure evolutions helps to explore the detailed evolution mechanism of TKGs and thus can obtain more accurate TKG representations. Unfortunately, there is currently no work in this area.

In this paper, we propose a new framework named *EvoExplore* to learn the representations for temporal knowledge graphs by jointly modeling their local and global structure evolutions. Specifically, for local structure evolution modeling, we note that the continuous establishment of relations can be viewed as discrete sequential events. Inspired by the temporal point process, which has been widely used to model discrete sequential events, we propose a novel hierarchical-attention-based temporal point process that decouples the influence of historical knowledge in the similarity of the relations and the similarity of the entities, which helps to capture the fine-grained formation process of the graph structure. Global structure evolution modeling is used to capture the smooth evolution of community partitions while considering the heterogeneity of the TKGs. We design a novel soft modularity parameterized by entity representations as the metric for optimization, which imposes constraints on representation learning by leveraging the structural dependencies among the entities. Finally, by jointly optimizing the above two parts, *EvoExplore* can effectively learn the mutual influence of the local and global structure evolutions. The local structure evolution helps our framework capture the detailed evolution mechanism of TKGs while the global structure evolution helps to consider the community structure of TKGs. Our main contributions are as follows:

- To the best of our knowledge, this is the first study to incorporate both the local and global structure evolutions into TKG representation learning.
- We propose a new framework named *EvoExplore*, to simultaneously model the local and global structure evolutions. It contains a novel hierarchical-attention-based temporal point process and a novel community partition component based on soft modularity.
- We conduct comprehensive experiments on three benchmark datasets, and the experimental results demonstrate the superiority of *EvoExplore* over the state-of-the-art methods.

2. Related work

In this section, we give an overview of the typical knowledge graph and temporal knowledge graph representation learning methods.

2.1. Representation learning for a knowledge graph

Recently, a number of methods have been proposed to learn representations for static knowledge graphs, and they can be roughly divided into three categories: linear models, factorization models, and neural network models. Linear models encode the interactions between the entities by applying linear operations. For example, TransE [13] views relations as a projection from subject entity to object entity in the vector space, and an additive operation is applied. Since TransE fails to handle one-to-many relations, a series of improved methods have been proposed, such as TransH [14] that projects entities to relation-specific hyperplanes, TransR [15] that maps entities into relation-specific vector spaces, and TransD [16] that constructs the dynamic mapping matrices for the relations. All of the above methods use Euclidean distance to measure the distance between the relational projections of the entities. TransA [17] uses the Mahalanobis distance to achieve more adaptive metric learning, and KG2E [18] projects the entities and relations into Gaussian space and uses KL-divergence. Recently, several methods regard the relational projection as a rotation operation, such as RotatE [19], while others attempt to take the ontology information into account, such as TransO [20]. Factorization models attempt to decompose a knowledge graph into low-rank matrices. For example, RESCAL [21] utilizes three-way factorization to learn KG representations, and DistMult [22] simplifies RESCAL by using diagonal matrices. Instead of learning representations in a real-valued vector space, ComplEx [23] extends DistMult by introducing complex-valued representations. By introducing three-way Tucker tensor decomposition, TuckER [24] decomposes a knowledge graph as a core tensor and the

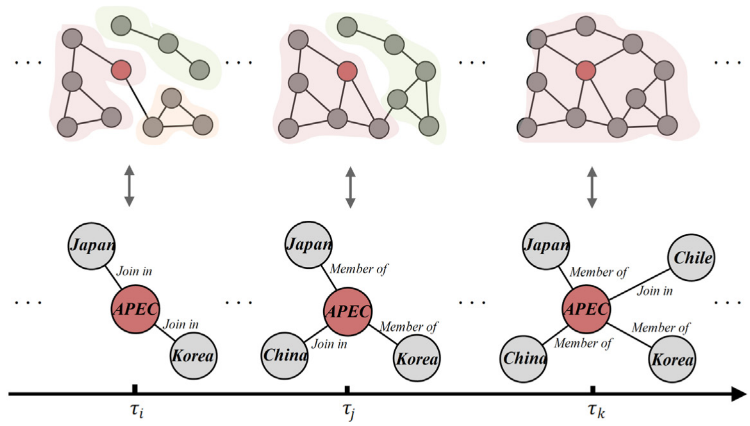


Fig. 1. Illustration of how temporal knowledge graph evolves over time in both local and global structures.

representation vectors of the entities and relations. To solve the independence issue of the entity representations, simplE [25] introduces the inverse of the relations and considers the distance of both the original triples and inverse triples. Neural network models have yielded a remarkable predictive performance in many fields, and can also be used to encode the semantics of knowledge graphs. Since convolutional neural networks (CNNs) can learn deep expressive features, ConvE [26] uses 2D convolution over the reshaped representations of entities and relations to model their interactions. ConvKB [27] applies CNNs over the matrix obtained by concatenating the representations of entities and relations. SACN [28] introduces the translational property of linear models to ConvE. Recently, graph neural networks (GNNs) have proven efficient in modeling connectivity structural data. Several methods regard a knowledge graph as a relational directed graph and utilize a GNN to learn the representations. For example, R-GCN [29] utilizes relation-specific matrices to model the multi-relational nature of a knowledge graph. KBAT [30] utilizes relation representations to calculate the attention score between entities, and CompGCN [31] performs a composition operation over each relation and the corresponding neighborhood entity to capture their interactions. Despite their effectiveness, all the above methods are designed for static knowledge graphs, which are unable to accurately learn representations for temporal knowledge.

2.2. Representation learning for temporal knowledge graph

To extend static knowledge graph representation methods to the temporal domain, some researchers attempt to learn the representations for timestamps. For example, HyTE [8] uses time representations to project knowledge at each snapshot to a time-specific hyperplane and then applies TransE in each hyperplane to learn the representations for each snapshot. TTransE [32] first adds a relation representation and time representation to obtain a translation vector and then uses it to translate the subject entity to the object entity in a vector space. TNTComplEX [9] regards a temporal knowledge graph as a 4-order tensor and learns the TKG representation via a tensor decomposition based on a new temporal regularization scheme. TA-DistMult [33] utilizes the temporal information to constrain relation representations and constructs temporal relation representations for each knowledge instance with a digital-level LSTM model. However, these methods assume that knowledge occurring at different times is independent and separately learn representations on each snapshot, which fails to capture the formation mechanism of the graph structure. Other works learn diachronic entity representations to describe the evolution of TKG. Know-evolve [34] uses a bilinear model and employs a deep recurrent neural network to learn non-linearly evolving entity representations. DE-DistMult [10] regards entity representations as timestamp-dependent variables and learns a nonlinear representation function to generate entity representations based on their corresponding timestamps. ATiSE [11] considers the uncertainty of entity semantics and learns representations in the space of multi-dimensional Gaussian distributions. DyERNIE [35] is a non-Euclidean representation learning method that learns evolving entity representations in a product of Riemannian manifolds. However, these methods ignore the graph structure and thus are unable to leverage the structural dependencies in TKGs. Recently, several methods attempt to learn TKG representations via message passing mechanisms. For example, TeMP [12] proposes a temporal message passing framework by combining graph neural networks and temporal dynamics models. RE-NET [36] proposes a recurrent event encoder to encode historical knowledge and employs a neighborhood aggregator to model the concurrent knowledge. Although they can use the information of the graph structure, they only consider the local structure. Recently, increasing attention has been given to future forecasting in TKGs, and several methods have been proposed. CyGNet [37] uses a dynamic historical vocabulary to model the frequency of historical knowledge. xERTE [38] constructs an inference graph for each query and employs attentive propagation to find the target entities. RE-GCN [39] learns the evolutionary representations of the entities and relations by recurrently modeling a TKG sequence. TITer [40] combines path representations with a reinforcement learning model for explainable forecasting. However, they implicitly model the history information to learn whether knowledge occurs, which is unable to capture the TKGs' detailed evolution mechanism. Although GHNN [41] also employs a temporal point process to model TKGs, it uses a continuous-time LSTM to encode the historical information as a low-dimensional vector, in which a large amount of information will be lost during the dimension compression. In contrast, our *EvoExplore* avoids compression by decoupling the history as the similarity of entities and relations and employs a hierarchical attention mechanism to model the historical information in a fine-grained manner. Therefore, despite the effectiveness of the existing methods, they either are unable to explore the fine-grained evolution process of the temporal knowledge graphs or fail to leverage the structural dependencies of the TKGs, especially in a global structure. None of them can adequately explore how temporal knowledge graphs evolve over time.

