TKGAT: Temporal Knowledge Graph Representation Learning Using Attention Network

Shaowei Zhang, Zhao Li, Xin Wang(✉), Zirui Chen, and WenBin Guo

College of Intelligence and Computing, Tianjin University, Tianjin, China
{zhangsw,lizh,wangx,zrchen,wenff}@tju.edu.cn

Abstract. Temporal knowledge graph representation learning models can capture more comprehensive semantic information, which has higher practical application value and gradually attracts wide attention. However, the existing temporal knowledge graph representation learning models usually have challenges in encoding temporal information and capturing rich structural information. In this paper, we propose a novel temporal knowledge graph representation learning model, named TKGAT, which is based on graph neural networks using Bochner's theorem to design time encoding function that can flexibly learn relative time information. Furthermore, attention network is adopted to model different relations features and the self-attention mechanism is optimized by the decoupled attention method, so that the attention weight matrix incorporates more extensive temporal and structural information and learns the correlations between entity and temporal features. The extensive experiments have shown that the proposed model can consistently outperform state-of-the-art models over all benchmark datasets.

Keywords: temporal knowledge graph · representation learning · decoupled attention

1 Introduction

A great amount of data generated in daily life often takes the form of graph structure, such as social networks, financial transactions and literature citations. Researchers have adopted the form of triple $(subject, relation, object)$ to represent semantic information in data, and construct large-scale knowledge graphs (KG) such as DBpedia, FreeBase, and WordNet [25]. However, the KGs are usually incomplete due to data sparsity, which makes knowledge graph completion (KGC) a priority task. Knowledge graph representation learning expresses underlying semantic information by mapping the triples into continuous low-dimensional vector spaces, which is proved to be an efficient method for KGC [11].

S. Zhang, Z. Li—Contributed equally to this research.

X. Yang et al. (Eds.): ADMA 2023, LNAI 14177, pp. 46–61, 2023.
https://doi.org/10.1007/978-3-031-46664-9_4

Static KG representation learning models that ignore the temporal information, which can lead to an inaccurate semantic representation. As depicted in Fig. 1 (a), there are three relations *Praise or endorse*, *Make optimistic comment* and *Criticize or denounce* between *Barack Obama* and *Iran*, such knowledge

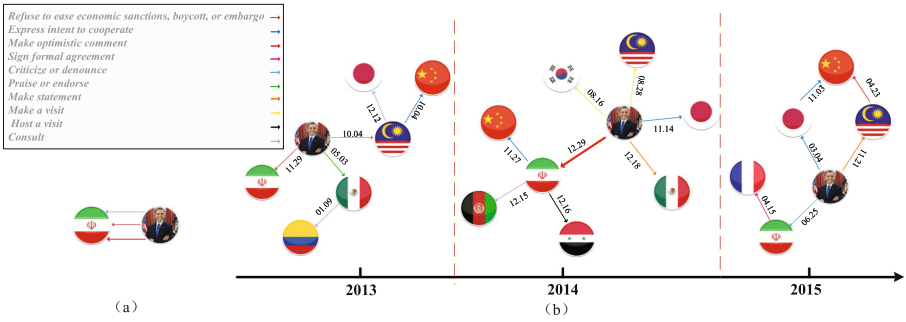

Fig. 1. Example of the temporal knowledge graph

can cause confusion when temporal information is neglected since these three relations are in conflict. Figure 1 (b) depicts a sample of the temporal knowledge graph (TKG), the relations between *Barack Obama* and *Iran* made clarity as the temporal information has been added. We can also observe that *Iran* has an *Express intent to cooperate* with *China*, *Consult* with *Afghanistan* and *Host a visit* with *Syria*, these three relation types will have various impacts on *Iran*, and the topology of countries and relations around *Iran* also determines the character of *Iran*. Therefore, effectively modeling the topological features of KG is essential for KG representation learning. Besides, capturing temporal features in TKG is also crucial. As shown in Fig. 1 (b), the relation *Make a visit* between *Barack Obama* and *South Korea* occurred at time *2014-08-16*, however, *Barack Obama* has an relation *Make optimistic comment* with *Iran* at time *2014-12-29*, since the long time interval between the two events, the former will have less influence on the latter as time passes, which also reveals that more significant temporal characteristics are typically provided by the relative time. Our model aims to well capture the topological and temporal features in TKG, in contrast to the static KG representation learning models, which ignore temporal information and process the TKG directly in a static manner, resulting in incomplete and inaccurate expression of semantic information.

In recent years, TKG representation learning has received extensive attention from both academia and industry [7], which incorporates the corresponding temporal features when expressing the semantic information in data. However, most of the current TKG representation learning models usually face many challenges. (1) The sensible time encoding, since the TKG topology is dynamic, entities should have various features at different times. Besides, time encoding should satisfy the inherent properties of time, such as the relative time can usually carry more meaningful information than absolute time, for example, when a

user buys a product on the internet, the temporal information of browsing and staying on a certain product is more important than the order of browsing the products. However, the previous models mostly used simple feed-forward neural networks or recurrent neural networks to capture temporal features, which lack of in-depth theory; (2) Modeling relations appropriately, distinct relations around an entity should have different influences on current one, most of existing models fail to take into account relation attention. Topological information incomplete when various relations are addressed with the same attention weights; (3) Effectively modeling structure, the most TKG representation learning models extend on those in static KG, which focus more attention on quadruples inherent characteristics and treat the quadruples independently while ignoring structural information, and the model should also capture correlations between entity intrinsic features and temporal features when modeling structure, which is still challenging.

