Spatial-Temporal Attention Network for Temporal Knowledge Graph Completion

Jiasheng Zhang[1,2], Shuang Liang[1], Zhiyi Deng[1], and Jie Shao[1,2](✉)

[1] University of Electronic Science and Technology of China, Chengdu 611731, China
{zjss12358,shuangliang,zhiyideng}@std.uestc.edu.cn, shaojie@uestc.edu.cn
[2] Sichuan Artificial Intelligence Research Institute, Yibin 644000, China

Abstract. Temporal knowledge graph completion, which aims to predict missing links in temporal knowledge graph (TKG), is an important research task due to the incompleteness of TKG. Recently, TKG embedding methods have proved to be effective for this task. However, most of existing methods regard TKG as a set of independent facts and consequently ignore the implicit relevance among facts. Actually, as a kind of dynamic heterogeneous graph, the evolving graph structure of TKG is able to reflect a wealth of information. To this end, in this paper we regard temporal knowledge graph as heterogeneous and discrete spatial-temporal resource, and propose a novel spatial-temporal attention network to learn TKG embeddings by modeling spatial-temporal property of TKG while considering its special characteristics. Specifically, our model employs a **M**ulti-**F**aceted **G**raph **At**tention Network (MFGAT) to extract rich structural information from the egocentric network of each entity. Additionally, an **Ad**aptive **T**emporal **At**tention Mechanism (ADTAT) is utilized to flexibly model the correlation of entity representations in the time dimension. Finally, by combing our obtained representations with existing static KG completion methods, they can be extended to spatial-temporal versions to predict missing links in TKG while considering its inherent graph structure and time-evolving property. Experimental results on three real-world datasets demonstrate the superiority of our model over the state-of-the-art methods.

Keywords: Temporal knowledge graph completion · Temporal knowledge graph embedding learning · Spatial-temporal data mining

1 Introduction

Temporal knowledge graph (TKG) is a knowledge base system which contains facts happened in real-world with the corresponding happened times. As shown in Fig. 1, TKG can be represented as a dynamic heterogeneous graph in which nodes denote entities in real-world and labeled edges represent relations among entities. Moreover, nodes and edges in the graph will appear or disappear with the development of time which leads to that the structure of the graph evolves over time and the static graph in each timestamp is called a snapshot.

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12681, pp. 207–223, 2021.
https://doi.org/10.1007/978-3-030-73194-6_15

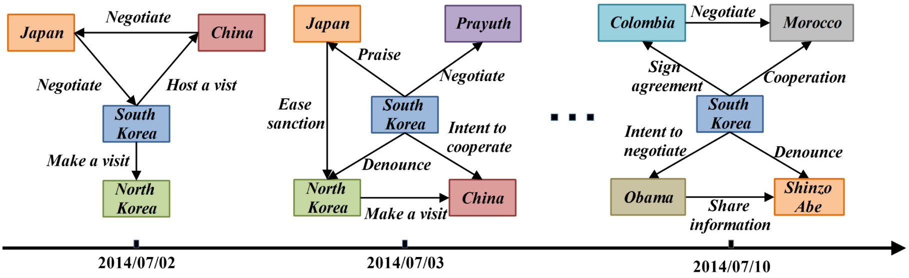

Fig. 1. An example of temporal knowledge graph. In each snapshot we give an example of the egocentric network of *South Korea*.

Compared with static knowledge graph (KG) which ignores the time annotations of facts, TKG is more adequate for real-world scenarios and thus receives a surge of interest in recent years. However, same as static knowledge graph, temporal knowledge graph is also far from complete. Therefore, the task of predicting missing links in TKG, which is known as temporal knowledge graph completion (TKGC) becomes an increasingly important research task in this field.

KG embedding methods, which aim to map each element of KG to a hidden vector representation, is a powerful technique for static knowledge graph completion. However, such methods fail to consider the time annotations of facts. Therefore, some researchers turn to temporal knowledge graph embedding methods for the TKGC task in recent years, several methods have been proposed such as TAE [10] and HyTE [3]. Although these methods outperform KG embedding methods on the TKGC task, they mostly regard TKG as a set of independent facts and thus ignore the graph structure of TKG, which fails to capture the implicit relevance among facts. Furthermore, most of them treat facts in each snapshot separately and thus ignore the time-evolving property of TKG, which fails to obtain more accurate representations based on the information of history snapshots. Therefore, the performance of TKGC is still far from satisfactory and it is necessary to develop a model that can consider graph structure and time-evolving property of TKG simultaneously.

Actually, we notice that temporal knowledge graph can be viewed as a kind of spatial-temporal resource where graph structure in each snapshot reflects its spatial property and the correlation of different snapshots in the time dimension reflects its temporal property. Recently, deep spatial-temporal models [24] have achieved successes in many fields due to their effectiveness in modeling spatial-temporal correlation of data, so we argue that learning TKG embeddings via deep spatial-temporal models can effectively consider its graph structure and time-evolving property. However, there are still no studies applying such models to TKG because TKG has two characteristics: 1) *heterogeneity*, as shown in Fig. 1, nodes in the graph correspond to entities in the real world, which leads to that different nodes have different semantics and thus play different roles in the graph; 2) *discreteness*, facts in TKG are discretely distributed in the time dimension, which leads to that data quantities of different snapshots are inhomogeneous. For a particular entity, some snapshots contain more related facts while others contain fewer or even no related facts.