Table 1
Notations and descriptions.

Notation	Description
$\mathcal{G}$	A temporal knowledge graph
$\mathcal{E}, \mathcal{R}, \mathcal{T}$	Entity set, relation set and valid timestamp set
$\mathcal{A}$	Community partition set
$\mathcal{C}_\tau$	Community partition at time τ
(s, r, o, τ)	A quadruple in temporal knowledge graph
d	Dimension of representations
K	Number of communities
$\mathbf{u}_e^\tau$	Representation of entity e at time τ
$\mathbf{z}_r$	Representation of relation r
$\mathbf{c}_i^\tau$	Community representation of entity i at time τ
α, β	Attention scores
$\mathbf{B}$	Soft modularity matrix
$\mathbf{H}$	Community assignment matrix
$\mathbf{S}$	Connection strength matrix
$\mathbf{W}, \mathbf{Q}, \mathbf{V}, \mathbf{F}$	Parameter matrix

3. Preliminaries

We now present the necessary background and formally formulate the problem, and introduce the preliminaries regarding the evolution of temporal knowledge graphs. We list the specific notations and their descriptions in Table 1.

3.1. Evolution of the temporal knowledge graph

Definition 1 (*Temporal Knowledge Graph*)**.** A temporal knowledge graph (TKG) is defined as a directed graph with timestamps $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity set and relation set, respectively. $\mathcal{T}$ is the set of valid timestamps. Each knowledge in TKG is represented as a quadruple (s, r, o, τ), in which $s \in \mathcal{E}$ and $o \in \mathcal{E}$ are the subject and object entities respectively, and $r \in \mathcal{R}$ is the relation between them. $\tau \in \mathcal{T}$ indicates the valid time of the fact.

Definition 2 (*Evolution in a Local Structure*)**.** Given a TKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, the local structure evolution refers to the establishment process of the relations between the entities, denoted as a discrete event sequence $\mathcal{I} = \{(s, r, o, \tau)_n\}_{n=1}^{|\mathcal{I}|}$ which is ordered by the corresponding time annotation τ. $|\mathcal{I}|$ is the number of observed knowledge in TKG and each element (s, r, o, τ) represents a temporal event that relation r establishes between entities s and o at time τ. We only consider the establishment of relations because the deletion time of the knowledge is not provided in TKG.

Definition 3 (*Evolution in a Global Structure*)**.** Given a TKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, the global structure evolution describes the evolution process of the community partition in TKG, denoted as $\mathcal{A} = \{\mathcal{C}_\tau\}_{\tau=1}^{|\mathcal{T}|}$, in which each element $\mathcal{C}_\tau = \{c_1, c_2, \dots, c_K\}$ denotes the community partition of the temporal knowledge graph at time τ and c_i is a set of tightly connected entities in a TKG, termed a community. K is the number of communities.

3.2. Temporal point process

The temporal point process is a stochastic process used to model discrete sequential events, which assumes that the historical events before time τ will influence the occurrence of current events and that the probability of each event occurring is characterized by a conditional intensity function $\lambda(\tau)$ given the historical events. In this paper, we use the Hawkes process [42] to model the establishment of relations, since it supposes that the occurrence of an event is influenced by both the current state and historical events, and the historical events can temporarily excite current events, which can consider both the connection strategy and the historical influence of the participating entities. The conditional intensity function of the Hawkes process is defined as

$$\lambda(\tau) = \mu(\tau) + \int_{-\infty}^{\tau} k(\tau - s) dN(s), \tag{1}$$

where $\mu(\tau)$ is the base rate used to model the arrival of spontaneous events. $k()$ is a kernel function used to model the time decay effects of the historical events, while $N(s)$ is the number of historical events until τ.

3.3. Problem formulation

The goal of our *EvoExplore* is to learn the representation for a temporal knowledge graph, by jointly modeling its local and global structure evolutions with their mutual influences. We formally define the problem as follows:

Definition 4 (*Representation Learning for Temporal Knowledge Graph*)**.** Given a TKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, representation learning for a temporal knowledge graph aims to learn two mapping functions $f_e : \mathcal{E} \to \mathbb{R}^d$ and $f_r : \mathcal{R} \to \mathbb{R}^d$ to map the entities and relations in the TKG to a vector space, respectively, where d is the dimensionality of the vector representations. The objective of the mapping functions is to capture the evolutionary nature of the temporal knowledge graph from both the local and global structure evolutions.

4. The proposed approach

Our proposed framework *EvoExplore* takes a TKG as input and learns representations for the entities and relations in a TKG by jointly modeling the evolutions of the local and global graph structures. It consists of three basic components. As depicted in Fig. 2, first, to model the local structure evolution, we characterize the relation establishments via a temporal point process, and propose a hierarchical attention mechanism to learn the influence of the historical events with self-adaptive importance for each participating entity (a). Then, to model the global structure evolution, we design a soft modularity as the metric to capture the community structure in a TKG, which considers both the smoothness of the community evolution and the heterogeneity of the TKG (b). Finally, we optimize *EvoExplore* by jointly predicting the relation establishment and community partition over time (c). We will elaborate the details in the following sections.

4.1. Local structure evolution modeling

With temporal knowledge graphs evolving over time, new relations constantly establish between entities, which can be regarded as a series of discrete sequential events. Obviously, the occurrence of an event is determined not only by the participating entities but also by the related historical events, which affects the current events to varying degrees. Inspired by the temporal point process, which has been widely used to model the discrete sequential events, we propose a novel hierarchical-attention-based temporal point process to model the establishment of each relation in a TKG. Specifically, given a temporal event (s, r, o, τ), which denotes that relation r is established between entities s and o at time τ, we follow the strategy of the Hawkes process to define the corresponding occurrence intensity as

$$\tilde{\lambda}_{s,o}^{r}(\tau) = \mu_{s,o}^{r}(\tau) + \theta \sum_{\tau_i < \tau} g_{s,o}^{r}(\tau_i) k(\tau - \tau_i), \tag{2}$$

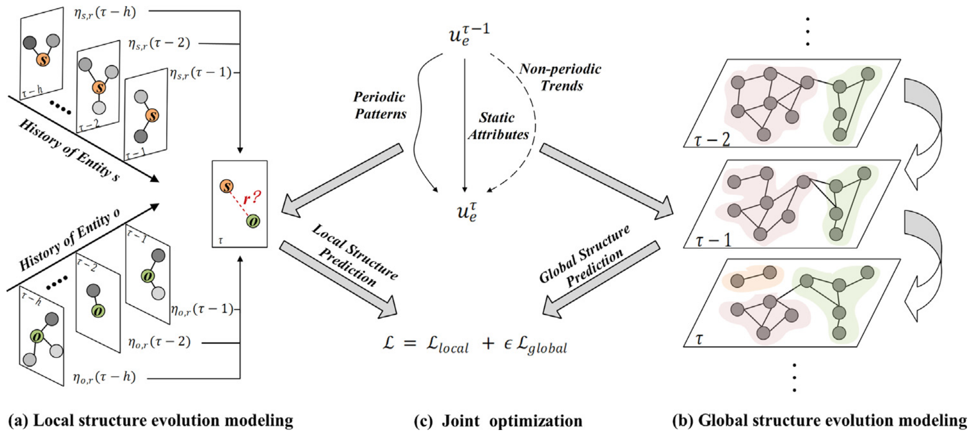


Fig. 2. Overall architecture of *EvoExplore*. (a) Local structure evolution modeling by predicting the relation establishment in the TKGs. (b) Global structure evolution modeling by leveraging the time-evolving community partition of the TKG. (c) Joint optimization to consider the mutual influence of the above two parts. The learning of the entity representations is guided by the above two parts, which can integrate the evolution patterns of both the local and global structures into the learned representations.

where $\mu^r_{s,o}(\tau)$ is the base rate of the temporal event (s, r, o, τ) which describes its spontaneous intensity, and $g^r_{s,o}(\tau_i)$ denotes the influence of the historical events that occurred at time τ_i. θ is a trade-off parameter used to balance the magnitude of the above two parts. Because the influence of historical events should decrease with time, the term $k(\tau-\tau_i) = exp(-\gamma|\tau - \tau_i|)$ is a time decay function with a learnable decay rate γ.