A TKG attention networks, named TKGAT, is proposed to solve the common problems in existing TKG representation learning models. The ***time encoding function*** based on the Bochner's theorem [23] has been adopted to capture temporal features, which is well suited to model the properties of relative time and has a deep theoretical foundation. The weights of the different relation types are constructed by the attention network to reflect the relevant to central entity. The self-attention mechanism [19] has proved its powerful ability in various tasks, the position encoding is replaced by time encoding and ***decoupled attention*** [6] is applied to optimize self-attention, which can incorporate more extensive knowledge graph features and effectively capture the correlations between entity and time. Our contributions in this paper can be summarized as follows.

(1) We propose a novel temporal knowledge graph representation learning model, TKGAT, which encodes temporal information based on Bochner's theorem and uses attention networks to capture different relations weight in order to efficiently model relational information and improve model performance.
(2) By separating structure and time encoding to optimize the traditional self-attention mechanism, a decoupled attention approach is designed, which combines graph neural networks to efficiently capture correlations between entity and temporal features.
(3) The model proposed in this paper achieves the best experimental results on three public datasets, further demonstrating the effectiveness of the model and outperforming baseline methods.

The rest of this paper is organized as follows. Section 2 presents related works. We introduce preliminaries in Sect. 3. We describe the proposed model in detail in Sect. 4. Section 5 reports the experimental results, and we conclude in Sect. 6.

2 Related Work

In this section, the traditional static KG representation learning models and the TKG representation learning models are introduced.

2.1 Static Knowledge Graph Representation Learning

At present, most of the existing knowledge graph representation learning models are suitable for static KG, which can be classified into three categories. The first category is the translation-based model, which makes the head and tail entities satisfy the translation constraints of the relation, and measure the truth of the triples by calculating the Euclidean distance between the head and tail entity vectors after the translation. TransE [1], TransH [20], and TransR [13] are the most representative models, since the simple and efficient nature of TransE, there are a series of subsequent works that extended on TransE. The second category is the semantic matching based model, which evaluates the plausibility of a fact by matching the underlying semantic information of entities and relations in the vector space. RESCAL [15], DistMult [24], ComplEx [18], and SimplE [9] are the simplest and most widely used models. The third category is neural network-based model, which mainly takes advantage of the excellence of neural networks in feature extraction and non-linear fitting to model KG features, representative models include ConvE [3], ConvKB [14], and RGCN [16]. However, all these models ignore the temporal information and fail to reflect the real-world change properties, resulting in lower accuracy in TKG.

2.2 Temporal Knowledge Graph Representation Learning

In recent years, temporal knowledge graph representation learning has gradually become a hot research topic. Most existing models primarily focus on extending static KG representation learning to TKG. TTransE [7] adds temporal information to the score function of the TransE and makes it satisfy the temporal information based translation constraint. HyTE [2] extends the TransH model, which projects entities and relations to a time-specific hyperplane to realize the embedding of temporal information. TA-TransE [4] represents the relation type and temporal information as a sequence of characters, then uses the LSTM to learn the time-aware representation of relation types. TComplEx [10] extends ComplEx and considers the score of each quadruple as fourth-order tensor decomposition. TeRo [21] borrows ideas from TransE and RotatE [17], which defines the temporal evolution of entity embedding as a rotation and regards relation as translation. ATiSE [22] incorporates temporal information into entity and relation representations by using additive time series decomposition and uses a multi-dimensional Gaussian distribution to represent temporal uncertainty. Inspired by diachronic word embedding, DE-SimplE [5] incorporates temporal information into diachronic entity embedding and has the capability of modeling various relation patterns. Compared to our model, these models fail to capture the rich structural information and the correlations between entity and temporal features. Another line of work on TKG representation learning employs neural networks, RE-NET [8] adopts a R-GCN based aggregator and recurrent event encoder to model the historical information. RE-GCN [12] learns the evolutional representations of entities and relations by capturing the structural dependencies and sequential patterns. However, those models focus on TKGC extrapolation

task, i.e., inferring the feature facts in a sequence, which are fundamentally different from our work.

3 Preliminaries

In this section, we present the preliminaries of our work, including the definition of temporal knowledge graph and graph neural network.

3.1 Temporal Knowledge Graph

In this paper, we represent a temporal knowledge graph as $\mathcal{G} = \{(s, r, o, t)\} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V} \times \mathcal{T}$, where $\mathcal{V}$, $\mathcal{R}$ and $\mathcal{T}$ indicate the sets of nodes, edges, and timestamps, respectively. Temporal knowledge graph completion (TKGC) is to solve the problem of incompleteness in TKG. Assume that the whole true facts set is $\mathcal{F} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V} \times \mathcal{T}$, TKG should be a subset of the whole true facts set since the incompleteness of TKG, i.e., $\mathcal{G} \subseteq \mathcal{F}$. TKGC is the reasoning from $\mathcal{G}$ to $\mathcal{F}$. According to the time range, TKGC has two settings, interpolation and extrapolation. Given a temporal knowledge graph $\mathcal{G}$ with timestamps t range from t_1 to t_T, for the interpolation setting, TKGC predicts missing fatcs with $t_1<t<t_T$; In contrast, for the extrapolation setting, TKGC predicts missing fatcs with $t>t_T$, i.e., predicting future facts based on past ones. More formally, the purpose of TKGC is to predict either the subject in a given query $(?, r, o, t)$ or the object in a given query $(s, r, ?, t)$. Our work is focus on the TKGC for the interpolation settings.

