

---

# Sequence Encoder-based Spatiotemporal Knowledge Graph Completion

---

Wei Jia<sup>1,2</sup>, Xuan Wang<sup>1</sup>, Jing Shan<sup>2</sup>, Li Yan<sup>2</sup>,  
Weinan Niu<sup>2</sup> and Zongmin Ma<sup>2,\*</sup>

<sup>1</sup>*State Key Laboratory of Air Traffic Management System and Technology, China*

<sup>2</sup>*Nanjing University of Aeronautics and Astronautics, China*

*E-mail: zongminma@nuaa.edu.cn*

*\*Corresponding Author*

Received 07 May 2022; Accepted 10 September 2022;  
Publication 09 November 2022

## Abstract

Knowledge graph (KG) completion aims to infer new facts from incomplete knowledge graphs. Most existing solutions focus on learning from time-aware fact triples and ignore the spatial information. In reality, knowledge graphs can evolve with time as well as the changing locations, such as the flight domain. Therefore, integrating spatiotemporal information into knowledge graph representation is important for the knowledge graph completion. To address this problem, this paper proposes two SpatioTemporal-aware knowledge graph completion models based on the Sequence Encoder, namely STSE and S-TSE, which incorporate the spatial and temporal information into relations. Specifically, the model consists of two steps: spatiotemporal-aware relation encoding and final scoring function evaluation. The first stage composes the spatiotemporal information into different tokens. Then two methods are proposed to obtain the embedding of spatiotemporal-aware relation by utilizing the Recursive Neural Network. The second stage proposes different scoring functions for two models. Empirically evaluation of the

*Journal of Web Engineering, Vol. 21.6, 1913–1936.*

doi: 10.13052/jwe1540-9589.2166

© 2022 River Publishers

proposed models is conducted on spatiotemporal-aware KG completion task on two public datasets. Experimental results demonstrate the effectiveness of the proposal for spatiotemporal knowledge graph completion.

**Keywords:** Knowledge graph completion, recursive neural network, spatiotemporal information.

## 1 Introduction

In recent years, a large number of Knowledge Graphs (KGs) have been constructed, such as DBpedia [1], YAGO [2], and Freebase [3]. A KG is regarded as a multi-relational graph, where the nodes represent the entity and the edges denote the relationships between two entities. Because KGs have excellent expressiveness over structured knowledge, they have been found to be useful for various applications including information extraction [4], question answering [5, 35], and recommendation systems [6]. However, KGs are far from complete. For example, in Freebase, more than 70% of person entities have an unknown place of birth. Therefore, knowledge graph completion (KGC), i.e., automatically predicting missing entities for incomplete queries from a knowledge graph has been extended to an increasingly important task.

Over the last few years, most existing KG completion models mainly focus on two categories: static KGC [9–18] and temporal KGC [19–27]. In detail, a fact in the static KG is modeled as an RDF (Resource Description Framework) triple (*subject*, *predicate*, *object*), denoted as  $(s, p, o)$ , where *subject* and *object* are called *head entity* and *tail entity*, respectively. Specifically, the static KGC ignores the temporal information, which cannot capture the temporal validity of facts. Recent advancements in the KGC task have been extended to a more challenging domain of temporal KGs. In general, a temporal fact in a temporal knowledge graph is modeled as (*subject*, *predicate*, *object*, *timestamp*), denoted as  $(s, p, o, t)$ . The solutions for temporal KGC encode the temporal information into relations or regard the temporal KG as a series of snapshots. Although the majority of existing approaches have obtained effectiveness for temporal KGC, they ignore the spatial information of facts. In fact, a huge number of entities and relations contain spatial and temporal information. For example, in the flight domain, the origin of flight A is B1 with time C1 and location (D1, D2), where D1 and D2 represent the longitude and latitude, respectively. Then, the destination of this flight A is B2 with time C2 and location (D3, D4). We can note that knowledge graphs can

evolve with time as well as the changing locations. Therefore, it is significant to integrate temporal and spatial information into the entity and relations for KGs. Hence, extensive studies have been done on spatiotemporal data for knowledge graphs, such as spatiotemporal RDF modeling [7, 8]. However, there are few efforts on spatiotemporal KGC. To deal with this limitation, spatiotemporal KGC mainly faces two challenging tasks: How to incorporate the spatiotemporal information into KG representation and how to define the scoring function to evaluate the plausibility of facts.

To address the above two challenges, this paper proposes two SpatioTemporal-aware knowledge graph completion models based on the Sequence Encoder, named STSE and S-TSE for the SpatioTemporal-aware Knowledge Graph completion task. More specifically, utilizing the Recursive Neural Network (RNN) to train spatial information and temporal information together or not, the STSE model and S-TSE model are proposed, respectively. These two models consist of two steps: spatiotemporal-aware relation encoding and final scoring function evaluation for triples. As for the first stage, the two models propose the ST and S-T method to learn the spatiotemporal-aware relation representations by training an RNN with sequences of tokens that represent the relation, timestamp, and location. And the last hidden state of the recurrent network represents the spatiotemporal-aware relation representation. As for the second stage, the models define two different novel scoring functions to evaluate the plausibility of facts by combining the representations obtained in the first stage and the existing scoring function-based KGC methods, such as TransE [9]. The contributions of this work can be concluded as follows.

- This manuscript draws attention to the important but relatively unexplored domain of spatiotemporal-aware Knowledge Graph Completion. In particular, based on the RNN, two models are proposed to encode the relation with spatial and temporal information.
- Two extension scoring functions are designed for spatiotemporal knowledge graph completion.
- Extensive experiments are conducted on two public datasets, which demonstrates the effectiveness of the proposal.

The remainder of this paper is organized as follows. Section 2 reviews the related work to knowledge graph completion. Then, Section 3 provides the details of two spatiotemporal-aware models. In Section 4, experiments are conducted to evaluate the performance of the proposal. Finally, the conclusions and future directions are introduced in Section 5.