Base rate. Since whether a relation spontaneously established is related not only to the semantics of the relation itself but also to the semantics of the two participating entities, we calculate the base rate of a temporal event based on both the relations and entities. Specifically, inspired by TransE [13] which views the relations as a translation from subject entity to object entity, we believe that if the transformed subject entity is similar to the object entity, the spontaneous intensity of a temporal event will be high. However, TransE utilizes a non-parameterized operation to measure the similarity, which lacks an expression. Therefore, we employ a parameterized inner product operation to obtain the base rate of the temporal event (s, r, o, τ) as

$$\mu^r_{s,o}(\tau) = \mathbf{u}_o^{\tau\top}\mathbf{W}(\mathbf{u}^\tau_s + \mathbf{z}_r), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is a matrix used to measure the similarity. $\mathbf{z}_r \in \mathbb{R}^d$ is the representation of relation r, which is learnable during training, and $\mathbf{u}^\tau_s \in \mathbb{R}^d$ and $\mathbf{u}^\tau_o \in \mathbb{R}^d$ are representations of the subject entity s and object entity o at time τ, respectively. As we mentioned before, the connection strategy of an entity will affect its possibility of building relations with other entities. Since there are different entities with various connection strategies in TKG, and the connection strategy may evolve over time, we obtain the representation for each entity e at time τ as

$$\mathbf{u}^\tau_e = \sin(\boldsymbol{\theta}_e \cdot \tau) + \tanh(\boldsymbol{\omega}_e \cdot \tau) + \boldsymbol{\nu}_e, \tag{4}$$

where $\boldsymbol{\theta}_e$, $\boldsymbol{\omega}_e$, and $\boldsymbol{\nu}_e \in \mathbb{R}^d$ are entity-specific vectors that preserve the different evolution patterns of their connection strategy. Specifically, we use a periodic function in the first term of Eq. (4) to preserve the periodic evolution patterns, such as "*The Olympic Games are held every four years*". In the second term, we use a monotonic function to preserve the non-periodic evolution trends, e.g., a person ought to be born, marry, and die. The third term preserves some static attributes of an entity, such as the gender of a person or the location of a city. In this way, the obtained entity representations can accurately describe the different connection strategies of the entities at each timestamp, which motivates our framework to effectively recognize the spontaneous occurrences of temporal events.

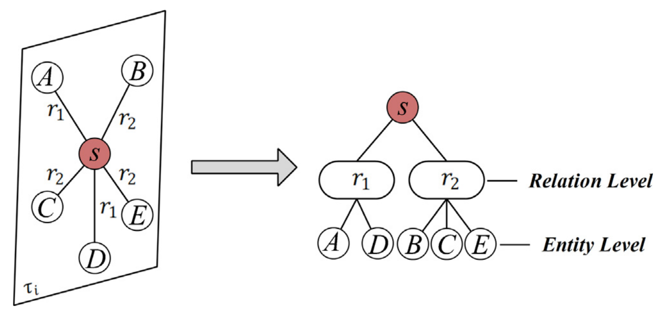


Fig. 3. Illustration of transforming the neighborhood of entity s at time τ_i to the hierarchical structure.

History's influence. In addition to the connection strategies of participating entities, whether a temporal event occurs also depends on the related historical events, in which both the history of the subject entity and object entity have impacts. For example, whether the temporal event (*Joe Biden*, *Win*, *US Presidential Election*, 2020) occurs not only depends on whether the USA needs a change of office in 2020 but also depends on whether Joe Biden runs. To this end, we formalize the influence of historical events on temporal event (s, r, o, τ) as

$$g^r_{s,o}(\tau_i) = \eta_{s,r}(\tau_i) + \eta_{o,r}(\tau_i), \tag{5}$$

where $\eta_{s,r}(\tau_i)$ and $\eta_{o,r}(\tau_i)$ denote the influence of the historical events that contain subject entity s and object entity o, respectively.

However, historical events involve various types of entities and relations, which are highly heterogeneous and difficult to directly leverage to calculate $\eta_{s,r}(\tau_i)$ and $\eta_{o,r}(\tau_i)$. We note that the historical events of each entity are exposed to different entities through multiple relations and can be naturally viewed as a hierarchical structure as shown in Fig. 3. The influence of historical events can be naturally decoupled as the similarity between the historical entity and the current target entity, as well as the similarity between the historical relation and the current target relation. The historical events with more similar entities and more similar relations to the current events will be more important. Furthermore, the similarities between the entities are more critical because similar entities are more likely to establish relations. To this end, we quantify the influence of each historical event as an entity similarity and propose a hierarchical attention mechanism to compute varying importances for the different historical events. In this way, the occurrence intensity is proportional to the similarity of the relations and entities. Formally, we define the influence of historical events that involve entity e (e can be either s or o) as

$$\eta_{e,r}(\tau_i) = \sum_{\bar{r}\in\mathcal{R}_e^{\tau_i}} \alpha_{r,\bar{r}} \sum_{h\in\mathcal{E}_{e,\bar{r}}^{\tau_i}} \beta_{h,x}(\mathbf{u}_h^{\tau_i\top}\mathbf{Q}\mathbf{u}_x^{\tau_i}), \tag{6}$$

where $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is a matrix used to measure the similarity between the entities. $\mathcal{R}_e^{\tau_i}$ is the set of relations in which entity e establishes at time τ_i, and $\mathcal{E}_{e,\bar{r}}^{\tau_i}$ is the set of entities, each of which establishes relation $\bar{r}$ with entity e at time τ_i. x is the current target entity (i.e., x is o if e is s). $\alpha_{r,\bar{r}}$ and $\beta_{h,x}$ are the attention scores used to model the varying importances of the historical events based on the similarity of relations and the similarity of entities, respectively.

To obtain the varying importances for the different historical events, we first focus on the relations, which greatly determine the relevance of a historical event to the current event. For example, since people are likely to graduate from the school where they studied, a temporal event that contains relation *Study In* is largely relevant to the temporal event that contains relation *Graduate From*. Historical events involving more relevant relations to current events tend to be more important. To model the importance of historical events based on their relations, we define the relation-level attention as

$$\alpha_{r,\bar{r}} = \frac{exp(\mathbf{z}_{\bar{r}}^{\top}\mathbf{V}\mathbf{z}_r)}{\sum_{r'\in\mathcal{R}_e^{\tau_i}} exp(\mathbf{z}_{r'}^{\top}\mathbf{V}\mathbf{z}_r)}, \tag{7}$$

where $\mathbf{V} \in \mathbb{R}^{d\times d}$ is a matrix used to measure the relevance between the relations, and $\mathbf{z}_{r'}, \mathbf{z}_{\bar{r}} \in \mathbb{R}^d$ are representations of the relations in historical events. In addition to the relations, the entities also need to be considered, because for each relation, an entity may interact with various other entities through it, and different entities will result in a different importance. For instance, China may have cooperation with various other countries in the past, but when it determines whether it currently cooperates with Japan, historical cooperation with Korea is more important than historical cooperation with the USA, because both Japan and Korea are APEC member countries. To this end, we regard entities that have the same relation with the target entity e as a group, and compute the entity-level attention as

$$\beta_{h,x} = \frac{exp(\mathbf{u}_h^{\tau_i\top}\mathbf{Q}\mathbf{u}_x^{\tau_i})}{\sum_{h'\in\mathcal{E}_{e,\bar{r}}^{\tau_i}} exp(\mathbf{u}_{h'}^{\tau_i\top}\mathbf{Q}\mathbf{u}_x^{\tau_i})}, \tag{8}$$

where we still use matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$ to measure the similarity between the entities for consistency. $\mathbf{u}_h^{\tau_i}, \mathbf{u}_{h'}^{\tau_i} \in \mathbb{R}^d$ are representations of historical entities at time τ_i.

Local structure evolution prediction. Based on the above calculations, we can obtain the occurrence intensity of each relation establishment event. As it may derive negative values, we apply an exponential function to transform it to a positive value (i.e., $\lambda_{s,o}^r(\tau) = exp(\tilde{\lambda}_{s,o}^r(\tau))$). On this basis, for each observed temporal event (s, r, o, τ), we construct two corresponding queries $(s, r, ?, \tau)$ and $(?, r, o, \tau)$. We hope the true answer has a higher occurrence intensity compared with the other entities in the entity set. To this end, we define the probability of a relation r being established between the entities s and o at time τ as

$$p(s, r, o|\mathcal{I}(\tau)) = \frac{\lambda_{s,o}^r(\tau)}{\sum_{e\in\mathcal{E}}(\lambda_{e,o}^r(\tau) + \lambda_{s,e}^r(\tau))}, \tag{9}$$

where $\mathcal{I}(\tau)$ denotes the observed temporal events before time τ and we maximize the above probability by minimizing the following objective function

$$\mathcal{L}_{local} = -\sum_{(s,r,o,\tau)\in\mathcal{I}} \log\ p(s, r, o|\mathcal{I}(\tau)). \tag{10}$$

Modeling the continuous establishments of relations helps to understand the detailed mechanism of how a graph structure evolves. Therefore, our framework effectively captures the evolutionary nature of temporal knowledge graphs in a fine-grained manner.