3.2 Graph Neural Network

Graph neural network (GNN) enjoys several advantages such as the ability to effectively handle non-Euclidean data, which makes it a great success in processing graph data. The core idea of GNN is the message propagation mechanism, i.e., the central node features are constructed by aggregating information from neighbors. In order to obtain the features of the central node i through multiple layers of GNN, each GNN layer will implement the following two steps: (1) Message Propagation, get messages from all neighbors of node i; (2) Message Aggregation, aggregate messages from all neighbor nodes then combines with the features of node i in the previous layer to obtain the features in the current layer. The above processes are defined as follows:

$$\mathbf{h}^l_{\mathcal{N}_i^k} \leftarrow AGG\left(\left\{\mathbf{h}_j^{l-1}, \forall j \in \mathcal{N}_i^k\right\}\right) \tag{1}$$

$$\mathbf{h}_i^l \leftarrow \sigma \mathbf{W}^l \left(\mathbf{h}_i^{l-1} \,||\, \mathbf{h}^l_{\mathcal{N}_i^k}\right) \tag{2}$$

Steps (1) and (2) correspond to the Eqs. 1 and 2, respectively. Where $\mathcal{N}_i^k$ denotes the k neighbors of node i, h_i^l denotes the hidden layer state of node i at l-th layer, and AGG is a specific function for aggregating the features of neighbors, which

can be implemented using long short term memory (LSTM), self-attention mechanisms, etc. In this paper, we use a decoupled attention approach to implement *AGG*, which is able to capture more extensive features. The representative GNN models include graph convolutional networks (GCN) and graph attention networks (GAT), both of which assign weights to neighbors explicitly or implicitly during the aggregating features.

4 Our Approach

The Fig. 2 depicts the architecture of our model. Overall, the model is based on the encoder-decoder architecture. The encoder module maps entities into a continuous low-dimensional vector space and incorporates structural and temporal features simultaneously. In view of the fact that the relations are usually irrelevant to the temporal information, the temporal features are integrated into the vector of the entity in our model. Since the different relation types have different impacts on subject, the encoder module first integrates the relation features into the objects according to the type attention weights, then employs a decoupled attention method to learn the interactions between the subjects and objects in terms of structure and time. Finally, the quadruple based (s, r, o, t) is converted into the triple (s_t, r, o_t), decoder module can directly evaluate triples using the static KG embedding methods.

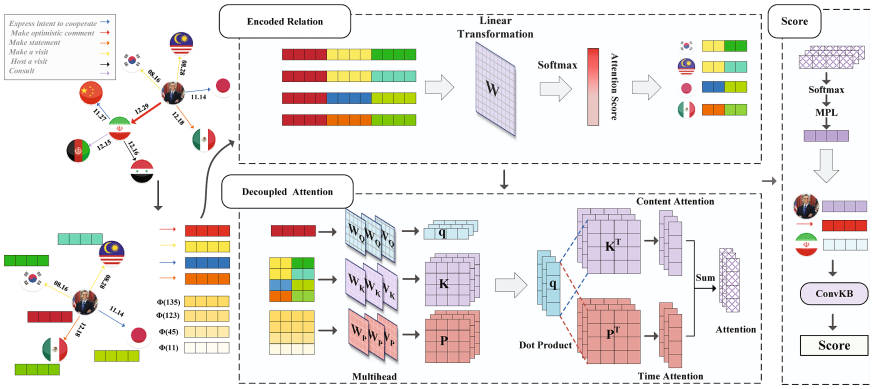

Fig. 2. The architecture of the TKGAT model. In this figure, in order to evaluate the truth of the quadruple (*Barack Obama*, *Make optimistic comment*, *Iran*, *2014-12-29*). Firstly, we find the temporal neighbors where the interaction time with *Barack Obama* before *2014-12-29*, encoded relation module combines the vectors of the subject *Barack Obama*, relations and temporal neighbors together to calculate attention weights and integrates the relation features into the temporal neighbors. Secondly, time encoding function based on Bochner's Theorem is applied to capture relative time features. Thirdly, decoupled attention module learns vector of *Barack Obama* by capturing the structural and temporal feature, an analogous approach is used for *Iran*. Finally, static KGs embedding model ConvKB is adopted to evaluate score of triple that integrated temporal features.