Based on above considerations, in this paper, we propose a novel spatial-temporal attention network to learn TKG embeddings by modeling its spatial-temporal property. First, in order to model the spatial property and heterogeneity of TKG, we focus on the egocentric network [8], which is defined as the induced graph of a node with its immediate neighbors. It is considered as the basic structure that dominates the attributes and behaviors of nodes in the field of social network analysis [1]. As shown in Fig. 1, we give an example of egocentric networks of *South Korea* in different snapshots. Compared with the star-like structure considered by previous graph neural network (GNN) models, such as GAT [23] and R-GCN [20], which can only consider the binary relationships between nodes, egocentric network can capture the multiple relationships among a node and its neighbors, and thus is able to describe the role of a node in the graph more accurately. In this way, we develop a novel **M**ulti-**F**aceted **G**raph **At**tention Network (MFGAT) based on the egocentric network. Specifically, for each snapshot, it firstly constructs rich structural features from the egocentric network of each entity, and then an attention mechanism is applied for each feature independently. Finally, by fusing different kinds of features, our MFGAT can effectively learn TKG embeddings of each snapshot while considering the graph structure and heterogeneity of TKG.

Additionally, in order to model the time-evolving property of TKG while addressing the inhomogeneity problem brought by discrete distribution, we propose a novel **Ad**aptive **T**emporal **At**tention Mechanism (ADTAT). The core component of ADTAT is a mask function which is able to dynamically select attention position for each entity to focus on the information of active snapshots. Furthermore, it can adaptively model the time span information based on the fact distribution of each entity in the time dimension. By employing an attention mechanism with our mask function, ADTAT is able to flexibly model the temporal correlation of entity representations in different snapshots.

Combining the above two parts, our spatial-temporal attention network can learn TKG embeddings while considering the graph structure and time-evolving property of TKG simultaneously. Furthermore, existing static knowledge graph embedding methods can be extended to a spatial-temporal version for the TKGC task by applying our obtained representations in the score function. Main contributions of our work are summarized as follow:

- We propose a novel spatial-temporal attention network for TKG completion. To the best of our knowledge, this is the first work that learns TKG embeddings from the perspective of spatial-temporal data modeling.
- We introduce egocentric network to the field of TKG, and propose a novel multi-faceted graph attention network based on egocentric network of each entity to capture the structural information of TKG more effectively.

- Experimental results on three real-world datasets demonstrate the superiority of our model. Our source code and datasets are publicly available at https://github.com/zjs123/ST-ConvKB.

2 Related Work

In this section, we first provide an overview of the typical methods for static knowledge graph embedding learning and temporal knowledge graph embedding learning respectively, and then briefly review deep spatial-temporal model and its recent advances in several fields.

2.1 Static Knowledge Graph Embedding Methods

Static knowledge graph embedding methods aim to represent each element of knowledge graph as a low-dimensional vector while preserving its inherent semantic. There exist two kinds of typical methods, namely translation methods and semantic matching methods. TransE [2] is a typical translation method, which maps each entity to a vector and regards relation as the translation from subject entity to object entity. Based on TransE, a number of improved methods have been proposed, such as TransH [25], TransR [15], and TransD [9]. RESCAL [19] is the first semantic matching method that utilizes restricted Tucker decomposition for static knowledge graph embedding learning. Due to too many parameters of RESCAL, DistMult [27] simplifies RESCAL by using diagonal matrix. Other semantic matching methods have been further proposed, such as HoIE [18] and ComplEx [22]. Besides the above two kinds of methods, in recent years, some researchers attempt to learn KG representations based on convolution, such as ConvE [4] and ConvKB [17]. Furthermore, there are also some works attempt to learn KG representations based on graph neural networks, such as R-GCN [20] and KBAT [16].

2.2 Temporal Knowledge Graph Embedding Methods

Temporal knowledge graph embedding methods aim to learn representations for each element of TKG while considering the happened times of facts. TAE [10] is the first work that attempts to incorporate temporal order information between relations into TKG embeddings. Based on this, TKGFrame [28] formally defines the relation chain of TKG and incorporates it into TKG embeddings. Inspired by the objective of TransH, HyTE [3] projects the embeddings of entity and relation to a time-specific hyperplane and applies TransE score function for the embeddings in each hyperplane. TTransE [14] is an extension of TransE by considering time embeddings in the score function. TA-DistMult [6] constructs temporal relation embeddings for each fact by encoding corresponding time annotation with an LSTM model. Recently, DE-DistMult [7] provides a diachronic entity embedding function to distinguish entities in different time stamps. Inspired by the canonical decomposition of tensors of order 4, TNTComplEX [13] proposes a new regularization scheme and presents a temporal extension of ComplEX. Although these methods have achieved significant performance on the TKGC task, all of them ignore graph structure of TKG and they mostly are unable to capture the correlation of facts in the time dimension. RE-NET [11] is the only work that considers both of them, but this model is designed for extrapolation problem rather than learning embeddings for TKGC.

2.3 Deep Spatial-Temporal Models

Deep spatial-temporal models are a kind of spatial-temporal data mining model based on deep learning techniques. These models mostly contain a spatial part to model the spatial property of data, and the most used deep learning models are convolutional neural network and graph convolutional network (GCN) [12]. A temporal part is used to capture the temporal correlation of data, in which recurrent neural network (RNN) is widely used. Based on the above architecture, several models have been proposed in different fields to model data with spatial-temporal property. For example, GMAN [29] combines a spatial attention model and a temporal attention model with a gated mechanism to predict future traffic conditions, ConvLSTM [21] integrates the structure of CNN and LSTM to predict the spatial-temporal sequences and ST-GCN [26] combines spatial and temporal convolutions for action recognition. These successful attempts demonstrate the universality of deep spatial-temporal models and inspire us to design a spatial-temporal model for temporal knowledge graph embedding learning. More detailed introduction of deep spatial-temporal models can be viewed in [24].