4.2. Global structure evolution modeling

With the continuous establishment of relations, there is an integration of jointly make up the slow evolution process of the global structure of a TKG. Incorporating the evolution of the global structure not only helps our framework to explore the high-order evolution trends of the entities but also helps to capture the global structural dependency at each timestamp. In this section, we regard the global structure evolution of the temporal knowledge graph as its dynamic community partition, which evolves over time, and design a soft modularity to model the community structure in the TKGs. Specifically, modularity [43] is a metric used to describe the quality of the community partition in complex networks, which is defined as the difference between the number of edges in communities and the expected number of such edges over all pairs of nodes. The modularity of a network can be formalized as

$$Q = \frac{1}{4m} Tr(\mathbf{H}^{\top}\mathbf{B}\mathbf{H}), \tag{11}$$

where $Tr()$ is the trace of a matrix, while m is the total number of edges in the network. $\mathbf{H} \in \mathbb{R}^{|\mathcal{E}|\times K}$ denotes the community assignment matrix, in which K is the total number of communities and each element $\mathbf{H}_{ik}$ is set as 1 if entity i belongs to community k, or set as 0 otherwise. $\mathbf{B} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is the modularity matrix, in which each element is defined as

$$\mathbf{B}_{ij} = \mathbf{A}_{ij} - \frac{\rho_i\rho_j}{2m}, \tag{12}$$

where $\mathbf{A} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is the adjacency matrix of the network and ρ_i is the degree of node i. Note that $\frac{\rho_i\rho_j}{2m}$ is the expected number of edges between nodes i and j if the edges are placed randomly. Even though modularity has been widely used to model the communities in graphs, it is designed for static homogeneous networks. For temporal knowledge graphs, the entities are connected via different types of relations, which results in different connection strengths, and the graph structure also varies over time, which leads to the community partition at each timestamp being different.

To appropriately model the communities in a temporal knowledge graph, we propose a novel soft modularity. To model the varying connection strengths between entities, we first replace the adjacency matrix of the graph with our connection strength matrix, which is derived from the relations between a pair of entities. Formally, each element of the connection strength matrix is defined as

$$\mathbf{S}_{ij}^{\tau} = \sum_{r\in\mathcal{R}_{i,j}^{\tau}} \varphi(\mathbf{z}_r^{\top}\mathbf{a}), \tag{13}$$

where $\mathbf{S}^{\tau} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is the connection strength matrix at each timestamp τ. i and j are the indices of two adjacent entities. $\mathcal{R}_{i,j}^{\tau}$ is the set of relations that exist between entity i and entity j at time τ, and $\mathbf{a} \in \mathbb{R}^d$ is a learned vector used to measure the connection strength of each relation. φ is an activation function. On this basis, we first treat the graph structure at each timestamp separately, and for each timestamp τ, we can obtain the corresponding soft modularity matrix $\tilde{\mathbf{B}}^{\tau} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$, in which each element is defined as

$$\tilde{\mathbf{B}}_{ij}^{\tau} = \mathbf{S}_{ij}^{\tau} - \frac{\rho_i^{\tau}\rho_j^{\tau}}{2m^{\tau}}, \tag{14}$$

where ρ_i^τ and ρ_j^τ are the degrees of entity i and entity j at time τ, respectively. m^τ is the total number of edges in the TKG at time τ. By replacing the modularity matrix in Eq. (11) with our soft modularity matrix, we can obtain a series of time-specific soft modularities, and by maximizing these time-specific soft modularities, we can model the TKG community at each timestamp separately.

It should be noted that, first, the above soft modularity requires a hard assignment of community label for each entity. However, each entity in a TKG may belong to several communities simultaneously, e.g., *Japan* may belong to both the community *APEC member countries* and the community *WTO member countries*. Second, the community partition of TKG at different timestamps is not independent, so separately treating them ignores their temporal relevance. To this end, we follow the strategy in [44] to relax the hard assignment at each timestamp as $Tr(\mathbf{H}^{\tau\top}\mathbf{H}^\tau) = |\mathcal{E}|$. It is easy to ascertain that for each entity i, this constraint is equivalent to $\sum_{k=1}^K \mathbf{H}_{ik}^{\tau\,2} = 1$. Therefore to meet the constraint, we normalize each element of $\mathbf{H}^\tau \in \mathbb{R}^{|\mathcal{E}|\times K}$ as

$$\tilde{\mathbf{H}}_{ik}^\tau = \frac{\sqrt{\mathbf{H}_{ik}^\tau}}{\sqrt{\sum_{k=1}^K \mathbf{H}_{ik}^\tau}}, \tag{15}$$

where $\tilde{\mathbf{H}}_{ik}^\tau$ denotes the probability of entity i belonging to community k at time τ. Considering the smoothness of the community partition over time, we parameterize the community assignment of each entity via its representation at the current timestamp and its previous community assignment. Formally, the community assignment of entity i at timestamp τ is defined as

$$\mathbf{H}_i^\tau = \varphi(\mathbf{F}\cdot(\mathbf{u}_i^\tau + \mathbf{c}_i^{\tau-1})), \tag{16}$$

where $\mathbf{F} \in \mathbb{R}^{K\times d}$ is an auxiliary matrix used to map an entity representation to a community assignment. $\mathbf{c}_i^{\tau-1} \in \mathbb{R}^d$ is the representation of the community to which entity i belongs at time $\tau - 1$ with the highest probability. We define the representation of a community as the average of the representations of the involved entities, so that it can be easily pre-calculated before each iteration.

In this way, our soft modularity can be easily optimized while considering both the smoothness of the community evolution and the heterogeneity of TKGs. By jointly maximizing the soft modularity at each timestamp, *EvoExplore* can effectively model the time-evolving community partition of a TKG and thus capture the evolutionary nature of the temporal knowledge graph from its global structure. The corresponding global structure evolution loss is defined as

$$\mathcal{L}_{global} = -\sum_{\tau\in\mathcal{T}} \frac{1}{4m^\tau} Tr(norm(\mathbf{H}^\tau)^\top \tilde{\mathbf{B}}^\tau norm(\mathbf{H}^\tau)), \tag{17}$$

where *norm*() is the normalization procedure defined as Eq. (15). By minimizing the above loss function, our framework can leverage the global structural dependency of TKGs to impose constraints on representation learning, which helps to enhance the capability of the learned representations to preserve the graph structure and the evolutionary trends of entities.

4.3. Joint modeling

As local and global structure evolutions are interdependent, and mutually drive the evolution of the temporal knowledge graphs, we design a multi-task loss function to learn representations for the entities and relations by jointly predicting the relation establishment and community partition of TKGs, which is given by

$$\mathcal{L} = \mathcal{L}_{local} + \epsilon\mathcal{L}_{global}, \tag{18}$$

where $\epsilon \in [0, 1]$ is a hyperparameter used to balance the magnitude of these two terms.

Optimization. Since the temporal knowledge graph contains a large number of entities, calculating all the combinations of positive and negative samples as defined in Eq. (9) is computationally expensive. To address this problem, we employ a sampling technique to approximately optimize $\mathcal{L}_{local}$. Specifically, since we have transferred the occurrence intensity to a positive number via an exponential function, the probability of a temporal event occurring $p(s, o, t|H(\tau))$ is actually a softmax unit applied to $\tilde{\lambda}_{s,o}^r(\tau)$, which can be approximately optimized by a negative sampling [45]. Negative sampling can avoid the calculation over the entire entity set, and thus effectively reduce the computation. For each temporal event $(s, r, o, \tau) \in \mathcal{I}$, we randomly sample L negative entities and generate two negative knowledge instances (e, r, o, τ) and (s, r, e, τ) for each negative entity. In this way, the loss function Eq. (10) can be rewritten as

$$\mathcal{L}_{local} = -\sum_{(s,r,o,\tau)\in\mathcal{I}} \log\ \sigma(\tilde{\lambda}_{s,o}^r(\tau)) - \sum_{l=1}^{L} \mathbb{E}_e[\log(\sigma(-\tilde{\lambda}_{e,o}^r(\tau)))] - \sum_{l=1}^{L} \mathbb{E}_e[\log(\sigma(-\tilde{\lambda}_{s,e}^r(\tau)))], \tag{19}$$

where $\sigma(x) = exp(x)/(1 + exp(x))$ is the sigmoid function. In addition, as the length of the history event sequence largely influences the computation complexity of $\tilde{\lambda}_{s,o}^r(\tau)$, we fix the maximum length of history h and only retain the most recent history events, and we will discuss the influence of the different history lengths in Section 5.