4.1 Encoded Relation Information

Assume that there are $|\mathcal{R}|$ relation types and $|\mathcal{V}|$ entities in the temporal knowledge graph $\mathcal{G}$, the initial vectors of all entities and relations are represented as sets $\mathrm{E} = \{\mathbf{e}_i\}_{i=1}^{|\mathcal{V}|}$ and $\mathrm{R} = \{\mathbf{r}_i\}_{i=1}^{|\mathcal{R}|}$ respectively, where $\mathbf{e}_i \in \mathbb{R}^{d_e}$ represents the initial vector of i-th entity and $\mathbf{r}_i \in \mathbb{R}^{d_r}$ represents the initial vector of i-th relation, d_e and d_e represent the initial vectors dimension of entity and relation respectively. Given a quadruple (s, r, o, t), according to the inherent characteristics of time, i.e., information about future events cannot influence the ones of the present moment, the temporal neighbors of subject s are denoted as $\mathcal{N}_s^{t_k<t} = \{(r_i, o_j, t_k) | (s, r_i, o_j, t_k) \in \mathcal{G}, t_k < t\}$. Since various relation types have different effects on the subject, we combine the subject vector $\mathbf{e}_s$, relation vector $\mathbf{r}_i$, and the object vector $\mathbf{e}_j$ together and calculate the attention weights by the sofmax function. Finally, the relation feature is incorporated into the corresponding object vector, where the attention weights are calculated as follows.

$$\mathbf{u}_{r_i,o_j} = \mathbf{W}_1 \left(\mathbf{e}_s \,||\, \mathbf{r}_i \,||\, \mathbf{e}_j\right) \tag{3}$$

$$\alpha_{i,j} = \mathrm{softmax}\left(\mathbf{u}_{r_i,o_j}\right) = \frac{\exp\left(\sigma\left(\mathbf{p} \cdot \mathbf{u}_{\mathrm{r_i,o_j}}\right)\right)}{\sum_{(r_m,o_n,t_i)\in\mathcal{N}_s^{t_i<t}} \exp\left(\sigma\left(\mathbf{p} \cdot \mathbf{u}_{r_m,o_n}\right)\right)} \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_e \times (2d_e + d_r)}$, $\mathbf{p} \in \mathbb{R}^{d_e}$ are parameters learned during the model training, σ employs the LeakyReLU activation function. After obtaining the attention weights $\alpha_{i,j}$ of the relation type, the temporal neighbors vectors that incorporated relation types features are calculated as follows:

$$\mathbf{x}_{i,j} = \alpha_{i,j} \mathbf{W}_2 \left(\mathbf{r}_i \,||\, \mathbf{e}_j\right) \tag{5}$$

where $\mathbf{W}_2 \in \mathbb{R}^{d_e \times (d_e + d_r)}$ is model parameter matrix.

4.2 Encoded Temporal Information

Having obtained the vectors of entitits that incorporated the relations information, our aim is to further integrate the temporal information. Since the TKG's structure are no longer static and the entity features may change, the time encoding should be able to show temporal characteristics, e.g. the events that happened a long time ago have less impact on the current events. We employ the time encoding function mapping from the time domain to the continuous differentiable functional domain proposed by literature [23], which is based on Bochner's Theorem and can be compatible with gradient descent in model training, we denoted it as $\Phi(t)$ and the definition as follows:

$$t \to \Phi(t) := \sqrt{\frac{1}{d_t}} \left[\cos(\omega_1 t), \sin(\omega_1 t), ..., \cos(\omega_n t), \sin(\omega_n t)\right] \tag{6}$$

where $\omega = [\omega_1, ..., \omega_{d_t}]^T$ are learnable parameters.

4.3 Encoded Structural Information

Since the topology of the TKG contains important information, we borrow the core idea of GNN, i.e., using message propagation mechanism to capture the structural information. In order to aggregate the messages from neighbors coupled with attention weights, we adopt the decoupled attention method based on self-attention mechanism.

Given a quadruple (s, r, o, t), the temporal neighbors of subject s are $\mathcal{N}_s^{t_k<t}$. At time t, the vector of the subject s at layer l-th is represented as $\mathbf{h}^l$, when $l = 1$, $\mathbf{h}^l = \mathbf{e}_s$, i.e., the initial vector of s. The subject s corresponding object under relation r_j is o_i, and its vector at lth layer is represented as $\mathbf{h}_i^l$, when $l = 1$, $\mathbf{h}_i^l = \mathbf{x}_{i,j}$, which is obtained by the encoded relation module. Since the relative time, rather than absolute time, usually reveals critical temporal information, we directly encode the relative time $\{t - t_1, t - t_2, .., t - t_k\}$ using the time encoding function, then we obtain the temporal encoding of neighbors $\{\Phi(t - t_1), \Phi(t - t_2), ..., \Phi(t - t_k)\}$, where k denotes the number of neighbors of s at time t.

The traditional self-attention mechanism are used to process sequence structure, which add or combine the two vectors that are used to represent the content and position information of the token to construct its feature. However, this approach can't effectively capture the correlation between content and position features. Inspired by DeBERTa [6], we apply time encoding to replace position encoding and calculate the weights by decoupled attention method.