## 2 Related Work

This section mainly overviews the related works in three aspects. The first work is KG completion models for static KG. Then, the KG completion models for temporal KG are introduced. The spatiotemporal data models in KG are described at the end of this section.

### 2.1 Static Knowledge Graph Completion

Static knowledge graph completion mainly consists of two steps: knowledge graph embedding and scoring function, which aims to learn the representations of entity/relation and utilizes a scoring function to measure the plausibility of each fact. Specifically, as to the representations of entity and relation, almost all the studies jointly learn their embeddings. As to scoring function, these solutions can be classified into three branches, including distance matching, semantic matching, and others. Distance-based models mainly focus on computing the Euclidean distance between the projections of entities and relations, such as TransE [9], RotatE [10], TransMS [11], and KG2E [12], etc. For example, TransE [9] embeds entities and relations in  $d$ -dimension vector space. This model supposes that the embedding of the tail entity should be translated by the corresponding head entity and relation. Specifically, based on the TransE, some extensions have been presented, including TransR [13] and TransH [14]. In addition, RotatE [10] regards each relation as a rotation from head entity to tail entity. Another direction is semantic matching, which aims to calculate semantic similarity. For example, HolE [15] utilizes the tensor product to obtain the compositional representations. Utilizing the diagonal matrix to denote the relation matrix, Yang et al. [16] proposed the DistMult model. In addition, Trouillon et al. [17] presented the complex embedding model, named ComplEx. The main idea of this model is to handle a large variety of binary relations by matrix factorization. Moreover, Schlichtkrull et al. [18] proposed Relational Graph Convolutional Networks (R-GCN), which utilizes the relation-specific transformation to model the directed nature of knowledge graphs. Lv et al. [33] proposed a new benchmark setting that refines a novel pre-trained language models (PLM)-based model. The proposed model can better infer the implicit knowledge hidden in the PLM's parameters. However, all these methods ignore the temporal information and spatial information of facts, which are not suitable for spatiotemporal knowledge graphs.

## **2.2 Temporal Knowledge Graph Completion**

For the facts in a knowledge graph may be associated with a specific period, the temporal graph completion has attracted an increasing number of researchers. According to whether the representation of entities and relations is joint learning, this subsection overviews the related works from two aspects: non joint embedding and joint embedding.

Combining the temporal information and relations, ChronoR [20] supposes that each head entity can be transformed into a tail entity by rotation. To capture the dynamical entity embeddings, study [21] proposed the TeRo model. This model regards the temporal evolution of entity embedding as a rotation from the initial time to the current time in the complex vector space. In addition, TA-TransE [23] composes a given timestamp into a sequence, and then it combines this sequence and relations by Long Short-Term Memory (LSTM). Due to the attention mechanism can automatically capture the facts' relevance by selectively concentrating on a few important aspects, Han et al. [29] presented a temporal relational attention mechanism. In detail, they regard the reasoning process as the extraction of an enclosing subgraph around the query. However, the above models can not satisfy both the entity completion and relation completion. To address this problem, joint embeddings of entity and relation-based works have been proposed.

Xu et al. [19] incorporate time information into entity or relation embeddings and proposed the ATiSE model. This model regards the evolution of entity or relation as a multi-dimensional additive time series. And it supposes the additive time series consists of three components, namely trend component, seasonal component, and random component. Different from study [19], Xu et al. [22] divide the temporal knowledge graph into a sequence of graphs first. Then, they utilize the Markov chain to update the parameters in the adjacent graphs. Study [24] aims to learn the embeddings by capturing the temporal consistency and the contextual consistency. In addition, HyTE [25] divides the temporal knowledge graph into many subgraphs by the timestamps. Then it focuses on projecting the entities/relations of each subgraph into specific hyperplanes. At last, TransE is utilized to learn the entity/relation embeddings. Incorporating the temporal smoothness, RTGE [26] preserves structural information and evolutionary patterns for each graph. From a new perspective, Zhu et al. [27] proposed the CyGNet model. This model mainly depends on two components, namely Copy mode and Generation mode. In detail, the Copy mode and Generation mode are utilized to infer the entity

probabilities from the known entity vocabulary and the whole entity vocabulary, respectively. To consider the evolving strength of temporal ordering relations, Zhang et al. [32] presented a novel two-phase framework, namely TKGFrame. This framework mainly contains two models, named relation evolving enhanced model and refinement model. The attention mechanism attempts to selectively concentrate on a few important aspects that can automatically capture the facts' relevance. Considering the attention mechanism, Jung et al. [30] proposed the Temporal GNN with Attention Propagation (T-GAP) model, which contains an encoder and a decoder to consider both temporal information and graph structure. This model performs path-based inference by propagating attention through the graph. Therefore, utilizing the inferred attention to denote the natural interpretation of the prediction. When addressing the prediction task, human beings usually select useful information. Li et al. [31] proposed the CluSTeR model that utilizes the attention mechanism to predict future facts. In detail, the CluSTeR model mainly contains two stages: clue searching and temporal reasoning. As for the clue searching stage, it utilizes the reinforcement learning to induce multiple clues from historical facts. As for the temporal reasoning stage, it utilizes the graph convolution network to deduce answers from clues. In addition, Jin et al. [34] proposed the Recurrent Event Network (RE-NET) architecture. The RE-NET mainly contains an event sequence encoder and a neighborhood aggregation module. As for the neighborhood aggregation module, the RE-NET defines three aggregators, including mean aggregator, attentive aggregator, and graph convolutional aggregator. However, all these methods mainly focus on the temporal information and ignore the spatial information of facts, which are not suitable for spatiotemporal knowledge graphs.

### 2.3 Spatiotemporal Data Model

Recently, some researchers try to combine spatial data with temporal data to generate spatiotemporal data. Lu et al. [28] proposed a spatiotemporal fuzzy approach to model the complex nonlinear distributed parameter systems. They proposed two models to denote nonlinear spatial dynamics and nonlinear temporal dynamics. Bai et al. [8] presented the fuzzy spatiotemporal RDF data model, which is a 6-tuple,  $FSP = (s, p, o, FL, FT, \mu)$ , where  $FL$  and  $FT$  represent the location and time interval. From the above analysis, we can note that these methods aim to expand RDF triple into quad or quintuple and they generate extra labels, which may result in overhead for the system.