3 Preliminaries

Definition 1 (Temporal Knowledge Graph). *Temporal knowledge graph can be denoted as a sequence of static snapshots $G = \{G^1, G^2, ..., G^{|\mathcal{T}|}\}$, where each snapshot contains facts happened in the same time. $G^t = \{(s_i, r_i, o_i, t)\}$ in which $s_i \in \mathcal{E}$ and $o_i \in \mathcal{E}$ are subject entity and object entity respectively, $r_i \in \mathcal{R}$ is the relation and $t \in \mathcal{T}$ denotes the happened time of these facts.*

Definition 2 (Temporal Knowledge Graph Completion). *Temporal knowledge graph completion (as known as link prediction) aims to predict fact (s, r, o, t) when s or o is missing. It can be divided into two subtasks, one is subject entity prediction to predict s given r and o in time t, and the other is object entity prediction to predict o given r and s in time t.*

Definition 3 (Egocentric Network). *Given a node u in network G, the egocentric network of u is a subgraph which is composed of u, its neighbors $\mathcal{N}(u)$, and edges between them, which can be denoted as $G_u = (V_u, E_u)$, where $V_u = u \cup \mathcal{N}(u)$ and E_u are node set and edge set of G_u respectively. Particularly, in this paper we use G_e^t to denote the egocentric network of entity e in the snapshot G^t.*

4 Proposed Model

In this section, we give an introduction of our model in detail. As shown in Fig. 2, our model takes a sequence of snapshots $\{G^1, G^2, ..., G^{|\mathcal{T}|}\}$ as input (part (b)), the multi-faceted graph attention network (part (a)) is first used to obtain entity and relation representations in each snapshot, and then adaptive temporal attention mechanism is utilized to model the temporal correlation of entity representations in different snapshots. After obtaining final entity and relation representations, they can be used to predict missing links via a score function (part (c)).

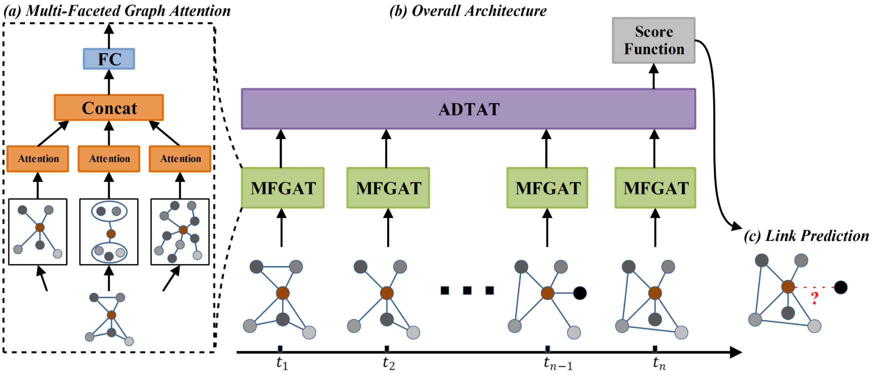

Fig. 2. We give an overview of the architecture of our proposed model in part (b), the detailed illustration of MFGAT is shown in part (a), and after obtaining the final embeddings, they will be used to predict missing links as shown in part (c).

4.1 Multi-faceted Graph Attention Network

As shown in Fig. 2(a), first, due to the complex structure of egocentric network, our MFGAT constructs three kinds of structural features called triple feature, group feature, and path feature based on egocentric network of each entity to adequately describe its structure. Then, the attention mechanism is applied for each feature independently to screen out important information. Finally, the representation of each entity is obtained via a fully connected layer. In this part, we take entity e in snapshot G^t as an example to introduce the detailed process of our MFGAT to obtain its representation and representations of other entities can be obtained in the same way.

Triple Feature. Triple is the basic structure in temporal knowledge graph which can describe the binary relation among entities. In the egocentric network G_e^t, triples that involve e are able to illustrate the direct relevance between e and its neighbors, therefore it is important to integrate the information of such basic structure. We construct triple feature for each fact (e, r_i, e_i, t) in the egocentric network G_e^t as follows:

$$\mathbf{u}_i^t = \mathbf{r}_i \circledast \mathbf{e}_i, \tag{1}$$

in which $\mathbf{r}_i \in \mathbb{R}^d$ is the initial embedding of relation r_i and $\mathbf{e}_i \in \mathbb{R}^d$ is the initial embedding of entity e_i. We obtain the triple feature $\mathbf{u}_i^t \in \mathbb{R}^d$ via circular-correlation operation $\circledast$ which is employed in HoIE [18] due to its high expressivity. Finally, by constructing triple feature for each fact that involves e, we can obtain a set of triple features $\{\mathbf{u}_1^t, \mathbf{u}_2^t, ..., \mathbf{u}_{|\mathcal{N}^t(e)|}^t\}$ where $|\mathcal{N}^t(e)|$ is the number of neighbors of entity e in snapshot G^t.