The query vector at layer l is $\mathbf{q} = \mathbf{W}_q\mathbf{h}^{l-1}$,$\mathbf{W}_q \in \mathbb{R}^{d_h \times d_e}$ is the model parameter matrix, the vector of temporal neighbours and temporal encoding are constructed as matrices $\mathbf{Z}_E$ and $\mathbf{Z}_T$ respectively, which are represented at the $l-1$ layer as:

$$\mathbf{Z}_E = \left[\mathbf{h}_1^{(l-1)}, \mathbf{h}_2^{(l-1)}, ..., \mathbf{h}_k^{(l-1)}\right] \in \mathbb{R}^{d_e \times k} \tag{7}$$

$$\mathbf{Z}_T = [\Phi(t - t_1), \Phi(t - t_2), ..., \Phi(t - t_k)] \in \mathbb{R}^{d_t \times k} \tag{8}$$

Applying linear transformation on matrices $\mathbf{Z}_E$ and $\mathbf{Z}_T$:

$$\mathbf{K} = \mathbf{W}_K\mathbf{Z}_E, \mathbf{P} = \mathbf{W}_T\mathbf{Z}_T, \mathbf{V} = \mathbf{W}_V\mathbf{Z}_E \tag{9}$$

where $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_h \times d_e}$, $\mathbf{W}_T \in \mathbb{R}^{d_h \times d_t}$ are model parameters, the attention matrix obtained by the decoupled attention approach as following:

$$\widetilde{\mathbf{A}}_{0,j} = [\mathbf{q}]^\top \mathbf{K}_j + [\mathbf{q}]^\top \mathbf{P}_j \tag{10}$$

the attention matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{1 \times k}$, where $\mathbf{K}_j$ and $\mathbf{P}_j$ denote the j-th column of the matrix $\mathbf{K}$ and $\mathbf{P}$ respectively. In the process of calculating attention, $[\mathbf{q}]^\top \mathbf{K}_j$ is used to capture the correlation between the subject s and the j-th neighbour object in terms of structure, and $[\mathbf{q}]^\top \mathbf{P}_j$ is used to capture the correlation between the subject s and the j-th neighbour object in terms of time, the final attention matrix is obtained by adding the two above. We apply

the softmax function to get the weights, then the final feature vector of temporal neighbors is obtained by weighted sum.

$$\mathbf{h}^{l}_{\mathcal{N}_s^{<t}} = \text{softmax}\left(\frac{\widetilde{\mathbf{A}}_{0,j}}{\sqrt{2d_h}}\right)\mathbf{V} \tag{11}$$

In order to maintain the original features of the subject s, we concatenate the final feature vector of temporal neighbors with the s hidden vector at $(l-1)$-th layer, then pass it to a multilayer perceptron to capture non-linear interactions.

$$\begin{gathered}\mathbf{h}^{l} = \text{MPL}\left(\mathbf{h}^{l}_{\mathcal{N}_s^{\tau_k<\tau}} \,||\, \mathbf{h}^{l-1}\right) = \text{ReLU}\left(\left[\mathbf{h}^{l}_{\mathcal{N}_s^{\tau_k<\tau}} \,||\, \mathbf{h}^{l-1}\right]\mathbf{W}_0^{l} + \mathbf{b}_0^{l}\right)\mathbf{W}_1^{l} + \mathbf{b}_1^{l} \\ \mathbf{W}_0^{l} \in \mathbb{R}^{2d_h \times d_h}, \mathbf{b}_0^{l} \in \mathbb{R}^{d_h}, \mathbf{W}_1^{l} \in \mathbb{R}^{d_h \times d_o}, \mathbf{b}_1^{l} \in \mathbb{R}^{d_o}\end{gathered} \tag{12}$$

where $\mathbf{W}_0^{l}$, $\mathbf{b}_0^{l}$, $\mathbf{W}_1^{l}$ and $\mathbf{b}_1^{l}$ are model parameters, d_o denotes the dimension of the final output vector. We also show that the proposed model can be easily extended to the multi-head setting which can improve performance and stability. Suppose there are m different head, and $\text{head}^{(i)} = \mathbf{h}^{l(i)}_{\mathcal{N}_s^{t_k<t}}$, we concatenate the m head outputs with s and then carry out the same procedure as Eq. 12.

$$\widetilde{\mathbf{h}}^{l} = \text{MPL}\left(\text{head}^{(1)} \,||\, , ..., || \,\text{head}^{(m)} \,||\, \mathbf{h}^{l-1}\right) \tag{13}$$

4.4 Decoder and Training

Given a quadruple $\eta = (s, r, o, t)$, the encoder module of the TKGAT provides vectors with temporal information $(\tilde{\mathbf{s}_t}, \mathbf{r}, \tilde{\mathbf{o}_t})$. Since the temporal information has been incorporated into the entity vector, the static KG model score function can be used to evaluate the triples. Among the currently existing methods, TKGAT adopts ConvKB as the decoder, the score function defined as following:

$$f\left(\eta\right) = \left(\Big\|_{n=1}^{|\Omega|} g\left(\left[\mathbf{s}_t, \mathbf{r}, \mathbf{o}_t\right] * \omega^{n}\right)\right)\mathbf{W} \tag{14}$$

where Ω denotes the set of convolution kernels, ω^{n} denotes the n-th convolution kernel, and $\omega \in \Omega$. $\mathbf{W}_c$ denotes the parameters matrix of the linear transformation, Ω and $\mathbf{W}_c$ share parameters during the model training, the activation function $g(\cdot)$ employs ReLU, $*$ denotes the convolution operation. The output vectors of the $|\Omega|$ convolution operations are concatenated into a single vector, then linear transformation is applied to obtain the final score.