To deal with this problem, Zhu et al. [7] proposed a novel model, named stRDFS model. This model can be denoted as  $f = (s, p: \langle t, l \rangle, o)$ , where  $s$ ,  $p$ , and  $o$  are head entity, relation, and tail entity, respectively. Specifically,  $t$  and  $l$  are temporal information and spatial information. Inspired by these models, this paper maps the facts in the spatiotemporal knowledge graph into stRDFS. Then, we expect to complete the spatiotemporal knowledge graph completion task.

### 3 The Proposal

This section first gives some definitions mentioned in this article. Then, the proposal is described in detail.

#### 3.1 Definitions and Task Description

By extending the RDF data model  $(s, p, o)$ , study [7] proposed a novel spatiotemporal data model, namely stRDFS. In detail, the specific definition of stRDFS is described as follows:

**Definition 1 (A spatiotemporal data model: stRDFS).** The stRDFS model can be regarded as a triple, i.e.,  $\text{stRDFS} = (s, p: \langle t_p, l_p \rangle, o)$ , where

- $s$ ,  $p$ , and  $o$  denote the subject (i.e., head entity), predicate (i.e., relation), and object (i.e., tail entity), respectively.
- $t$  and  $l$  represent the temporal information and spatial information, respectively. In detail,  $t_p$  and  $l_p$  are the temporal data and spatial data relevant to the predicate  $p$ , respectively.

**Definition 2 (SpatioTemporal Knowledge Graph: STKG).** An STKG is defined as a graph  $G = (E, R, T, L)$ , where  $E$  represents the set of entities (i.e., nodes),  $R$  is the set of relations (i.e., edges),  $T$  and  $L$  are the valid temporal information and location information, respectively.

**Definition 3 (Fact).** The STKG consists of a series of facts, i.e., node-edge-node. Specifically, each fact can be formalized as a tuple  $\text{stRDFS} = (s, p: \langle t_p, l_p \rangle, o)$ , where  $s, o \in E$ ,  $p \in R$ ,  $t \in T$ , and  $l \in L$ .

**Definition 4 (SpatioTemporal Knowledge Graph Completion: STKGC).** The STKGC task aims to infer the missing object of an incomplete fact  $(s, p: \langle t_s, l_s \rangle, ?)$  or subject of an incomplete fact  $(?, p: \langle t_s, l_s \rangle, o)$  from the given spatiotemporal knowledge graph.

### 3.2 The Proposed Models for Spatiotemporal Knowledge Graph Completion

This part gives a detailed introduction to the proposed models for spatiotemporal knowledge graph completion. Given an STKG, the models first aim to learn representations for entities and relations. Then, utilizing a scoring function to evaluate the plausibility of facts, such that true facts receive high scores. Thus, the representations of entities and relations can be learned by optimizing an appropriate cost function. At last, according to the learned embeddings of each entity and relation, the models can predict the missing entities.

Different from the previous works that ignore the spatial information of facts, in this paper, two models are presented to incorporate spatiotemporal information into facts, namely STSE and S-TSE. Firstly, the two models utilize the LSTM model to capture the sequential data, i.e., relation, temporal tokens, and spatial tokens. Then, to increase usability, the two models apply the existing static KGC’s scoring functions to evaluate the plausibility of facts.

In detail, an overview of the proposed model is illustrated in Figure 1. As can be seen from Figure 1, there are mainly two different components in the STSE and S-TSE, namely relation encoding and scoring function. As for the relation encoding stage, the STSE supposes that the relation, temporal information, and location information are inseparable. Thus, it integrates spatial and temporal information together into relation. In detail, it processes the relation, temporal information, and location information serially by the LSTM model. Another model, namely S-TSE, supposes that the temporal information and location information are independent. Thus, the S-TSE

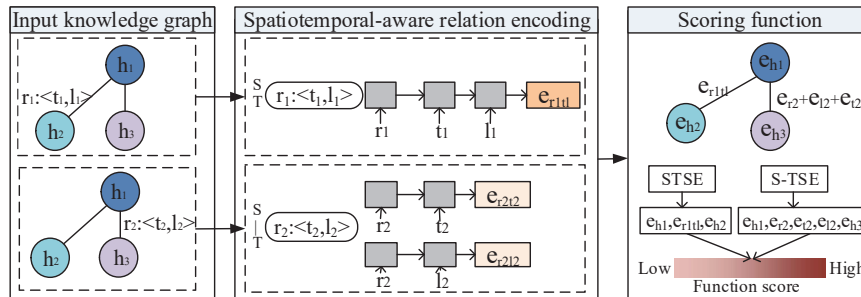


Figure 1 Overview of STSE and S-TSE.

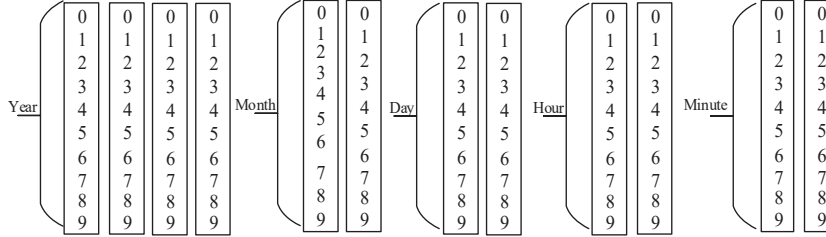
utilizes two LSTM models to encode the spatial-aware relation and temporal-aware relation, respectively. As for the scoring function, based on the TransE model [9], we design different scoring functions for two models.