Group Feature. Neighbors in egocentric network can be divided into several independent groups based on their connectivity and the connected neighbors in each group are generally a set of entities that have similar characteristics to the central entity. Specifically, we regard each group in the egocentric network as a set of nodes that can be connected through paths that do not go through the central entity. As shown in Fig. 1, there are two groups in the egocentric network of *South Korea* in 2014/07/10, one contains *Obama* and *Shinzo Abe* which are the presidents of partner countries of *South Korea* while the other contains *Colombia* and *Morocco* which are cooperation countries. Groups in the egocentric network can reflect the multiple relations among neighbor entities and provide an abstract perspective for the relevance between an entity and its neighbors. Therefore, in order to consider the information of such structure, we define the graph feature of each graph in the egocentric network G_e^t as follows:

$$\mathbf{v}_i^t = \mathbf{MAXPOOL}\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n\}, \tag{2}$$

where $\mathbf{e}_k \in \mathbb{R}^d$ is the initial embedding of each entity in the group and n is the total number of entities in the group. The group feature $\mathbf{v}_i^t \in \mathbb{R}^d$ is obtained by applying max-pooling operation for entities in the group to screen out the most prominent features of them. Finally, we can obtain a set of group features $\{\mathbf{v}_1^t, \mathbf{v}_2^t, ..., \mathbf{v}_{|\mathcal{G}^t(e)|}^t\}$ where $|\mathcal{G}^t(e)|$ is the number of groups in G_t^e.

Path Feature. Relational path is widely used to model complex graph structure of knowledge graph because it can reflect multi-hop relations between entities. In the egocentric network G_e^t, relational path between e and each of its neighbors is able to illustrate indirect relevance between them. In this part, for each neighbor entity e_i in G_e^t, we randomly find a relational path of length 2 from e to e_i in the egocentric network, which can be denoted as (e, r_{i1}, r_{i2}, e_i). The corresponding path feature is obtained as follows:

$$\mathbf{o}_i^t = \mathbf{W}_o[\mathbf{r}_{i1} : \mathbf{r}_{i2} : \mathbf{e}_i], \tag{3}$$

in which $\mathbf{r}_{i1} \in \mathbb{R}^d$ and $\mathbf{r}_{i2} \in \mathbb{R}^d$ are initial embeddings of relations involved in the path and $\mathbf{e}_i \in \mathbb{R}^d$ is the initial embedding of neighbor entity e_i, $\mathbf{W}_o \in \mathbb{R}^{d \times 3d}$ denotes the linear transform matrix and [:] is concatenation operation. By constructing path feature for each neighbor, we can obtain a set of path features $\{\mathbf{o}_1^t, \mathbf{o}_2^t, ..., \mathbf{o}_{|\mathcal{N}^t(e)|}^t\}$ where $|\mathcal{N}^t(e)|$ is the number of neighbors of entity e in snapshot G^t.

Feature Fusion. After obtaining the above three kinds features $\{\mathbf{u}_i^t\}$, $\{\mathbf{v}_i^t\}$ and $\{\mathbf{o}_i^t\}$, we then apply the attention mechanism to each of them independently, and for each kind of feature we can obtain a set of attention weights $\{\alpha_1^t, \alpha_2^t, ..., \alpha_{N_c}^t\}$ which quantify the importance of feature $\{\mathbf{c}_1^t, \mathbf{c}_2^t, ..., \mathbf{c}_{N_c}^t\}$ for entity e. $\mathbf{c}$ can be $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{o}$, and N_c is the length of each feature sequence.

$$\alpha_i^t = \frac{exp(\mathbf{e}^\top \mathbf{U}\mathbf{c}_i^t)}{\sum_{j=1}^{N_c} exp(\mathbf{e}^\top \mathbf{U}\mathbf{c}_j^t)}, \tag{4}$$

$$\tilde{\mathbf{c}}^t = \sum_{i=1}^{N_c} \alpha_i^t \mathbf{c}_i^t, \tag{5}$$

in which $\mathbf{U} \in \mathbb{R}^{d\times d}$ is the transfer matrix to be learned and $\mathbf{e}$ is the initial embedding of entity e. As shown in Eq. 5, we obtain the corresponding output vector $\tilde{\mathbf{c}}^t \in \mathbb{R}^d$ of each kind of feature as the weighted average. Finally, we concatenate the obtained three kinds of feature vectors with the initial embedding of e and employ a fully connected layer to obtain the output representation of entity e in snapshot G^t as follows:

$$\tilde{\mathbf{e}}^t = \sigma(\mathbf{W}[\mathbf{e} : \tilde{\mathbf{u}}^t : \tilde{\mathbf{v}}^t : \tilde{\mathbf{o}}^t] + \mathbf{b}). \tag{6}$$

Unseen Entity Transform. If there are no related facts of entity e in snapshot G^t, our MFGAT obtains the corresponding representation via another fully connected layer as follows:

$$\tilde{\mathbf{e}}^t = \sigma(\mathbf{W}_{ent}\mathbf{e} + \mathbf{b}_{ent}). \tag{7}$$

Finally, by applying MFGAT for entity e in different snapshots, we can obtain a sequence of output representation vectors for different snapshots, which can be denoted as $\{\tilde{\mathbf{e}}^1, \tilde{\mathbf{e}}^2, ..., \tilde{\mathbf{e}}^{|\mathcal{T}|}\}$, where $|\mathcal{T}|$ is the total number of snapshots.

Relation Transform. Further, after obtaining entity representations via multifaceted graph attention network, the relation representations are also transformed as follows:

$$\tilde{\mathbf{r}} = \mathbf{r} \cdot \mathbf{W}_{rel}, \tag{8}$$

where $\mathbf{r} \in \mathbb{R}^d$ is the initial relation embedding and $\tilde{\mathbf{r}} \in \mathbb{R}^d$ is the transformed relation embedding. $\mathbf{W}_{rel} \in \mathbb{R}^{d\times d}$ is the learnable transform matrix used to project relation embeddings to the same vector space as entity embeddings.