Furthermore, since $\mathcal{L}_{global}$ can be easily optimized via the gradient descent algorithm, we adopt stochastic gradient descent (SGD) to optimize the overall loss function. In each iteration, we sample a mini-batch of temporal events to calculate $\mathcal{L}_{local}$, and use the entities involved in the current batch to calculate $\mathcal{L}_{global}$. Both of them will be used to update the parameters during back propagation.

Complexity analysis. The time complexity of the local structure evolution modeling is $\mathcal{O}(MhL|\mathcal{I}||\mathcal{N}|d^2)$, where M is the number of iterations and d is the representation dimensionality. h and L are the length of the history and the number of negative samples respectively. $|\mathcal{I}|$ is the total number of temporal events in TKG and $|\mathcal{N}|$ is the number of neighbors of each temporal event. The above time complexity can be reduced to $\mathcal{O}(Mh|\mathcal{I}||\mathcal{N}|d^2)$ via the in-batch negative sampling strategy. It only samples negative entities from the active entity set of the current timestamp, and thus can avoid calculating the historical influence of the negative samples from scratch. The space complexity of the local structure evolution modeling is $\mathcal{O}(|\mathcal{E}|d + |\mathcal{R}|d + 3d^2)$, in which $|\mathcal{E}|$ and $|\mathcal{R}|$ are the total numbers of the entities and relations in TKG, respectively. Since $\mathcal{L}_{global}$ is also optimized via the mini-batch strategy, the global structure modeling will not radically increase the time complexity and space complexity. For the time complexity, the calculation cost of the global structure modeling mainly comes from Eq. (16), for which the time complexity is $\mathcal{O}(M|\mathcal{I}|d^2)$. For the space complexity, because the community representations are obtained via entity representations, we only introduce the auxiliary matrix $\mathbf{F}$ in this part. Therefore, the space complexity of the global structure modeling is $\mathcal{O}(Kd)$, where $K \ll |\mathcal{E}|$.

5. Experiments

In this section, we compare the performance of *EvoExplore* with a large number of baseline methods to verify its superiority, and provide some variants to demonstrate the effectiveness of each component in our framework. Finally, we analyze *EvoExplore* from several different aspects. Our code and implementation details are publicly available at https://github.com/zjs123/EvoExplore and the version in MindSpore [46] is available at https://github.com/zjs123/EvoExplore_MindSpore.

Table 2
Statistics of datasets.

Dataset	$\vert\mathcal{E}\vert$	$\vert\mathcal{R}\vert$	$\vert\mathcal{T}\vert$	$\vert Train\vert$	$\vert Validation\vert$	$\vert Test\vert$
ICEWS 14	7,128	230	365	72,826	8,941	8,963
ICEWS 05-15	10,488	251	4,071	386,962	46,275	46,092
ICEWS 18	23,033	256	304	373,018	45,995	49,545
GDELT	500	20	366	2,735,685	341,961	341,961

5.1. Datasets

We evaluate the performance of our method and baselines on four public datasets derived from two popular temporal knowledge graph resources, namely ICEWS [7] and GDELT [47]. Simple statistics of the four datasets are summarized in Table 2, and we detail each dataset as follows:

- **ICEWS 14**: This is a short-range version subset of ICEWS recourse, which is released by TA-DistMult [33]. It contains knowledge from 2014/1/1 to 2014/12/31 with a daily granularity, and there are 7128 distinct entities and 230 types of relations.
- **ICEWS 05-15**: This is a long-range version subset of ICEWS recourse released by TA-DistMult, which is almost 5 times larger than ICEWS14. It contains knowledge from 2005/1/1 to 2015/12/31 with a daily granularity, and there are 10,488 distinct entities and 251 types of relations in this dataset.
- **ICEWS 18**: This is another short-range version subset of ICEWS recourse released by xERTE [38]. It contains knowledge from 2018 with a daily granularity, and there are 23,033 distinct entities and 256 types of relations.
- **GDELT**: This is a subset of GDELT resource released by DE-DistMult [10], which contains knowledge from 2015/4/1 to 2016/5/31, with a daily granularity. There are 500 distinct entities and 20 types of relations in this dataset.

5.2. Evaluation protocol

We evaluate the quality of the learned TKG representations via three tasks: link prediction, future forecasting, and relation prediction. Specifically, link prediction is an interpolation task that also refers to graph reconstruction. It aims to predict missing knowledge for timestamps that have appeared in the training process (formally, predicting missing entity for $(s, r, ?, \tau)$ or $(?, r, o, \tau)$ where $\tau \in \mathcal{T}_{train}$). Future forecasting is an extrapolation task that aims to predict missing knowledge for timestamps that are unseen during the training process (formally, predicting missing entity for $(s, r, ?, \tau)$ or $(?, r, o, \tau)$ where $\tau \notin \mathcal{T}_{train}$). The relation prediction task aims to predict whether a given relation r will appear between entities s and o at timestamp τ. Then, by performing various ablation experiments, we evaluate the effectiveness of each component of our framework. We also visualize the learned representations on two-dimensional space. Finally, we perform a parameter sensitivity study.

For each task, we follow the strategy of TransE [13] to measure the model's performance. Specifically, for each test knowledge (s, r, o, τ), we corrupt it by replacing the subject entity, object entity or relation with all possible entities and relations in turn and obtain a list of candidate knowledge. Then, the candidate knowledge and original fact are ranked in a descending order of their plausibility score. The rank of the original knowledge denoted as $rank(s, r, o, \tau)$ is the basic metric, and then we use two kinds of refined metrics based on this to evaluate the performance of each model. One is the mean reciprocal rank (MRR) defined as $MRR = \frac{1}{|Test|}\sum_{(s,r,o,\tau)\in Test}\frac{1}{rank(s,r,o,\tau)}$, which is the average of the reciprocal of the rank of each test fact, and a higher MRR denotes an improved model performance. The other is Hits@N which is defined as $Hits@N = \frac{1}{|Test|}\sum_{(s,r,o,\tau)\in Test}, ind(rank(s, r, o, \tau) \leq N)$ where $ind()$ is 1 if the inequality holds and is 0 otherwise.

5.3. Baseline methods

We compare our *EvoExplore* with both static and temporal knowledge graph representation learning methods to verify its superiority. For static knowledge graph representation learning methods, we select the typical method for each category, i.e., TransE [13] for linear models, DistMult [22] for factorization models, and ConvKB [27] for neural network models. For temporal knowledge graph representation learning methods, we use the methods introduced in Section 2.2. In addition, to verify the effectiveness of the global evolution modeling of our framework, we also create a variant *EvoExplore_l*, which only considers the local structure evolution (i.e., $\epsilon = 0$ in Eq. (18)).

5.4. Implementation details

For the baseline methods, we use the released official implementation and the hyperparameter settings in the corresponding papers. We train them with the same number of epochs as our framework for a fair comparison. For our model, we tune the hyperparameters using a grid search. We create 100 mini-batches for each epoch during training, and the number of epochs is set as 500. The learning rate is set to 0.001, the dimension of representations $d \in \{50, 100, 200\}$, negative sampling ratio $L \in \{10, 20, 30, 40\}$, history length $h \in \{1, 2, 3, 4, 5\}$, trade-off $\epsilon \in \{1, 0.1, 0.01\}$ and the number of communities $K \in \{3, 5, 7, 9\}$. The configuration is chosen based on the MRR value on the validation set. All the experiments are performed on an Intel Xeon CPU E5-2640 with 128 GB of main memory and an Nvidia TITAN RTX. We randomly initialize the entity and relation representations, and the negative entities are uniformly sampled from the entire entity set.