### 3.2.1 Spatiotemporal-aware relation encoding

One of the main components of the proposed models is to learn the representations of relations over spatiotemporal information. Given a fact  $(h_1, r_1: \langle t_1, l_1 \rangle, h_2)$ , two methods are designed to incorporate spatial and temporal information into relations to learn the relation embedding, namely Spatiotemporal together sequence encoding (ST) and Spatiotemporal independent sequence encoding (S-T). These two methods mainly consist of two stages: spatiotemporal information decomposition and spatiotemporal-aware relation encoding. As for the first stage, the two methods decompose the spatial and temporal information into the sequence of spatiotemporal tokens. As for the second stage, the two methods utilize different ways to encode the relation. In detail, the ST supposes that the relation, temporal information and location information are inseparable. Thus, it integrates spatial and temporal information together into relation. Specifically, it processes the relation, temporal information, and location information serially by the LSTM model. Another model, namely S-T, supposes that the temporal information and location information have independent impacts on the relation. Thus, the S-T utilizes two LSTM models to encode the spatial-aware relation and temporal-aware relation, respectively. The detailed description of the two stages for each method is given as follows.

#### 3.2.1.1 Spatiotemporal information decomposition

This stage aims to decompose the spatiotemporal information into the sequence of temporal tokens. As for this stage, the first challenge is how to decompose the temporal and spatial information into tokens. And the second challenge is to choose the length of the total tokens. Study [23] proposed a novel way to decompose the temporal information into 32 tokens, however, it faces two shortcomings. The first one is that it only processes the year, month, and day, ignoring the fine-grained temporal information, such as minute and hour. In addition, it utilizes the 1 to 12 to represent the month tokens, which is not inconsistent with the day tokens that are 0 to 9. Therefore, this paper presents a new way to decompose the temporal information into five parts, including year, month, day, hour, and minute. Each part can be denoted by ten different tokens (i.e., 0 to 9), which is shown in Figure 2.



**Figure 2** Temporal tokens are decomposed by Year, Month, Day, Hour, and Minute.

For the first challenge, the two models decompose the temporal information into 50 tokens. From Figure 2, we can note that the tokens scope for year, month, day, hour, and minute range from 0 to 9, 10 to 19, 20 to 29, 30 to 39, and 40 to 49, respectively. For the second challenge, the two models set the length of the sequence tokens for each temporal information to 12, where the length of the year, month, day, hour, and minute is 4, 2, 2, 2, and 2, respectively. For example, as for a fact (B-2006, destination:<2019-07-31 00:10, (18, 23)>, EDDM), the model first decomposes the temporal information (i.e., 2019-07-31 00:10) into the following tokens [2, 0, 1, 9, 10, 17, 23, 21, 30, 30, 41, 40]. Then, the model decomposes the spatial information i.e., (18, 23) into two tokens [18, 23]. Because the number of temporal tokens is 50, the model utilizes the [68, 73] (i.e.,  $18 + 50 = 68$ ,  $23 + 50 = 73$ ) to represent the final spatial information.

### 3.2.1.2 Spatiotemporal-aware relation encoding

This stage focuses on encoding the spatiotemporal-aware relation to obtain the relation embedding. The main idea of this stage is regarding the relation, temporal information, and spatial information as sequence tokens, which is the input of RNN. This paper proposes two methods, namely ST and S-T to capture the relation, temporal information, and spatial information. Because the tokens can be regarded as sequence data, the solutions should adopt the Recurrent Neural Network. The LSTM model is a neural network architecture, which is suitable for processing sequential data. Therefore, this paper utilizes the LSTM to model the relation and its spatiotemporal tokens. In detail, these sequences of tokens are used as input to the LSTM. And the output is adopted as the spatiotemporal-aware relation embedding. Specifically, the output  $h_t$  of the LSTM unit is described as follows.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (1)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{6}$$

Where  $i_t$ ,  $f_t$ , and  $o_t$  are input gate, forget gate, and output gate, respectively. Specifically,  $W_i$ ,  $W_f$ , and  $W_o$  are the trainable weight matrices for the input gate, forget gate, and output gate, respectively. Similarly,  $b_i$ ,  $b_f$ ,  $b_c$ , and  $b_o$  are the bias vectors. The memory  $c_t$  is updated by forgetting part of the existing memory  $c_{t-1}$  and new memory  $\tilde{c}_t$ . In addition,  $\sigma$  is a sigmoid function, and  $\tanh$  is an activation function.

Specifically, the difference between these two methods is whether or not to encode the spatial information and temporal information together into relation. As for the ST method, the input sequence of the LSTM is [relation, temporal tokens, and spatial tokens], which combines the temporal and spatial tokens at the same time. And the output of LSTM is regarded as the embeddings of spatiotemporal-aware relation. Different from the ST, the S-T adopts two LSTM models. The input sequence of the two LSTM models is [relation, temporal tokens] and [relation, spatial tokens], which represent temporal-aware relation and spatial-aware relation, respectively. Figure 3 shows the process of the spatiotemporal-aware relation representation for ST and S-T.

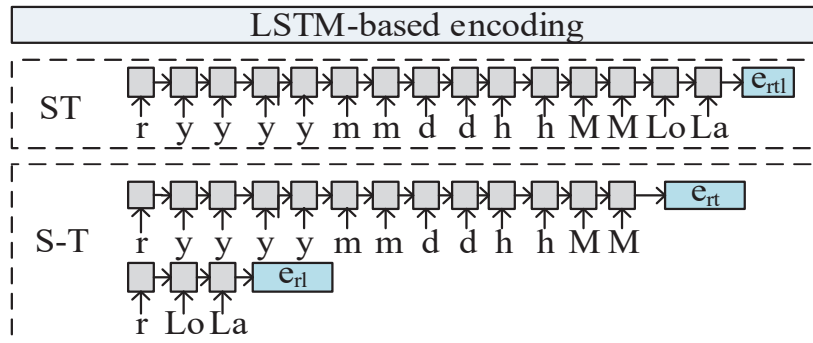


Figure 3 The calculation of spatiotemporal-aware relation.