4.2 Adaptive Temporal Attention Mechanism

In temporal knowledge graph, the temporal correlation of entity representations in different snapshots mainly relies on two parts. First, it is affected by the inherent semantic correlation of entity representations. As shown in Fig. 1, representations of *South Korea* in 2014/07/02 and 2014/07/03 tend to have high correlation because *South Korea* interact with *Japan*, *China*, and *North Korea* in both two snapshots. Second, it is also affected by the time span between snapshots, and entity representations with long time span tend to have low correlation because the effects of facts will attenuate over time. Our MFGAT can effectively learn entity representations in each snapshot, but it fails to model the correlation of entity representations in different snapshots. Furthermore, as we mentioned, data quantities of different snapshots are inhomogeneous in temporal knowledge graph which leads to the complexity of modeling temporal correlation of entity representations. To this end, we develop a novel adaptive temporal attention mechanism (ADTAT) to flexibly capture the correlation of entity representations in different snapshots. For each entity e, our ADTAT takes the output representation sequence $\{\tilde{\mathbf{e}}^1, \tilde{\mathbf{e}}^2, ..., \tilde{\mathbf{e}}^{|\mathcal{T}|}\}$ of our MFGAT as input and the correlation of its representations in time t and t_j ($t_j \leq t$) is measured as follows:

$$\beta^{t,t_j} = \frac{m_e(t,t_j)exp(\sigma(\mathbf{a}^\top \cdot [\mathbf{W}_1\tilde{\mathbf{e}}^t : \mathbf{W}_2\tilde{\mathbf{e}}^{t_j}]))}{\sum_{t_k \leq t} m_e(t,t_k)exp(\sigma(\mathbf{a}^\top \cdot [\mathbf{W}_1\tilde{\mathbf{e}}^t : \mathbf{W}_2\tilde{\mathbf{e}}^{t_k}]))}, \tag{9}$$

where $\tilde{\mathbf{e}}^t \in \mathbb{R}^d$ and $\tilde{\mathbf{e}}^{t_j} \in \mathbb{R}^d$ are representations of entity e in time t and t_j respectively, $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are two learned transform matrices, and $\mathbf{a} \in \mathbb{R}^{2d}$ is the attention vector. $m()$ is a mask function, in which firstly, in order to avoid attention smooth problem brought by inhomogeneous data distribution, for each entity e, if there are no facts that involve e in the snapshot G^{t_j}, $m_e(t,t_j)$ will be set as 0, which forces our attention mechanism to focus on the active snapshots of entity e. In addition, in order to capture time span information of TKG, we employ a temporal attenuation function with a dynamic attenuation coefficient γ_e^t, since facts of each entity are distributed inhomogeneously in the time dimension, too large attenuation coefficient will lead to local sparse entities fails to capture sufficient history information, but too small attenuation coefficient will make our model unable to adequately consider the effect of time span. Therefore, we define the dynamic attenuation coefficient as follows:

$$\gamma_e^t = \frac{\sum_{|t_i - t| \leq \frac{\sqrt{|\mathcal{T}|}}{2}} |\mathcal{N}^{t_i}(e)|}{\sqrt{|\mathcal{T}|} - 1} \cdot \lambda, \tag{10}$$

in which $|\mathcal{N}^{t_i}(e)|$ is the number of neighbors of entity e in the snapshot G^{t_i}, and λ is the basic attenuation coefficient. For each entity e, the size of γ_e^t is related to the distribution of facts around snapshot G^t, and the sparser distribution will lead to the smaller attenuation coefficient. Combining above two parts, the mask function of our ADTAT can be defined as follows:

$$m_e(t,t_j) = \begin{cases} exp(-\gamma_e^t(|t - t_j|)), & e \in G^{t_j} \\ 0, & otherwise \end{cases}. \tag{11}$$

Based on the mask function, our ADTAT is able to model the temporal correlation of entity representations while effectively tackle the inhomogeneity problem of TKG. The output representation of each entity e in time t is obtained as follows:

$$\mathbf{h}_e^t = \sum_{t_j \leq t} \beta^{t,t_j} \tilde{\mathbf{e}}^{t_j}. \tag{12}$$

Finally, our ADTAT can obtain the final representations $\{\mathbf{h}_e^1, \mathbf{h}_e^2, ..., \mathbf{h}_e^{|\mathcal{T}|}\}$ of each entity e in different snapshots while considering the graph structure and temporal correlation of TKG.

4.3 Training

After obtaining the final representations of entity and relation, they can be used in the score function of existing static knowledge graph embedding methods such as TransE [2] and DistMult [27] to obtain the spatial-temporal version of these methods for TKGC. Here, we give the illustration of using ConvKB [17] score function because it achieves the best performance in our experiment and the performances of different score functions will be presented in Sect. 5. The score function of each fact (s, r, o, t) can be defined as follows:

$$f(s, r, o, t) = \mathbf{contact}(g([\mathbf{h}_s^t : \tilde{\mathbf{r}} : \mathbf{h}_o^t] * \Omega)) \cdot \mathbf{w}, \tag{13}$$

where $\mathbf{h}_s^t \in \mathbb{R}^d$ and $\mathbf{h}_o^t \in \mathbb{R}^d$ are obtained representations of s and o in time t respectively, and $\tilde{\mathbf{r}} \in \mathbb{R}^d$ is the obtained relation representation for r. After obtaining the score of each fact, the model is then trained using soft-margin loss as follows:

$$L = \sum_{x \in \{S \cup S'\}} log(1 + exp(l_x \cdot f(x))) + \frac{\lambda}{2}||\mathbf{w}||_2^2, \tag{14}$$

where S is the set of positive facts, and S' is a set of negative facts obtained by randomly replacing subject or object entity of each positive fact. l_x is the indicator variable which is set as 1 when $x \in S$ and -1 when $x \in S'$.