5.5. Performance comparison for link prediction

Table 3 illustrates the results of the different methods on the link prediction task. According to the results, we have three major findings. (1) The temporal-aware representation learning methods unsurprisingly have a better performance than the corresponding basic static methods, e.g., HyTE outperforms TransE on all metrics, and DE-DistMult outperforms DistMult on all metrics. This result verifies that incorporating the temporal information helps to obtain more accurate representations for the knowledge graph. However, it should be noted that not all the temporal-aware methods are superior to the static methods. For example, TTransE fails to outperform TransE, which may be because adding time representations in the TransE score function breaks the translation between the entities. This illustrates the importance of modeling the temporal information in a reasonable way. (2) We find that our proposed *EvoExplore* outperforms the baseline methods by a significant margin, and achieves state-of-the-art results on all three datasets in terms of *MRR*. Compared with the other methods, our result achieves 8.3%, 6.9%, and 12.4% higher for the *MRR* evaluation on each dataset respectively, which demonstrates that modeling the evolutionary nature of TKG based on the local and global structure evolutions can generate more accurate TKG representations. One interesting observation is that our framework achieves an impressive performance in *Hits@10* on the GDELT dataset but fails to outperform DyERNIE in *Hits@1*. This is because the GDELT dataset is denser than the other datasets. DyERNIE employs hyperbolic space which can better distinguish between the different entities but the local structure evolution component of our framework will be weakened since several entities may have the same historical interactions, which leads to a limited improvement in *Hits@1*.

Table 3
Comparison of different methods on three datasets for link prediction. The best and second best results in each column are boldfaced and underlined respectively (higher is better for each metric).

Dataset	ICEWS 14				ICEWS 05-15				GDELT			
Models	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*
TransE [13]	32.6	15.4	43.1	64.4	33.0	15.2	44.6	66.3	15.5	6.0	17.8	33.5
DistMult [22]	44.1	32.5	49.8	66.8	45.7	33.8	51.5	69.1	21.0	13.3	22.5	36.6
ConvKB [27]	33.5	22.4	38.7	56.6	–	–	–	–	–	–	–	–
TTransE [32]	22.7	7.2	30.1	58.2	24.3	8.6	31.5	60.9	11.8	0.0	16.0	31.8
HyTE [8]	29.7	10.8	41.6	65.5	31.6	11.6	44.5	68.1	11.8	0.0	16.1	32.5
TA-DistMult [33]	43.5	31.6	49.1	68.0	46.8	35.2	51.8	72.8	20.6	12.4	21.9	36.5
DE-DistMult [10]	50.1	39.2	56.9	70.8	48.4	36.6	54.6	71.8	21.3	13.0	22.8	37.6
TNTComplEX [9]	61.6	51.8	65.7	75.8	66.5	59.0	70.5	80.7	22.3	14.2	23.7	37.9
ATiSE [11]	56.9	46.3	63.9	76.3	52.0	39.7	59.5	77.3	11.6	5.7	11.4	23.5
TeMP [12]	60.7	48.4	68.4	84.0	69.1	56.6	78.2	**91.7**	27.5	19.1	29.7	43.7
DyERNIE [35]	66.9	59.9	71.4	79.7	73.9	67.9	77.3	85.5	45.7	**39.0**	47.9	58.9
EvoExplore_l	68.7	63.5	74.9	83.8	76.6	67.9	82.6	91.0	46.7	33.6	52.3	61.9
EvoExplore	**72.5**	**65.3**	**77.8**	**85.2**	**79.0**	**71.9**	**84.3**	91.5	**51.4**	35.3	**60.2**	**74.8**

Table 4
Comparison of different methods on three datasets for future forecasting. The best and second best results in each column are boldfaced and underlined respectively (the higher is better for each metric).

Dataset	ICEWS 14				ICEWS 05-15				ICEWS 18			
Models	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*
TransE [13]	22.4	13.3	25.6	41.2	22.5	13.0	25.6	42.0	12.2	5.8	12.8	25.1
DistMult [22]	27.6	18.1	31.1	46.9	28.7	19.3	32.1	47.5	10.1	4.5	10.3	21.2
ComplEx [23]	30.8	21.5	34.4	49.5	31.6	21.4	35.7	52.0	21.0	11.8	23.4	39.8
TTransE [32]	13.4	3.1	17.3	34.5	15.7	5.0	19.7	38.0	8.3	1.9	8.5	21.8
TA-DistMult [33]	26.4	17.0	30.2	45.4	24.3	14.5	27.9	44.2	16.7	8.6	18.4	33.5
DE-DistMult [10]	30.2	22.3	34.7	47.1	34.1	24.6	36.8	50.5	17.2	10.0	18.2	29.4
TNTComplEX [9]	32.1	23.3	36.0	49.1	27.5	19.5	30.8	42.8	21.2	13.2	24.0	36.9
CyGNet [37]	32.7	23.6	36.3	50.6	34.9	25.6	39.0	52.9	24.9	15.8	28.2	42.6
RE-NET [36]	38.2	28.6	41.3	54.5	42.9	31.2	46.8	63.4	28.8	19.0	32.4	47.5
xERTE [38]	40.7	**32.7**	45.6	57.3	46.6	37.8	52.3	63.9	29.3	21.0	33.5	46.4
EvoExplore_l	42.2	31.8	48.0	61.6	48.6	37.7	55.1	69.2	30.3	21.1	32.6	48.1
EvoExplore	**43.6**	32.1	**50.6**	**64.7**	**50.0**	**39.4**	**56.1**	**69.6**	**31.5**	**21.7**	**35.9**	**51.0**

However, DyERNIE is unable to consider the structural information of a TKG, so the learned representations lack generalization. Our framework can effectively leverage the structural dependencies among entities. Therefore, our framework can achieve a better result in *Hits@10*. (3) One can see that *EvoExplore* has a better performance than *EvoExplore_l* on all three datasets. This observation demonstrates the benefit of modeling the global structure evolution process of a TKG. It comes from two parts. First, the global structure helps to impose constraints on the learned representations, and it can prevent the entity representations from changing drastically at consecutive timestamps. Second, it helps to preserve the information of the graph structure. Furthermore, although *EvoExplore_l* only considers the local structure evolution of the temporal knowledge graph, it still outperforms most of the baselines, which demonstrates the necessity of capturing the detailed mechanism of relation establishments and the effectiveness of our hierarchical-attention-based temporal point process to capture the fine-grained evolution process of the temporal knowledge graph.

5.6. Performance comparison for future forecasting

We show the performance of the different methods on the future forecasting task in Table 4. It should be noted that we replaced the GDELT dataset with the ICEWS 18 dataset because the ICEWS 18 dataset is more widely used for future forecasting tasks, and many baseline methods did not report the experiential results on the GDELT dataset. According to the results, we find that our proposed *EvoExplore* can still outperform the baseline methods. In terms of *MRR*, our result achieves 7.1%, 8.3% and 7.5% improvements over the state-of-the-art method on each dataset, respectively, which demonstrates the superiority of our model in predicting future links. Furthermore, we notice that our *EvoExplore* gains more improvements on the ICEWS 05-15 dataset. This is because ICEWS 05-15 has a longer period compared with the others. Existing methods are required to encode the historical information as vectors, which limits their ability to model long-term information. Our *EvoExplore* can model the historical information in a fine-grained manner via the hierarchical attention mechanism. Finally, the performance of *EvoExplore_l* is still worse than that of *EvoExplore*, which provides the evidence that modeling the global structure evolution helps to learn more accurate TKG representations.

5.7. Performance comparison for relation prediction

We present the performance for the relation prediction task in terms of MRR in Table 5. Since some methods are unable to be directly applied to the relation prediction task, such as CyGNet [37] and RE-NET [36], we select the typical ones for comparison. According to the results, we observe that *EvoExplore* achieves a better performance than the baseline methods, especially on the ICEWS resource. *EvoExplore* fails to outperform RE-GCN on the GDELT dataset because it contains many concept entities that do not have specific semantics. The concept entities will limit the ability of the attention mechanism to model the similarity. Finally, the advantage of *EvoExplore* for the relation prediction task demonstrates that the joint modeling of the local and global evolutions helps to learn more accurate TKG representations.

5.8. Ablation study

Effect of varying evolution pattern. So far, we obtain the entity representations as a sum of three parts as defined in Eq. (4).