### 3.2.2 Scoring function

Inspired by one of the most popular translation-based models TransE [9], in this paper, the STSE and S-TSE have the following scoring function for a fact  $(s, r : \langle t, l \rangle, o)$ .

$$\text{STSE: } f(s, r : \langle t, l \rangle, o) = \|e_s + e_{rtl} - e_o\|_{l1/l2} \quad (7)$$

$$\text{S-TSE: } f(s, r : \langle t, l \rangle, o) = \|e_s + e_{rt} + e_{rl} - e_o\|_{l1/l2} \quad (8)$$

Where  $e_r, e_{rtl}, e_{rt}, e_{rl}, e_o \in R^d$  and  $e_s, e_o$  are the representations for head entity  $s$  and tail entity  $o$ . In addition,  $e_{rtl}, e_{rt}$ , and  $e_{rl}$  denote the representation for the spatiotemporal-aware relation, temporal-aware relation, and spatial-aware relation, respectively.

As the task is a binary classification problem, the models utilize the binary cross entropy as the objective function to optimize the proposed two models. The loss function can be calculated as follows.

$$L = - \sum_{x \in F^+ \cup F^-} y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (9)$$

where,  $F^+$  denotes the positive dataset directly obtained from the training dataset, and  $F^-$  is the negative dataset which is sampled by randomly corrupting the subjects or the objects. In addition, the  $\hat{y}$  for STSE and S-TSE can be obtained as follows.

$$\begin{aligned} \hat{y}_{STSE} &= \text{soft max}(\text{sigmoid}(f(s, r : \langle t, l \rangle, o))) \\ &= \text{soft max}(\text{sigmoid}(\|e_s + e_{rtl} - e_o\|)) \end{aligned} \quad (10)$$

$$\begin{aligned} \hat{y}_{S-TSE} &= \text{soft max}(\text{sigmoid}(f(s, r : \langle t, l \rangle, o))) \\ &= \text{soft max}(\text{sigmoid}(\|e_s + e_{rt} + e_{rl} - e_o\|)) \end{aligned} \quad (11)$$

## 4 Experimental Results

This section mainly evaluates the performance of the proposals on two types of datasets that contain both temporal information and spatial information.

### 4.1 Experimental Setup

#### 4.1.1 Datasets

We evaluate the proposed models for the knowledge graph completion task on two real-world public datasets that are collected from OpenSky and

**Table 1** The detailed information of datasets

	Time Interval	#Fact	#Entity	#Relation	#Longitude Value	#Latitude Value
OpenSky	Aug 1, 2019 – Aug 7, 2019	857,993	59,348	2	399,170	463,218
YAGO	Apr 9, 1054 – Dec 26, 1899	8,391	10,677	1	2,191	2,233

YAGO, respectively. As for the first dataset, it can be downloaded from <https://zenodo.org/record/5377831#.YkP9pC1ByUk>. It is derived and cleaned from the full OpenSky dataset to introduce the development of air traffic during the COVID-19 pandemic. The original dataset contains several features, such as callsign, number, icao24, registration, typecode, origin, destination, firstseen, lastseen, day, latitude\_1, longitude\_1, and altitude\_1. In the experiment, we extract the weekly spatiotemporal data from the file (i.e., flightlist\_20190801-20190831.csv) to collect the relevant information from Aug 1, 2019 to Aug 7, 2019. And the fact of spatiotemporal graph is built as (registration, destination/origin:<day, (latitude\_1, longitude\_1), destination/origin airport code>). As for the YAGO dataset, we download the YAGO 2.0.0\_core version from <https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yagonaga/yago>. This native format contains a set of files. In these files, each fact is composed of a subject, a predicate, and an object. In the experiment, we extract the facts associated with a temporal dimension and a spatial dimension, such as the file “wasBornIn.tsv” and “wasBornOnDate.tsv”. In addition, we utilize the “hasGeoCoordinates.tsv” file to capture the longitude and latitude of locations. Specifically, the detailed information of these datasets is shown in Table 1.

#### 4.1.2 Baselines

There are a few studies related to spatiotemporal knowledge graph completion. Thus, in this section, we utilize the classic temporal knowledge graph completion model TA-TransE as a baseline. Specifically, the TA-TransE model also utilizes the RNN to encode the relation with temporal tokens, which is similar to our proposal. Although our model and TA-TransE both utilize a similar way to process the temporal information, the TA-TransE only contains the year, month, and day while ignoring the effect of the hour and minute, which may be not accurate. In addition, the TA-TransE ignores spatial information. Therefore, in this experiment, we remove the spatial information from the datasets and only retain the time information to serve the TA-TransE for knowledge graph completion.

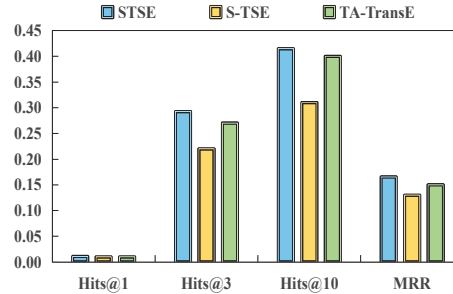
### 4.1.3 Implementation details

In this experiment, the embedding dimension of entity/relation is set to 200 and 300 for OpenSky and YAGO datasets, respectively. Adam is utilized as an optimizer. And the learning rate is chosen from the set  $\{0.0001, 0.001, 0.01\}$ . The epoch is a parameter, which is defined as an iteration over the entire dataset. Therefore, the training epoch is chosen from the set  $[10, 30, 50]$  and  $[10, 20, 30]$  for OpenSky and YAGO, respectively. The batch size is set to 1024 for the OpenSky dataset and 128 for the YAGO dataset. In addition, for each dataset, we split it into three subsets, i.e., train (80%), valid (10%), and test (10%). We report Mean Reciprocal Ranks (MRR) and Hits@1/3/10 to evaluate the performance of the proposal. In detail, the MRR is the average of the reciprocal ranks. And the fraction of test quadruples ranking in the top  $k$  is called Hits@ $k$ , where  $k$  is equal to 1, 3, and 10. Specifically, we implement all the models in Pytorch and train on GeForce GTX 2080 Ti.