During the model training, the parameters of are learned using gradient-based optimization in mini-batches. For each quadruple $\eta = (s, r, o, t) \in \mathcal{G}$, we sample a negative set of entities $S = \{o' | (s, r, o', t) \notin \mathcal{G}\}$, then the cross-entropy loss function is used to train the model, which defined as follows:

$$\mathcal{L} = -\sum_{\eta \in \mathcal{G}} \frac{\exp\left(f(s, r, o, t)\right)}{\exp\left(\sum_{o' \notin \mathcal{G}} f(s, r, o', t)\right)} \tag{15}$$

Note that, without losing generality, we used the above loss and negative samples for subject queries. The algorithm 1 shows the training process in detail.

Algorithm 1: TKGAT training algorithm

Input: Temporal knowledge graph $\mathcal{G}$, initialization vector dimension for entity, relation, and timestamp d_e, d_r, and d_t, number of negative samples n, number of iterative rounds n_{iter}, number of batches n_b, batch size m_b

Output: Vector representation of entities, vector representation of relations

Initialize the vector of entity $\mathbf{e}_i$ with $N\left(0, \frac{1}{d_e}\right)$;

Initialize the vector of relation $\mathbf{r}_i$ with $N\left(0, \frac{1}{d_r}\right)$;

for $n = 1, ..., n_{iter}$ **do**

  for $i = 1, ..., n_b$ **do**

    $\mathcal{D}_{batch} \leftarrow Sample(\mathcal{D}_{train}, m_b)$;

    `// Sample` m_b `instances from training set`

    for $(s, r, o, t) \in \mathcal{D}_{batch}$ **do**

      $\mathcal{D}_{batch} \leftarrow \mathcal{D'}_{train} \cup \{s', r, o', t\}$;

      `// Negative samples by replacing the subject and object`

      $\mathbf{x}_{i,j} \leftarrow \alpha_{i,j}\mathbf{W}_2 \left(\mathbf{r}_i \,||\, \mathbf{e}_j\right)$;

      `// Encoded relation information according to Equation 5`

      $\Phi(t - t_i) \leftarrow$ relative time encoding according to Equation 6;

      $\widetilde{\mathbf{h}} \leftarrow$ vector of entity according to Equation 10, 11, 13 ;

    end

    Training the model according to the Equation 14, 15 ;

  end

end

5 Experiments

In this section, to verify the effectiveness of the proposed model, we conduct experiments on link prediction tasks on three public datasets. We first introduce the experimental setup, including datasets, evaluation metrics, baselines, and implementation, and then analyze the experimental results. Furthermore, we perform several ablation studies to demonstrate the effectiveness of each main component of the proposed model.

5.1 Experimental Setup

Datasets. We evaluate our proposed models on the link prediction tasks, and three public TKGs datasets are used in our experiments. The statistics of the datasets are summarised in Table 1. For the Integrated Crisis Early Warning System (ICEWS) dataset, we use two subsets provided by [4]: ICEWS14, corresponding to facts in 2014, and ICEWS05-15, corresponding to facts between 2005 and 2015. For the Global Database of Events, Language, and Tone (GDELT) dataset, we use subsets which corresponding to facts from 1 April 2015 to 31 March 2016, each piece of data has a corresponding timestamp. We use the same splits of training, validation, and testing sets as provided by [5].

Evaluation Metrics. For each quadruple $(s, r, o, t) \in \mathcal{D}_{test}$, where $\mathcal{D}_{test}$ represents the test dataset, we generate two queries: $(s, r, ?, t)$ and $(?, r, o, t)$. For the first query, the model evaluates all entities and obtains scores $f(s, r, o', t)$, $\forall o' \in \mathcal{E}$, with an analogous approach used for the second query. According to the final scores, the rank of the given quadruple is obtained, and we report *mean reciprocal rank* (MRR) which is defined as:

$$MRR = \frac{1}{2\left|\mathcal{D}_{test}\right|} \sum_{\eta \in \mathcal{D}_{test}} \left(\frac{1}{rank\left(o|s, r, t\right)} + \frac{1}{rank\left(s|r, o, t\right)} \right) \tag{16}$$

where $\eta = (s, r, o, t)$, $|\mathcal{D}_{test}|$ denotes the size of the test dataset. We also report $Hits@1$, $Hits@3$, and $Hits@10$ measures where $Hits@k$ represents the percentage of correct quadruple in the k highest ranked predictions, $Hits@k$ defined as:

$$Hit@k = \frac{1}{2\left|\mathcal{D}_{test}\right|} \sum_{\eta \in \mathcal{D}_{test}} \mathbb{I}_{(rank(o|s,r,t) \leq k)} + \mathbb{I}_{(rank(s|r,o,t) \leq k)} \tag{17}$$

where $\mathbb{I}_{(\cdot)}$ is an indicator function, $\mathbb{I}_{(cond)}$ is 1 if *cond* holds and 0 otherwise.

Table 1. Statistics of datasets.