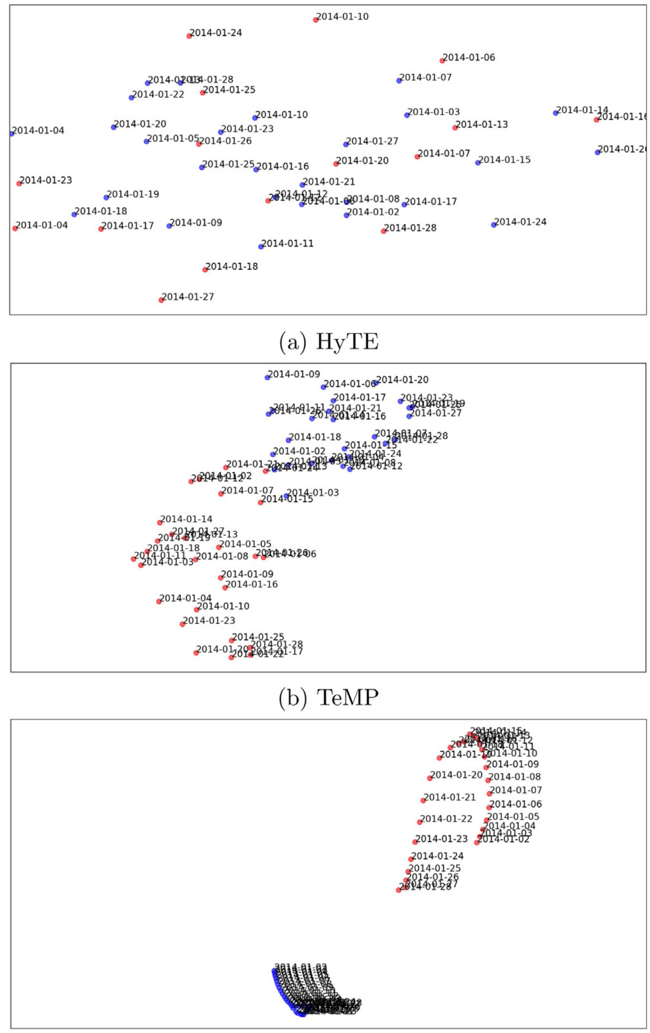
Table 5
Performance of different methods for relation prediction. The best and second best results in each column are boldfaced and underlined respectively.

Dataset	ICEWS 14	ICEWS 05-15	ICEWS 18	GDELT
Models	*MRR*	*MRR*	*MRR*	*MRR*
ConvE [26]	38.8	37.8	37.7	18.8
SACN [28]	38.4	38.2	38.0	18.9
HyTE [8]	29.6	27.5	28.6	11.2
DE-DistMult [10]	39.1	39.6	38.0	17.7
RE-GCN [39]	41.0	40.6	40.5	**19.1**
EvoExplore	**42.3**	**41.4**	**40.8**	18.5

The performance of obtaining entity representations via other approaches can be seen in Table 6. From this table, it can be seen that, first, variants that obtain dynamic entity representations have better performances than the method using static entity representations, which demonstrates the need to model the different connection strategies of the entities at different timestamps. Second, we observe that using the *sin* function has a better performance than using the *tanh* function. We believe this is because compared with *tanh* which is a monotonic function and corresponds to smooth one-off temporal switches, *sin* as a periodic function is more suitable for modeling a repeated establishment of the relations between the entities. Finally, all of them fail to outperform our original method, which provides evidence for the effectiveness of fully considering the different evolution patterns of the entities.

Effect of hierarchical attention mechanism. To verify the effectiveness of our proposed hierarchical attention mechanism, we create a series of variants of *EvoExplore*, including only using entity-level attention, only using relation-level attention, and replacing the attention mechanism with the average operation. As shown in Table 6, one notices that the variants with attention mechanisms have better performances, which demonstrates the need to consider the varying importance of the historical events. Furthermore, our original method using a hierarchical attention mechanism has better performance than those variants using a single attention mechanism, which verifies the superiority of decoupling the influence of the historical knowledge as the similarity of relations and the similarity of entities. One interesting observation is that using only entity-level attention performs much better than only using relation-level attention. This may be because in a TKG, the number of entity types is much greater than the number of relation types, which leads to modeling the varying importance based on entities being more fine-grained.

Effect of soft modularity. When modeling the community partition of a TKG, we hypothesize that different relations will result in different connection strengths, and the community partitions at the different timestamps are related. Here, we verify the validity of the above assumptions. We first attempt to replace our soft modularity matrix with an adjacent matrix, as shown in Table 6. It is outperformed by our original method in all metrics, which demonstrates the effectiveness of considering varying connection strengths brought by the different relations. Then, we attempt to separately learn the community partition at each timestamp, and according to the result, we can see that it also fails to outperform our original method. This may be because without considering the historical community partition, the change in community partition over time will be steep, which leads to a reduction in the generalization ability. Furthermore, we notice that the performance improvement that on the ICEWS14 dataset is smaller than on the GDELT dataset, especially in *Hit@10*. We believe this is because the ICEWS14 dataset is sparser, so the community structure is not significant, which limits the advantage of the global structure evolution modeling.



(a) HyTE

(b) TeMP

(c) *EvoExplore*

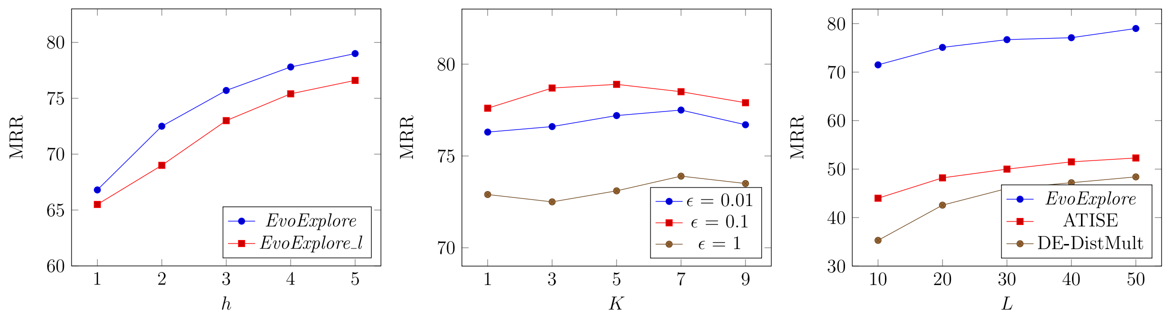
Fig. 4. Representation visualization results of different methods on the ICEWS14 dataset.

5.9. Visual analysis

To verify the ability of the different methods to capture the evolutionary nature of the temporal knowledge graphs, we choose a time-sensitive entity (i.e., "Benjamin Netanyahu") and a time-insensitive entity (i.e., "China"), and obtain the representations of the entities at different timestamps. Then, we employ t-SNE [48] to project these representations into a 2-dimensional space. Fig. 4 gives the visualization results of the representations obtained by HyTE, TeMP, and *EvoExplore*, where the red dots denote the time-sensitive entity and the blue dots denote the time-insensitive entity. First, we notice that HyTE fails to distinguish between the different entities and one entity in different periods. This is because HyTE learns representations for each timestamp separately and thus ignores the correlations among the different timestamps. TeMP can form natural clusters based on the timestamps, and the time-insensitive entity has a more compact cluster, since the semantics of the time-insensitive entity will change slightly over time. However, TeMP simplifies the evolution of a TKG as a series of static snapshots, and hence the obtained representations fail to reveal the evolution process of the entity. Finally, it is not hard to find that the representations obtained by our method can form natural clusters and have obvious evolution trends with

Table 6
Performance of different variants of *EvoExplore* for link prediction.

Dataset	ICEWS 14			GDELT		
Variants	*MRR*	*Hit@1*	*Hit@10*	*MRR*	*Hit@1*	*Hit@10*
$\mathbf{u}_e^{\tau} = sin(\boldsymbol{\theta}_e \cdot \tau)$	69.1	62.7	83.3	47.3	31.5	65.3
$\mathbf{u}_e^{\tau} = tanh(\boldsymbol{\theta}_e \cdot \tau)$	67.2	59.8	83.5	46.5	30.6	65.9
Static entity representation	64.7	57.3	80.8	44.2	28.6	61.4
Only using entity-level attention	60.5	53.6	77.9	41.7	26.4	59.2
Only using relation-level attention	58.1	51.9	71.4	39.4	24.1	58.7
Without attention mechanism	55.4	48.5	67.6	37.6	23.3	55.0
Without soft modularity matrix	70.3	63.1	82.4	48.4	33.8	63.7
Separately treating each timestamp	70.8	63.4	81.7	48.6	34.0	62.5
EvoExplore	72.5	65.3	85.2	51.4	35.3	74.8



(a) Length of history (b) Number of communities (c) Number of negative samplings

Fig. 5. Effects of different parameters on the link prediction performance on the ICEWS05-15 dataset.

the development of time, while the time-insensitive entity has a more compact cluster. This demonstrates that *EvoExplore* can effectively capture the evolutionary nature of a TKG.