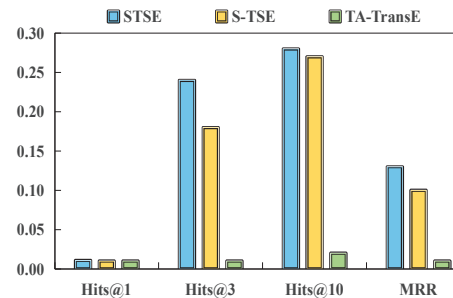
## 4.2 Experimental Results Analysis

### 4.2.1 Model evaluation for KGC

As mentioned in Section 3, two different models are proposed for spatiotemporal knowledge graph completion. In this section, we evaluate the performance of the proposed two models in terms of Hits@1, Hits@3, Hits@10, and MRR metrics. Specifically, the task is to predict the missing entity when an incomplete relational fact with its temporal and spatial information is provided. Figure 4 shows the prediction results of the proposed two models and TA-TransE model on two datasets. From Figure 4, we can observe that the STSE obtains better performance than the TA-TransE on both two datasets. The reason may be that combining the spatial information and temporal information can provide more useful information for the KGC task, which proves the superiority of considering the spatial information. From Figure 4, we can observe that STSE consistently outperforms S-TSE in terms of all metrics. In detail, as for the OpenSky dataset, the STSE is 31.82%, 33.33%, and 30.77% better than S-TSE on Hits@3, Hits@10, and MRR, respectively. In addition, as for the YAGO dataset, the STSE achieves the improvements of 33.33% and 3.70% on Hits@3 and Hits@10. By combining Figures 4(a) and 4(b), we can observe that the STSE and S-TSE have better performance on OpenSky than YAGO. The reason may be that the scale of the OpenSky is much larger than that of the YAGO. Overall result, we can conclude that considering the temporal information and spatial information together can improve the performance for spatiotemporal knowledge graph completion.



(a) OpenSky dataset



(b) YAGO dataset

**Figure 4** Results of the spatiotemporal knowledge graph completion on two datasets.

In addition, in order to have a clear explanation, the performance of the STSE, S-TSE, and TA-TransE models are evaluated by head entity prediction and tail entity prediction, which are shown in Table 2. From Table 2, we have two observations. Firstly, tail entity prediction performs better than head entity prediction for STSE, S-TSE, and TA-TransE models. By analyzing the two datasets, we find that the number of head entities and tail entities are 50,431 and 8,917 for OpenSky. And in the YAGO dataset, the number of head entities and tail entities are 8,389 and 2,287. Therefore, the reason may be that the number of entities may influence the results, and a smaller number of entities may lead to better performance for the same model. Secondly, on the OpenSky dataset, the TA-TransE and STSE obtain similar results and their performances are better than the S-TSE for entity completion. In addition, on the YAGO dataset, the STSE performs better than the S-TSE and TA-TransE in the aspects of head and tail entity prediction. This result proves that incorporating the temporal and spatial information into relation together could improve the performance.

**Table 2** The results on head entity prediction and tail entity prediction

	OpenSky						YAGO					
	Head			Tail			Head			Tail		
	Hits@3	Hits@10	MRR	Hits@3	Hits@10	MRR	Hits@3	Hits@10	MRR	Hits@3	Hits@10	MRR
STSE	<b>14.08%</b>	23.50%	8.82%	<b>42.04%</b>	<b>53.86%</b>	<b>22.68%</b>	<b>0.12%</b>	<b>0.36%</b>	<b>0.18%</b>	<b>43.43%</b>	<b>51.10%</b>	<b>22.44%</b>
S-TSE	3.43%	8.52%	3.14%	41.08%	51.82%	22.04%	0.01%	0.01%	0.01%	40.51%	50.49%	17.63%
TA-TransE	13.29%	<b>25.72%</b>	<b>8.91%</b>	34.10%	44.97%	18.66%	0.01%	0.01%	0.01%	0.36%	1.46%	0.56%

**Table 3** The results of STSE with different epochs on three metrics on two datasets

	Epoch	Hits@3	Hits@10	MRR
OpenSky	<b>10</b>	<b>30.76%</b>	<b>42.40%</b>	<b>17.25%</b>
	30	29.26%	41.48%	16.53%
	50	29.35%	42.12%	16.71%
YAGO	10	21.35%	23.78%	11.08%
	<b>20</b>	<b>24.09%</b>	<b>27.80%</b>	<b>12.50%</b>
	30	13.38%	20.38%	7.84%

## 4.2.2 Hyperparameters sensitivity

From Section 4.2.1, we can note that the STSE performs better than the S-TSE. This section validates the robustness of STSE by varying the epoch and learning rate in knowledge graph completion on Hits@3, Hits@10, and MRR.

### 4.2.2.1 The effect of Epoch

This section validates the effect of the epoch on the STSE model for the entity completion task. To do this, the training epoch is chosen from the set [10, 30, 50] for OpenSky and [10, 20, 30] for YAGO. In addition, we calculate the Hits@3, Hits@10, and MRR on the two datasets. The experimental result is shown in Table 3. Interestingly, STSE can obtain higher performance for smaller epoch numbers. In detail, when the number of epochs equals 10, the STSE is 2.22% and 0.66% better than 30 and 50 on Hits@10 on the OpenSky dataset. In addition, as for the YAGO dataset, the STSE can obtain the best performance when the number of epochs equals 20. Therefore, we can conclude that the number of the epoch may lead to different influences on the STSE model.