Dataset	Entities	Relations	Training	Validation	Test
ICEWS14	6,869	230	72,826	8,941	8,963
ICEWS05-15	10,094	251	368,962	46,275	46,092
GDELT	500	20	2,735,685	341,961	341,961

Baselines. We test the performance of the proposed model against a variety of strong baselines, including static KG representation learning models and TKG representation learning models. Note that all these static models are applied without considering the time information in the input, including: TransE [1], DistMult [24], ComplEx [18], and SimplE [9]. The other TKG representation learning baselines models include: TTransE [7], HyTE [2], TA-TransE [4], DE-SimplE [5], ATiSE [22], and TeRo [21]. As TGAT [23] is specifically designed to handle dynamic network graphs not TKG, we have not compared with it.

Implementation. We implemented our model and the baselines in PyTorch and conducted the experiments on an NVIDIA Tesla V100 GPU. The vectors dimension of the entity, relation, and time are fixed to 128. We also tried to use different score functions to train the model, finally, we chose the ConvKB model as our decoder. The number of temporal neighbors samples is set to 20 for ICEWS14 and ICEWS05-15 datasets, 50 for the GDELT dataset. Theoretically, the information from multi-hop neighbors can be aggregated in our model, to

Table 2. Evaluation results on link prediction. The best results are in bold and the second-best results are underlined.

Dataset	**ICEWS14**				**ICEWS05-15**				**GDELT**			
Metrics	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**
TransE	0.280	0.094	-	0.637	0.294	0.090	-	0.663	0.155	0.060	0.178	0.335
DistMult	0.439	0.323	-	0.672	0.456	0.337	-	0.691	0.210	0.133	0.224	0.365
ComplEx	0.474	0.370	0.523	0.689	0.485	0.377	0.531	0.702	0.213	0.132	0.234	0.374
Simple	0.478	0.373	0.530	0.689	0.486	0.376	0.535	0.705	0.211	0.128	0.231	0.382
TTransE	0.255	0.074	-	0.601	0.271	0.084	-	0.616	0.115	0.0	0.160	0.318
HyTE	0.297	0.108	0.416	0.655	0.316	0.116	0.445	0.681	0.188	0.0	0.165	0.326
TA-TransE	0.275	0.095	-	0.625	0.299	0.096	-	0.668	-	-	-	-
TA-DistMult	0.477	0.363	-	0.686	0.474	0.346	-	0.728	0.206	0.124	0.219	0.365
DE-TransE	0.326	0.124	0.467	0.686	0.314	0.108	0.453	0.685	0.126	0.0	0.181	0.350
DE-DisMult	0.501	0.392	0.569	0.708	0.484	0.366	0.546	0.718	0.213	0.130	0.228	0.376
DE-SimplE	0.526	0.418	0.592	0.725	0.513	0.392	0.578	0.748	0.230	0.141	0.248	0.403
ATiSE	0.550	0.436	0.629	0.750	0.519	0.378	0.606	0.794	-	-	-	-
TeRo	0.562	0.468	0.621	0.732	0.586	0.469	0.668	0.795	-	-	-	-
TKGAT (ours)	**0.574**	**0.502**	**0.655**	**0.752**	**0.607**	**0.504**	**0.676**	**0.813**	**0.256**	**0.154**	**0.290**	**0.441**

speed up training, only the information about the 2-hop neighbors is aggregated. The number of attention heads and negative samples is set to 4 and 200 respectively, and the Adam SGD optimizer is applied to train model, we set 0.001 as the learning rate for all datasets.

5.2 Results and Analysis

Table 2 shows the experimental results of link prediction on ICEWS14, ICEWS05-15, and GDELT datasets. From the result, we can observe that the static KG representation learning models fell behind TKG models in most cases. The primary reason is static KG models only learned one representation for each entity or relation, without taking into account the temporal information.

The results also demonstrate the state-of-the-art performance of our approach for link prediction tasks. As we can see, the TKGAT model significantly improves on the suboptimal TeRo model for most metrics. The typical TKG representation learning models DE-SimplE, ATiSE, and TeRo, which pay more attention to model temporal information while ignoring to capture of the TKG topology structural information. In contrast, our model is based on the GNN framework, which has the advantage of building structural features. Besides, our model adopted attention networks to model relation weights and decoupled attention is applied to incorporate more extensive TKG structural features, which allowed our model accurately to describe entities and relations characteristics. TKGAT obtained central entity features by aggregating temporal neighbours, a large number of network parameters were used to learn the features, which increased a little model complexity but improved the accuracy. Meanwhile, time encoding function based on Bochner's theorem was employed to model relative time features, which further improved the model performance.

The experimental results also exhibt that the improvement in ICEWS05-15 and GDELT is greater than ICEWS14 dataset. the main reason is the comparatively small scale of the ICEWS14 dataset, in order to achieve the best prediction

results, a large amount of training data is required. In addition, the results show that the model performance on the ICEWS14 and ICEWS05-15 datasets are better than those on the GDELT datasets, the major reason is the quite small scale of the entities and relation types in the GDELT dataset, however, the interactions between entities are extremely complex, which makes challenging to extract effective information from the extremely complex interactions. Furthermore, the quality of the GDELT dataset is slightly lower, resulting in a relatively lower accuracy.