5.10. Parameter analysis

Here, we turn to investigating whether the parameters of *EvoExplore* impact the performance of the link prediction task. As shown in Fig. 5(a), we first study an important parameter named history length h, which determines how many historical events are considered during a local structure evolution modeling. It can be seen that with the increase in history length, the performance of both *EvoExplore* and *EvoExplore_l* on *MRR* increases consistently. Furthermore, we notice that when h continues to increase, the performance improvement decreases. To strike a balance between the performance and complexity, we set h as a middle value.

We further study the impact of community number K and trade-off parameter ϵ. As shown in Fig. 5(b), we notice that with different ϵ values, the performance of *EvoExplore* increases first and then decreases as community number K increases, which indicates that the number of communities influences the performance of our method. Furthermore, we notice that both a too large ϵ and a too small ϵ lead to performance decrease. This may be because a too small ϵ makes *EvoExplore* fail to adequately capture the global structure evolution, but a too large ϵ makes *EvoExplore* have difficulty in focusing on the fine-grained relation establishment process of a TKG.

Finally, we study the impact of negative sampling number L on the model performance. From Fig. 5(c), we can see that the performance of each model improves with the increase in the number of negative samplings, and when L is large, the improvement is small. Furthermore, we observe that *EvoExplore* still has a good performance when L is small, which demonstrates that *EvoExplore* is highly efficient and does not depended on the large number of negative samplings.

6. Conclusion and future work

In this paper, we make the first attempt to incorporate both the local and global structure evolutions into a temporal knowledge graph representation. A novel framework *EvoExplore* is proposed to learn representations for TKGs by modeling the mutual influence of the local and global structure evolutions. For the local structure, it employs a hierarchical-attention-based temporal point process to model the establishment of each relation, which can capture the fine-grained evolution process of the graph structure. For the global structure, a component based on soft modularity is designed to model the time-evolving community partition, which helps to impose constraints on the learned representations by leveraging the structural dependencies. Finally, by jointly optimizing the above two parts, the obtained representations can effectively capture the mutual influence of the local and global structure evolutions. The experimental results demonstrate the superiority of *EvoExplore* when compared with a series of baseline methods. One interesting future direction of this work is to incorporate the information of the time-aware attributes into our framework.

CRediT authorship contribution statement

Jiasheng Zhang: Conceptualization, Methodology, Writing – original draft preparation. **Shuang Liang:** Investigation, Validation. **Yongpan Sheng:** Conceptualization, Methodology. **Jie Shao:** Supervision, Writing – reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61832001), Shenzhen Science and Technology Program (No. JCYJ20210324121213037) and CAAI-Huawei MindSpore Open Fund. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] K.D. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008, 2008, pp. 1247–1250.
[2] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: A core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, 2007, pp. 697–706.
[3] L. Zhang, Z. Kang, X. Sun, H. Sun, B. Zhang, D. Pu, Kcrec: Knowledge-aware representation graph convolutional network for recommendation, Knowl. Based Syst. 230 (2021) 107399.
[4] Z. Deng, C. Li, S. Liu, W. Ali, J. Shao, Knowledge-aware group representation learning for group recommendation, in: 37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021, 2021, pp. 1571–1582.
[5] M. Bakhshi, M. Nematbakhsh, M. Mohsenzadeh, A.M. Rahmani, Sparseqa: Sequential word reordering and parsing for answering complex natural language questions over knowledge graphs, Knowl. Based Syst. 235 (2022) 107626.
[6] A. Zeb, A.U. Haq, J. Chen, Z. Lei, D. Zhang, Learning hyperbolic attention-based embeddings for link prediction in knowledge graphs, Knowl. Based Syst. 229 (2021) 107369.
[7] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, M. Ward, ICEWS Coded Event Data, Harvard Dataverse, 2015.
[8] S.S. Dasgupta, S.N. Ray, P.P. Talukdar, Hyte: Hyperplane-based temporally aware knowledge graph embedding, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 2001–2011.
[9] T. Lacroix, G. Obozinski, N. Usunier, Tensor decompositions for temporal knowledge base completion, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.
[10] R. Goel, S.M. Kazemi, M. Brubaker, P. Poupart, Diachronic embedding for temporal knowledge graph completion, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020, 2020, pp. 3988–3995.
[11] C. Xu, M. Nayyeri, F. Alkhoury, H.S. Yazdi, J. Lehmann, Temporal knowledge graph completion based on time series Gaussian embedding, in: The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I, 2020, pp. 654–671.
[12] J. Wu, M. Cao, J.C.K. Cheung, W.L. Hamilton, Temp: Temporal message passing for temporal knowledge graph completion, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, 2020, pp. 5730–5746.
[13] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 2787–2795.
[14] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, QuÉBec City, QuÉBec, Canada, 2014, pp. 1112–1119.
[15] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, 2015, pp. 2181–2187.
[16] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, 2015, pp. 687–696.
[17] H. Xiao, M. Huang, Y. Hao, X. Zhu, TransA: An adaptive approach for knowledge graph embedding, 2015, CoRR, arXiv:1509.05490.
[18] S. He, K. Liu, G. Ji, J. Zhao, Learning to represent knowledge graphs with Gaussian embedding, in: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, 2015, pp. 623–632.
[19] Z. Sun, Z. Deng, J. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, 2019.
[20] Z. Li, X. Liu, X. Wang, P. Liu, Y. Shen, TransO: A knowledge-driven representation learning method with ontology information constraints, World Wide Web (2022) http://dx.doi.org/10.1007/s11280-022-01016-3.
[21] M. Nickel, V. Tresp, H. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, 2011, pp. 809–816.
[22] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
[23] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, 2016, pp. 2071–2080.
[24] I. Balazevic, C. Allen, T.M. Hospedales, Tucker: Tensor factorization for knowledge graph completion, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, 2019, pp. 5184–5193.
[25] S.M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, MontrÉAl, Canada, 2018, pp. 4289–4300.
[26] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D knowledge graph embeddings, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 1811–1818.
[27] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen, D.Q. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), 2018, pp. 327–333.
[28] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, B. Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, 2019, pp. 3060–3067.
[29] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, 2018, pp. 593–607.
[30] D. Nathani, J. Chauhan, C. Sharma, M. Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 2019, pp. 4710–4723.
[31] S. Vashishth, S. Sanyal, V. Nitin, P.P. Talukdar, Composition-based multi-relational graph convolutional networks, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.
[32] J. Leblay, M.W. Chekol, Deriving validity time in knowledge graph, in: Companion of the the Web Conference 2018 on the Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018, 2018, pp. 1771–1776.
[33] A. García-Durán, S. Dumancic, M. Niepert, Learning sequence encoders for temporal knowledge graph completion, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 4816–4821.
[34] R. Trivedi, H. Dai, Y. Wang, L. Song, Know-evolve: Deep temporal reasoning for dynamic knowledge graphs, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 3462–3471.
[35] Z. Han, P. Chen, Y. Ma, V. Tresp, Dyernie: Dynamic evolution of Riemannian manifold embeddings for temporal knowledge graph completion, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, 2020, pp. 7301–7316.

[36] W. Jin, M. Qu, X. Jin, X. Ren, Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, 2020, pp. 6669–6683.
[37] C. Zhu, M. Chen, C. Fan, G. Cheng, Y. Zhang, Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021, 2021, pp. 4732–4740.
[38] Z. Han, P. Chen, Y. Ma, V. Tresp, Explainable subgraph reasoning for forecasting on temporal knowledge graphs, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021.
[39] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, X. Cheng, Temporal knowledge graph reasoning based on evolutional representation learning, in: SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, 2021, pp. 408–417.
[40] H. Sun, J. Zhong, Y. Ma, Z. Han, K. He, Timetraveler: Reinforcement learning for temporal knowledge graph forecasting, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, 2021, pp. 8306–8319.
[41] Z. Han, Y. Ma, Y. Wang, S. Günnemann, V. Tresp, Graph hawkes neural network for forecasting on temporal knowledge graphs, in: Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020, 2020.
[42] A.G. Hawkes, Spectra of some self-exciting and mutually exciting point processes, Biometrika 58 (1) (1971) 83–90.
[43] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.
[44] H. Sun, F. He, J. Huang, Y. Sun, Y. Li, C. Wang, L. He, Z. Sun, X. Jia, Network embedding for community detection in attributed networks, ACM Trans. Knowl. Discov. Data 14 (3) (2020) 36:1–36:25.
[45] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 3111–3119.
[46] MindSpore, https://www.mindspore.cn/.
[47] K. Leetaru, P.A. Schrodt, Global database of events, language and tone, in: ISA Annual Convention, 2013.
[48] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605.