### 4.2.2.2 The effect of the learning rate

This section validates the effect of the learning rate on the STSE model for entity completion tasks. To do this, the learning rate is chosen from the set [0.001, 0.01, 0.1]. From Table 4, with the increase in the learning rate,

**Table 4** The results of STSE with different learning rates

	Learning Rate	Hits@3	Hits@10	MRR
OpenSky	0.0001	8.65%	16.73%	6.09%
	<b>0.001</b>	<b>30.76%</b>	<b>42.40%</b>	<b>17.25%</b>
	0.01	4.15%	6.87%	2.75%
YAGO	0.0001	11.74%	13.08%	6.08%
	<b>0.001</b>	<b>24.09%</b>	<b>27.80%</b>	<b>12.50%</b>
	0.01	0.00%	0.06%	0.09%

the performance of the model first increases and then decreases. When the learning rate equals 0.001, the STSE model can obtain a better performance than others on both two datasets. In detail, when the learning rate is equal to 0.001, the STSE can obtain 42.40% and 27.80% on OpenSky and YAGO in the aspect of Hits@10.

## 5 Conclusion

This paper proposes two models for spatiotemporal knowledge graph completion, called STSE and S-TSE. The proposed model is advantageous with its capability in modeling the relation with spatial and temporal information. Besides, two methods are designed to learn the representation of the spatiotemporal-aware relation by the LSTM model. In addition, this paper also proposes two loss functions to complete the missing entities, which can be incorporated into the prior static KGC methods. We also evaluate the effect of several hyperparameters on the proposed model. The effect of the learning rate is larger than the epoch for the KGC task. Compared with the temporal knowledge graph completion model, experimental results demonstrate that considering the temporal information and spatial information together can improve the performance for spatiotemporal knowledge graph completion.

In future work, we have several directions. Considering the spatial dependence and temporal dependence to learn the spatiotemporal knowledge embedding, which is useful for spatiotemporal knowledge graph completion. Another direction is considering semantic information (i.e., category, characteristic, etc) of entities and relations to better learn the embeddings of the knowledge graph. In addition, we can note that the proposed model cannot satisfy both entity completion and relation completion. In future work, we tend to propose a novel model to jointly learn embeddings of the knowledge graph's entities and relations. We expect that the proposed models could

motivate further research on spatiotemporal knowledge graph representation learning and spatiotemporal knowledge completion.

## Acknowledgments

The work was supported in part by the Foundation of State Key Laboratory of Air Traffic Management System and Technology (SKLATM202104), the National Natural Science Foundation of China (62176121), the Basic Research Program of Jiangsu Province (BK20191274), and the National Key Research and Development Program of China (2020YFB1600100).

## References

- [1] Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., and Ives Z., 'Dbpedia: a nucleus for a web of open data', Proc. In the 6th International Semantic Web Conference, pp. 722–735, 2007.
- [2] Mahdisoltani F., Biega J., and Suchanek F., 'Yago3: A knowledge base from multilingual wikipedias', Proc. In the Seventh Biennial Conference on Innovative Data Systems Research, 2015.
- [3] Bollacker K., Evans C., Paritosh P., Sturge T., and Taylor J., 'Freebase: a collaboratively created graph database for structuring human knowledge', Proc. In the ACM SIGMOD International Conference on Management of Data, pp. 1247–1250, 2008.
- [4] Liu Z., Xiong C., Sun M., and Liu Z., 'EntityDuet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval', Proc. In the 56th Annual Meeting of the Association for Computational Linguistics, pp. 2395-2405, 2018.
- [5] Li D. Y., Sheng J., Liu Y. X., et al., 'Military weapon QA system based on knowledge graph', *Command Information System and Technology*, 11(5): 58–65, 2020.
- [6] Fan H., Zhong Y., Zeng G., et al., 'Improving recommender system via knowledge graph based exploring user preference', *Applied Intelligence*, <https://doi.org/10.1007/s10489-021-02872-8>, 2022.
- [7] Zhu L., N Li, Bai L., et al., 'stRDFS: Spatiotemporal Knowledge Graph Modeling', *IEEE Access*, 8: 129043-129057, 2020.
- [8] Bai L., J Wang, Di X. et al., 'Fixing the Inconsistencies in Fuzzy Spatiotemporal RDF Graph', *Information Sciences*, 578: 166–180, 2021.

- [9] Antoine Bordes, Nicolas Usunier, Alberto GarciaDuran, Jason Weston, and Oksana Yakhnenko, ‘Translating Embeddings for Modeling Multi-relational Data’, Proc. In the 27th Annual Conference on Neural Information Processing Systems, pp. 2787–2795, 2013.
- [10] Sun Z., Deng Z. H., Nie J. Y. et al., ‘RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space’, Proc. In the 7th International Conference on Learning Representations, 2019.
- [11] S. Yang, J. Tian, H. Zhang, J. Yan, H. He, and Y. Jin, ‘TransMS: Knowledge graph embedding for complex relations by multidirectional semantics’, Proc. In the 28th International Joint Conference on Artificial Intelligence, pp. 1935–1942, 2019.
- [12] He S., K. Liu, G. Ji, and J. Zhao, ‘Learning to represent knowledge graphs with Gaussian embedding’, Proc. In the 24th ACM Int. Conference Information and Knowledge Management, pp. 623–632, 2015.
- [13] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, ‘Learning entity and relation embeddings for knowledge graph completion’, Proc. In the 29th AAAI Conference on Artificial Intelligence, pp. 2181–2187, 2015.
- [14] Z. Wang, J. Zhang, J. Feng, and Z. Chen, ‘Knowledge graph embedding by translating on hyperplanes’, Proc. In the 28th AAAI Conference on Artificial Intelligence, pp. 1112–1119, 2014.
- [15] M. Nickel, L. Rosasco, and T. Poggio, ‘Holographic embeddings of knowledge graphs’, Proc. In the 30th AAAI Conference on Artificial Intelligence, pp. 1955–1961, 2016.
- [16] B. Yang, W.-T. Yih, X. He, J. Gao, and L. Deng, ‘Embedding entities and relations for learning and inference in knowledge bases’, Proc. In the 3rd International Conference on Learning Representations, pp. 1–13, 2015.
- [17] K. Hayashi and M. Shimbo, ‘On the equivalence of holographic and complex embeddings for link prediction’, Proc. In the 55th Annual Meeting of the Association for Computational Linguistics, pp. 554–559, 2017.
- [18] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, ‘Modeling relational data with graph convolutional networks’, Proc. In the 15th International Conference on Extended Semantic Web, pp. 593–607, 2018.
- [19] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Jens Lehmann, and Hamed Shariat Yazdi, ‘Temporal knowledge graph embedding model based on additive time series decomposition’, Proc. In the International Semantic Web Conference, 2020.