5 Experiments

In this section, we first provide an overview of the detailed settings in our experiment, and then we report extensive experimental evaluations and provide the analysis of the experimental results.

5.1 Experimental Settings

Datasets. We evaluate our model and baselines on three public datasets released by TA-DistMult [6], which are derived from two popular temporal knowledge graph resources, namely ICEWS and Wikidata [5]. Simple statistics of three datasets are summarized in Table 1, and we detail each dataset as follows:

- **ICEWS14:** This is a short-range version subset of ICEWS recourse by collecting all facts from 2014/1/1 to 2014/12/31 with the granularity of daily, and there are 7,128 distinct entities and 230 types of relations in this dataset.
- **ICEWS05-15:** This is a long-range version subset of ICEWS recourse which is almost 5 times larger than ICEWS14. It contains facts from 2005/1/1 to 2015/12/31 with the granularity of daily and there are 10,488 distinct entities and 251 types of relations in this dataset.

- **WIKIDATA11k:** This is a subset of Wikidata which contains 11,134 distinct entities, 95 types of time-sensitive relations, and in total of 28.5k facts with the granularity of year.

Table 1. Statistics of datasets.

Datasets	Entity	Relation	Fact			Time
			Train	Valid	Test	
ICEWS14	6,869	230	72.8k	8.9k	8.9k	365
ICEWS05-15	10,094	251	368k	46.2k	46k	4017
WIKIDATA11k	11,134	95	121k	14.3k	14.2k	306

Since our model is designed for the TKGC task rather than extrapolation, we utilize random-split and sample roughly 80% of instances as training, 10% as validation, and 10% for testing on each dataset.

Baselines. We compare our model with a suite of state-of-the-art baselines which have been introduced in Sect. 2, such as TAE [10], HyTE [3], and DE-DistMult [7]. Note that, we did not compare our model with RE-NET [11] because RE-Net is designed for extrapolation task rather than TKGC. Furthermore, in order to compare the performance of our model using different score functions, we refer to the resulting models as ST-X, such as ST-TransE and ST-DistMult, where ST is short for **S**patial-**T**emporal.

Metrics. For each test fact (s, r, o, t), we corrupt it by replacing the subject or object entity by all possible entities in turn and obtain a list of candidate facts, and then these candidate facts and original fact are ranked in descending order of their plausibility score. The rank of original fact denoted as $rank(s, r, o, t)$ is the basic metric of the TKGC task, and then we use two kinds of refined metrics based on this to evaluate the performance of each model. One is mean reciprocal rank (MRR) defined as $MRR = \frac{1}{|Test|}\sum_{(s,r,o,t)\in Test}\frac{1}{rank(s,r,o,t)}$, which is the average of the reciprocal of the rank of each test fact, and the higher MRR denotes the better model performance. The other is Hits@N which is defined as $Hits@N = \frac{1}{|Test|}\sum_{(s,r,o,t)\in Test} ind(rank(s, r, o, t) \leq N)$, where $ind()$ is 1 if the inequality holds and 0 otherwise.

Implementation. We implement our model in PyTorch, and all the experiments are performed on an Intel Xeon CPU E5-2640(v4) with 128 GB main memory, and Nvidia TITAN RTX. We initialize all the baselines with the parameter settings in the corresponding papers and then turn them on our datasets for the best performance for a fair comparison. For our model, we create 100 mini-batches for each epoch during training. The dimension of embedding representations $d \in \{50, 100, 200\}$, learning rate $l \in \{10^{-2}, 10^{-3}, 10^{-4}\}$, negative sampling ratio $n \in \{1, 3, 5, 10\}$, basic attenuation coefficient $\lambda \in \{1, 3, 5\}$. The best configuration is chosen based on MRR on the validation dataset. The final parameters are $d = 100$, $l = 10^{-2}$, $n = 5$, $\lambda = 1$ for the ICEWS14 dataset. For the WIKIDATA11k and and ICEWS05-15 datasets, the best configuration is $d = 100$, $l = 10^{-2}$, $n = 3$, $\lambda = 3$.

Table 2. Comparison of different methods on three datasets for link prediction. The best and second best results in each column are boldfaced and underlined respectively (the higher is better for each metric).

Dataset	ICEWS14				ICEWS05-15				WIKIDATA11k			
Models	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*	*MRR*	*Hit@1*	*Hit@3*	*Hit@10*
TransE	0.280	9.4	–	63.7	0.294	9.0	–	66.3	0.316	18.1	–	65.9
DistMult	0.439	32.3	–	67.2	0.456	33.7	–	69.1	0.316	18.1	–	66.1
ConvKB	0.335	22.4	38.7	56.6	–	–	–	–	0.267	12.2	29.6	63.1
TAE	0.263	10.1	49.7	66.2	0.295	10.4	49.0	71.4	0.319	18.3	39.2	65.7
TA-DistMult	0.435	31.5	49.1	68.3	0.468	35.2	51.8	72.8	0.557	40.6	58.6	78.4
TTransE	0.227	7.2	30.1	58.2	0.243	7.6	26.5	57.8	0.294	18.3	35.2	60.9
HyTE	0.297	10.8	41.6	65.5	0.316	11.6	44.5	68.1	0.371	21.5	45.9	75.1
DE-DistMult	0.501	39.2	56.9	70.8	0.484	36.6	54.6	71.8	0.396	24.1	45.7	74.5
TNTComplEX	0.616	**51.8**	65.7	75.8	0.665	59.0	70.5	80.7	0.408	23.9	47.8	75.6
ST-TransE	0.396	9.1	66.8	**86.4**	0.457	12.4	76.2	**93.2**	0.647	56.3	70.4	**78.8**
ST-DistMult	0.603	48.3	67.2	83.0	0.673	55.1	75.0	91.6	0.625	54.9	67.0	75.8
ST-ConvKB	**0.629**	51.0	**71.5**	85.1	**0.704**	**59.3**	**79.6**	91.9	**0.649**	**57.3**	**73.4**	77.9