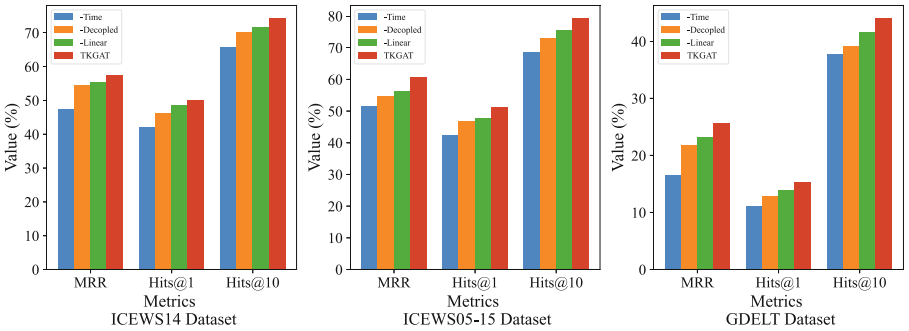

Fig. 3. Ablation study on three datasets

5.3 Ablation Study

To verify the effectiveness of each component in TKGAT, firstly, we implemented a version of TKGAT with all temporal attention weights set to the same value (-Time) to prove the validity of the time encoding function based on Bochner's theorem. Secondly, we removed the decoupled attention module (-Decoupled) and adopted the traditional self-attention mechanism directly to calculate attention scores between different entities. Finally, we incorporated relations information directly into the object using a linear transformation (-Linear) to verify the effectiveness of modeling relation weights.

As shown in Fig. 3, the TKGAT-Time model significantly reduced on MRR metric in all datasets, which proved the effectiveness of the time encoding function, and we can also notice that building temporal features in TKG is essential. In addition, the results show that the TKGAT-Decoupled model performed worse than the TKGAT model, which proved that the decoupled attention method is beneficial for improving the performance of the attention mechanism, and the correlations between entity and temporal features captured by decoupled attention are effective for TKG representation learning. We can also observe that the TKGAT-Linear model worked slightly worse than the TKGAT model, which indicates the effectiveness of capturing relations weights.

6 Conclusion

In this paper, we present a novel model, called TKGAT, for temporal knowledge graph representation learning. Specifically, time encoding function based on Bochner's theorem was applied to efficiently model relative time information, decoupled attention was adopted to capture the correlations between entity and temporal features, and the different relations influences were learned by attention network. Experimental results show that the TKGAT can effectively model temporal knowledge graph features. The ablation study also demonstrates the effectiveness of each component of TKGAT. For future work, the generation of time-aware discriminative negative samples is worth exploring.

Acknowledgment. This work is supported by the National Key R&D Program of China (2020AAA01 08504), the Key Research and Development Program of Ningxia Hui Autonomous Region (2023ZDYF0574), and the National Natural Science Foundation of China (61972275).

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 2787–2795. NIPS'13, Curran Associates Inc. (2013)
2. 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, pp. 2001–2011. EMNLP'18, Association for Computational Linguistics (2018)
3. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 1811–1818. AAAI'18, AAAI Press (2018)
4. García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4816–4821. EMNLP'18, Association for Computational Linguistics (2018)
5. Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 3988–3995. AAAI'20, AAAI Press (2020)
6. He, P., Liu, X., Gao, J., Chen, W.: Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654 (2020)
7. Jiang, T., et al.: Encoding temporal information for time-aware link prediction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2350–2354. EMNLP'16, Association for Computational Linguistics (2016)
8. 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. Association for Computational Linguistics, Online (2020)

9. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 4284–4295. NeurIPS'18, Curran Associates, Inc. (2018)
10. Lacroix, T., Obozinski, G., Usunier, N.: Tensor decompositions for temporal knowledge base completion. In: International Conference on Learning Representations, pp. 1–12. ICLR'20 (2020)
11. Li, Z., Liu, X., Wang, X., Liu, P., Shen, Y.: Transo: a knowledge-driven representation learning method with ontology information constraints. World Wide Web, pp. 1–23 (2022)
12. 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 (2021)
13. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2181–2187. AAAI'15, AAAI Press (2015)
14. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: 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, pp. 327–333. NAACL'18, Association for Computational Linguistics (2018)
15. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, pp. 809–816. ICML'11, Omnipress (2011)
16. 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., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.) The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
17. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: Proceedings of the Seventh International Conference on Learning Representations, pp. 328–337. ICLR'19 (2019)
18. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, pp. 2071–2080. ICML'16, JMLR.org (2016)
19. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
20. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1112–1119. AAAI'14, AAAI Press (2014)
21. Xu, C., Nayyeri, M., Alkhoury, F., Shariat Yazdi, H., Lehmann, J.: Tero: A time-aware knowledge graph embedding via temporal rotation. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1583–1593. COLING'20, International Committee on Computational Linguistics (2020)
22. Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H.S., Lehmann, J.: Temporal knowledge graph embedding model based on additive time series decomposition. arXiv preprint arXiv:1911.07893 (2019)

23. Xu, D., Ruan, C., Körpeoglu, E., Kumar, S., Achan, K.: Inductive representation learning on temporal graphs. In: 8th International Conference on Learning Representations. ICLR'20 (2020)
24. 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 third International Conference on Learning Representations, pp. 809–816. ICLR'15 (2015)
25. Zhang, F., Wang, X., Li, Z., Li, J.: Transrhs: A representation learning method for knowledge graphs with relation hierarchical structure. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 2987–2993. IJCAI'20, International Joint Conferences on Artificial Intelligence Organization (2020)