- [20] Sadeghian A., Armandpour M., Colas A. et al., ‘ChronoR: Rotation Based Temporal Knowledge Graph Embedding’, Proc. In the 35th AAAI Conference on Artificial Intelligence, pp. 6471–6479, 2021.
- [21] Xu C., Nayyeri M., Alkhoury F., et al., ‘TeRo: A Time-aware Knowledge Graph Embedding via Temporal Rotation’, Proc. In the 28th International Conference on Computational Linguistics, pp. 1583–1593, 2020.
- [22] Xu Y., Haihong E., Song M., ‘RTFE: A Recursive Temporal Fact Embedding Framework for Temporal Knowledge Graph Completion’, Proc. In the 2021 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 5671–5681, 2021.
- [23] García-Durán, A., Dumani S., Niepert M., ‘Learning Sequence Encoders for Temporal Knowledge Graph Completion’, Proc. In the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4816–4821, 2018.
- [24] Liu Y., Hua W., Xin K., et al., ‘Context-Aware Temporal Knowledge Graph Embedding’, Proc. In the 20th International Conference on Web Information Systems Engineering, pp. 583–598, 2019.
- [25] Dasgupta S. S., Ray S. N., Talukdar P., ‘HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding’, Proc. In the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2001–2011, 2018.
- [26] Xu Y., Sun S., Miao Y., et al., ‘Time-aware Graph Embedding: A temporal smoothness and task-oriented approach’, *ACM Transactions on Knowledge Discovery from Data*, 16(3): 1–23, 2022.
- [27] Zhu C., Chen M., Fan C., et al., ‘Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks’, Proc. In the Thirty-Fifth AAAI Conference on Artificial Intelligence, pp. 4732–4740, 2021.
- [28] Lu X. J., Hu T. T., Yin F., ‘A Novel Spatiotemporal Fuzzy Method for Modeling of Complex Distributed Parameter Processes’, *IEEE Transactions on Industrial Electronics*, 66(10): 7882–7892, 2019.
- [29] Han Z., Chen P., Ma Y. P., Tresp V., ‘Explainable subgraph reasoning for forecasting on temporal knowledge graphs’, Proc. In the International Conference on Learning Representations, 2020.
- [30] Jung J. H., Jung J. H., Kang U., ‘T-GAP: Learning to walk across time for interpretable temporal knowledge graph completion’, Proc. In the ACM SIGKDD conference on Knowledge Discovery and Data Mining, pp. 786–795, 2021.

- [31] Li Z., Jin X., Guan S., et al., ‘Search from History and Reason for Future: Two-stage Reasoning on Temporal Knowledge Graphs’, Proc. In the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021.
- [32] Zhang J., Sheng Y., Wang Z., Shao J., ‘TKGFrame: A Two-Phase Framework for Temporal-Aware Knowledge Graph Completion’, Proc. In the 4th Asia Pacific Web and Web-Age Information Management Joint Conference on Web and Big Data, 2020.
- [33] Lv X., Lin Y. K., Cao Y. X., et al. ‘Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach’, Proc. In the Association for Computational Linguistics, pp. 3570–3581, 2022.
- [34] Jin W., Zhang C. L., Szekely P., Ren X., ‘Recurrent Event Network for Reasoning over Temporal Knowledge Graphs’, Proc. In the International Conference on Learning Representations, 2019.
- [35] Liu H., Chen C. R., Xu Q. C., et al., ‘Question and answer robot’s technology for air traffic control’, *Command Information System and Technology*, 12(5): 32–37, 2021.

## Biographies



**Wei Jia** received the M.S. degree in computer science from Shandong University of Science and Technology, China in 2020. She is currently working toward the Ph.D. degree in the School of Computer Science and Technology at Nanjing University of Aeronautics and Astronautics, China. Her fields of research are social network, knowledge graph, service computing and intelligent information systems.



**Xuan Wang** received the B.E. degree from University of Hertfordshire, Hartfield, UK in 2011. Later on, he received his M.S. degree from Southampton University, Southampton, UK in 2012. He received his PhD degree in Electronics and Electrical Engineering from Southampton University in 2017. He is currently a senior engineer of State Key Laboratory of Air Traffic Management System and Technology, Nanjing, China. His research interests include information extraction and natural language understanding.



**Jing Shan** received the bachelor degree in information security in 2010 and the master degree in computer science in 2013 from Nanjing University of Aeronautics and Astronautics, China. She is a research associate at Nanjing University of Aeronautics and Astronautics, China, engaging in intelligent system for task allocation, optimization algorithm and application of intelligent transportation system.



**Li Yan** received her PhD degree from Northeastern University, China. She is currently a full professor at Nanjing University of Aeronautics and Astronautics, China. Her research interests mainly include big data processing, knowledge graph, spatiotemporal data management, and fuzzy data modeling. She has published more than fifty papers on these topics. She is the author of three monographs published by Springer.



**Weinan Niu** received the M.S. degree in computer science from Shandong University of Science and Technology, China in 2020. He is currently working toward the Ph.D. degree in the School of Computer Science and Technology at Nanjing University of Aeronautics and Astronautics, China. His fields of research are social network, influence maximization, service computing and intelligent information systems.



**Zongmin Ma** received his PhD degree from City University of Hong Kong, China. He is currently a full professor at Nanjing University of Aeronautics and Astronautics, China. His research interests include big data and knowledge engineering, the Semantic Web, temporal/spatial information modeling and processing, deep learning, and knowledge representation and reasoning with a special focus on information uncertainty. He has published more than two hundred papers in international journals and conferences on these topics. In addition, he (co-)authors six monographs with Springer. He is a Fellow of the IFSA and a senior member of the IEEE.