5.2 Performance Comparison

Table 2 illustrates the results of baselines and our proposed models using different score functions in the link prediction task. According to the results, firstly, our proposed model outperforms all the baselines by a significant improvement, which demonstrates the superiority of our model to obtain more accurate representation for temporal knowledge graph. The improvement of Hits@10 on the ICEWS05-15 dataset is the highest, which may be because that ICEWS05-15 is relatively larger and hence the subgraph in each snapshot is denser, so that our MFGAT can capture richer structural information. TNTComplEX [13] fails to achieve good performance on the WIKIDATA11k dataset because its model is sensitive to data sparsity. Furthermore, the spatial-temporal version of each static method outperforms original counterpart on all metrics, which gives evidence of the merit of considering graph structure and temporal correlation of TKG. DE-DistMult [7] outperforms static KG method DistMult [27] on all datasets, which demonstrates the importance of integrating temporal information for the TKGC task. However, DE-DistMult fails to consider structural information of TKG, therefore, our ST-DistMult consistently outperforms DE-DistMult, which shows the necessity of considering graph structure in the TKGC task. DistMult-based models consistently outperform TransE-based models [2] due to the higher expressivity of DistMult score function. ConvKB [17] has the highest expressivity and thus achieves the best performance. What is more, ST-TransE gets low Hit@1 on ICEWS14 and ICEWS05-15 but high on WIKIDATA11k because the number of relations in ICEWS14 and ICEWS05-15 is much larger than that of WIKIDATA11k which leads to higher complexity.

5.3 Model Variants and Ablation Study

We run experiments on the ICEWS14 dataset with several variants of our proposed model to provide a better understanding of the effectiveness of each part in our model. The results are shown in Table 3, which includes ST-ConvKB and its variants.

Table 3. Performance of different variants of our model for link prediction.

Variants	MRR	Hit@1	Hit@10
Replacing MFGAT with GAT [23]	0.480	29.7	81.7
Replacing MFGAT with KBAT [16]	0.582	45.8	82.4
Replacing MFGAT with R-GCN [20]	0.531	34.2	83.1
MFGAT without triple feature	0.568	42.6	83.6
MFGAT without group feature	0.583	46.3	81.5
MFGAT without path feature	0.581	46.0	81.1
ADTAT without temporal attenuation	0.598	47.4	85.4
ADTAT with static temporal attenuation coefficient ($\lambda = 0.1$)	0.605	48.9	84.8
ADTAT with static temporal attenuation coefficient ($\lambda = 1$)	0.610	49.6	84.3
ST-ConvKB	0.629	51.0	85.1

Effect of Different Spatial Models. First, as shown in Table 3, the performance of variants with different graph neural network models outperform most of baselines, which indicates the importance of integrating structure information of temporal knowledge graph. Hit@1 of the variant with GAT is lower than other variants because GAT only considers neighbor entities but ignores the information of relations. Hit@10 of all variants are at the same level because all of them are able to capture the co-occurrence relationship among entities. Furthermore, ST-ConvKB outperforms all these variants, which illustrates the superiority of egocentric network considered in our model.

Effect of Each Feature in MFGAT. As shown in Table 3, we compare our model with three variants without triple feature, group feature, and path feature respectively. First, all of these variants are unable to outperform our original model which illustrates that all three kinds of features are effective and contribute to the final performance of our model. Furthermore, the performance of the variant without triple feature drops most because triple feature provides the most intuitive relevance of an entity with its neighbors.

Effect of Adaptive Temporal Attenuation Function. We first compare our model with a simple attention version without temporal attenuation. The Hit@10 result of this variant is at the same level as ST-ConvKB, which indicates that both our original model and this variant are able to capture adequate history information for each snapshot. However, the MRR and Hit@1 results of this variant are lower because it is unable to consider time span and thus the information of long-range snapshots will confuse the model to obtain more accurate predictions. Furthermore, we compare our model with variants using different static temporal attenuation coefficients ($\lambda = 0.1$ and $\lambda = 1$). They are unable to outperform our original model because large attenuation coefficient will let the model fail to capture sufficient history information for locally sparse entities, and small attenuation coefficient will let the model fail to consider the effect of time span adequately.

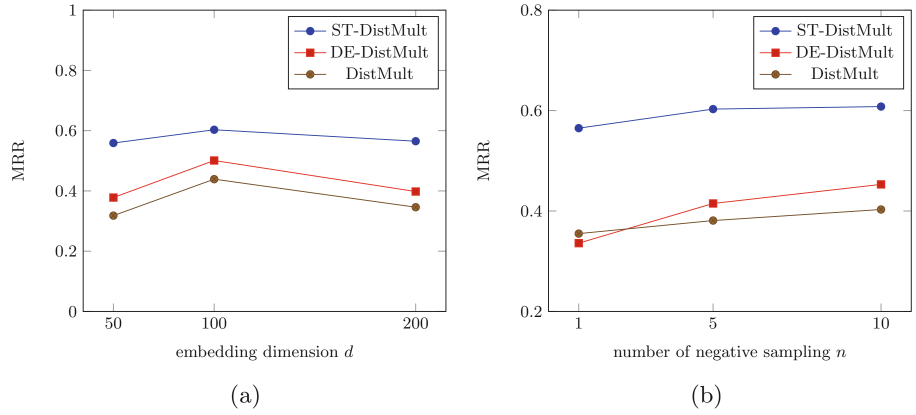

Fig. 3. Influence of the embedding dimension and negative sampling number.

5.4 Parameter Analysis

We study the impact of the training parameters of our model in this part, including the dimension of embedding representations d and the number of negative samples n.

Dimension of Embedding Representation. Here, we analyze the performance of ST-DistMult which considers both the graph structure and time-evolving property of TKG, DE-DistMult which only considers the time-evolving property and static KG method DistMult on changing the dimension of embedding representations. As shown in Fig. 3(a), with the increase of dimension d, the performance of each model increases firstly and then decreases. This is because when d is too small, representations have insufficient capacity to capture rich information from temporal knowledge graph, and when d is too large, the model will be trapped in overfitting problem. Furthermore, we notice that with the representation dimension d changes, the performance of our ST-DistMult changes less compared with the other two models, which is because ST-DistMult can extract more effective information from TKG and thus it is more stable.

Number of Negative Sampling. As shown in Fig. 3(b), by comparing the performance of ST-DistMult, DE-DistMult and DistMult with different negative sampling numbers, we observe that with the increase of negative sampling number n, the performance of each model increases consistently. This is because a larger negative sampling number can provide more positive-negative pairs for each model to learn, and thus provide more information. However, we notice that when n is large, keep on increasing n leads to small performance improvement of ST-DistMult, which is because obtaining negative facts by random sampling can only provide coarse-grained information. Furthermore, compared with the other two models, ST-DistMult can still achieve significant performance when n is small, which demonstrates our ST-DistMult is able to obtain richer representations and thus each positive-negative pair can provide more information for model to learn.

6 Conclusion

In this work, we study the temporal knowledge graph completion task. We take temporal knowledge graph as a kind of spatial-temporal resource, and develop a spatial-temporal attention network which is able to obtain representation for each element of TKG while considering the graph structure and time-evolving property of TKG simultaneously. Our model contains a multi-faceted graph attention network used to capture structural information of each snapshot, and an adaptive temporal attention mechanism to model the temporal correlation of different snapshots. The representations obtained by our model can be used in the score function of existing static knowledge graph methods and result in the spatial-temporal version of these methods for the TKGC task. We test our proposed model on the link prediction task on three benchmark datasets. The experimental results show the superiority of our model and the effectiveness of each component in our model. In the future work, we aim to model the temporal correlation of TKG based on the structure evolution of egocentric network of each entity.

Acknowledgments. This work is supported by the National Nature Science Foundation of China (No. 61832001) and Sichuan Science and Technology Program (No. 2021JDRC0067 and No. 2019YFG0535).

References

1. Arnaboldi, V., Conti, M., Gala, M.L., Passarella, A., Pezzoni, F.: Ego network structure in online social networks and its impact on information diffusion. Comput. Commun. **76**, 26–41 (2016)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
3. Dasgupta, S.S., Ray, S.N., Talukdar, P.P.: HyTE: hyperplane-based temporally aware knowledge graph embedding. In: EMNLP, pp. 2001–2011 (2018)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI, pp. 1811–1818 (2018)

5. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing Wikidata to the linked data Web. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 50–65. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_4
6. García-Durán, A., Dumancic, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: EMNLP, pp. 4816–4821 (2018)
7. Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: AAAI, pp. 3988–3995 (2020)
8. Gupta, S., Yan, X., Lerman, K.: Structural properties of ego networks. In: Agarwal, N., Xu, K., Osgood, N. (eds.) SBP 2015. LNCS, vol. 9021, pp. 55–64. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16268-3_6
9. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL, pp. 687–696 (2015)
10. Jiang, T., et al.: Encoding temporal information for time-aware link prediction. In: EMNLP, pp. 2350–2354 (2016)
11. Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: autoregressive structure inference over temporal knowledge graphs. In: EMNLP (2020)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
13. Lacroix, T., Obozinski, G., Usunier, N.: Tensor decompositions for temporal knowledge base completion. In: ICLR (2020)
14. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Champin, P., Gandon, F.L., Lalmas, M., Ipeirotis, P.G. (eds.) WWW, pp. 1771–1776 (2018)
15. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, pp. 2181–2187 (2015)
16. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: ACL, pp. 4710–4723 (2019)
17. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.Q.: A novel embedding model for knowledge base completion based on convolutional neural network. In: NAACL-HLT, pp. 327–333 (2018)
18. Nickel, M., Rosasco, L., Poggio, T.A.: Holographic embeddings of knowledge graphs. In: AAAI, pp. 1955–1961 (2016)
19. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: ICML, pp. 809–816 (2011)
20. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
21. Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: NIPS, pp. 802–810 (2015)
22. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML, pp. 2071–2080 (2016)
23. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
24. Wang, S., Cao, J., Yu, P.S.: Deep learning for spatio-temporal data mining: a survey. IEEE Trans. Knowl. Data Eng. (2020). https://doi.org/10.1109/TKDE.2020.3025580
25. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119 (2014)

26. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI, pp. 7444–7452 (2018)
27. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
28. Zhang, J., Sheng, Y., Wang, Z., Shao, J.: TKGFrame: a two-phase framework for temporal-aware knowledge graph completion. In: Wang, X., Zhang, R., Lee, Y.-K., Sun, L., Moon, Y.-S. (eds.) APWeb-WAIM 2020. LNCS, vol. 12317, pp. 196–211. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60259-8_16
29. Zheng, C., Fan, X., Wang, C., Qi, J.: GMAN: a graph multi-attention network for traffic prediction. In: AAAI, pp. 1234–1241 (2020)