Expert Systems With Applications 201 (2022) 117036

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Block term decomposition with distinct time granularities for temporal knowledge graph completion

Yujing Lai [a], Chuan Chen [a,b,*], Zibin Zheng [a,b], Yangqing Zhang [c]

[a] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China
[b] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510275, China
[c] Merchants Union Consumer Finance Co., Ltd., Kexing Science Park, Shenzhen 518057, China

ARTICLE INFO

Keywords:
Knowledge graph completion
Temporal knowledge graph
Tensor completion

ABSTRACT

To solve the incompleteness problem in temporal knowledge graphs (TKGs) and to discover the new knowledge, TKG completion remains an essential task always solved by graph embedding technology. Existing TKG completion methods encode time at only single granularity, which is insufficient in exploiting the rich information of distinct time granularities. Furthermore, most of them lack a comprehensive consideration of the characteristic of both time points and time periods, resulting in the inability to handle the two types of facts with different time forms, namely the discrete facts and continuous facts, simultaneously. In this paper, we propose a novel TKG embedding model which introduces the block term tensor decomposition and utilizes the core tensor and factor matrices to capture information presented by facts under distinct time granularities. By focusing on moments included in the time period and treating the discrete fact as a special case of the continuous fact, the model manages the processing of different types of facts in a unified manner. Besides, we explicitly design the static properties of entities and relations as well as their interactions to conform to reality. Experiments on 3 real datasets of different types verify the effectiveness of our proposed method compared with most state-of-the-art methods.

1. Introduction

A knowledge graph (KG) is a semantic information database that describes the relationships between various entities in the real world. It is typically expressed in the form of a multi-relational directed graph, in which nodes represent entities and edges represent relations between entities. Any two connected nodes and the edge between them form a triple denoted as (subject, relation, object), abbreviated as (s, r, o) which is called a *fact*. Fig. 1 is an example of a knowledge graph, where (Washington, IsCapitalOf, American) represents a fact with subject Washington, relation IsCapitalOf and object American. Due to the rich semantic information and powerful representation capabilities, KGs have been applied to various fields, such as social networks (Molokwu et al., 2020), question answering (Kacupaj et al., 2021) and recommender systems (Lee et al., 2020).

However, because of the difficulty in collecting facts or even data loss, practical KGs often fail to be complete, severely limiting their usability and availability. In addition, the new knowledge is expected to be inferred from the existing old knowledge to enrich the KGs. To this end, an essential task named KG completion aims to predict the missing object or subject given other elements in the fact, i.e., answers $(s, r, ?)$ or $(?, r, o)$. Recently, a substantial amount of researches employing graph embedding technique to conduct KG completion have presented promising results, which embeds nodes and edges in the low-dimensional continuous vector space, such as TransE (Bordes et al., 2013), TransH (Wang et al., 2014) and RESCAL (Nickel et al., 2011).

Although there are numerous KG embedding methods, most of them are designed for static knowledge graphs (SKGs), while temporal knowledge graphs (TKGs) containing time information receive relatively less attention. In TKGs, the time is attached to each fact indicating when it is valid. According to the forms of the time, TKGs can be classified into two categories : (i) discrete fact knowledge graphs, in which the valid time of the fact is a time point, denoted as (s, r, o, t_p), for instance, ICEWS14 (García-Durán et al., 2018); (ii) continuous fact knowledge graphs, in which the valid time of the fact is a time period, denoted as $(s, r, o, [t_b, t_e])$ with t_b and t_e representing the beginning and the end of the valid time, for instance, Wikidata12k (Dasgupta et al., 2018). For the unity of forms, we denote both these two types of facts as (s, r, o, t).

* Corresponding author at: School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China.
E-mail addresses: laiyj23@mail2.sysu.edu.cn (Y. Lai), chenchuan@mail.sysu.edu.cn (C. Chen), zhzibin@mail.sysu.edu.cn (Z. Zheng), zhangyq@mucfc.com (Y. Zhang).

https://doi.org/10.1016/j.eswa.2022.117036
Received 24 October 2021; Received in revised form 18 January 2022; Accepted 27 March 2022
Available online 9 April 2022
0957-4174/

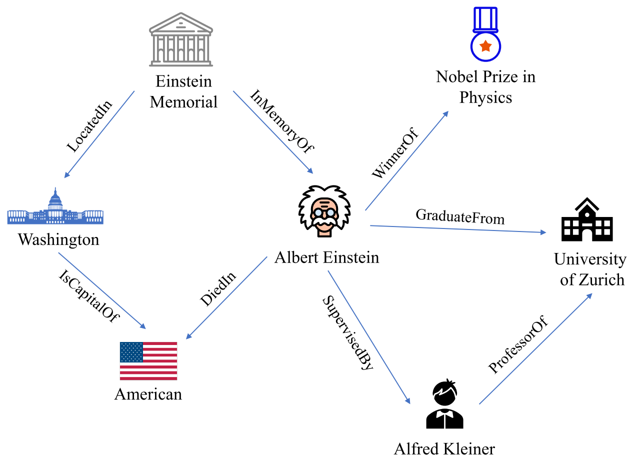


Fig. 1. An example of KG with various entities and relations.

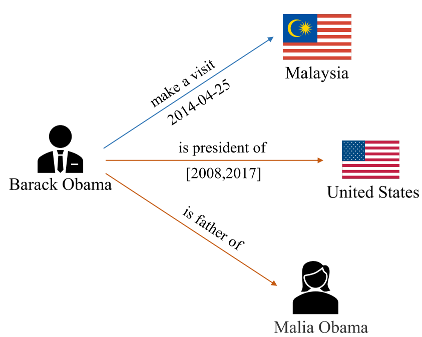


Fig. 2. Different types of facts in TKGs including the discrete fact, the continuous fact and the fact that remains true all the time, which are common in TKGs.

Unlike SKG embedding, TKG embedding is required to additionally learn the time properties, and how to leverage the time information effectively is exactly a crucial concern in the TKG completion. Existing TKG embedding methods generally encode time to obtain the time-specific representations for entities and relations, such as Hyte (Dasgupta et al., 2018), TA-DistMult (García-Durán et al., 2018), etc. However, these methods evaluate the plausibility of facts at a single time scale, neglecting that the occurrence of a temporal fact is affected by both the era background as well as the recent events. Besides, the time in the form of points or periods both provide significant information about the validity of facts, such as times in (Barack Obama, make a visit, Malaysia, 2014-04-25) and (Barack Obama, is president of, United States, [2008, 2017]) in Fig. 2 are crucial for political events. As the existing models are established for the discrete facts and then extended to the continuous ones, it leads to the limited consideration of the properties of time periods and the incapability in dealing with the discrete and continuous facts uniformly. Moreover, certain facts in TKGs, like (Barack Obama, is father of, Malia Obama) as shown in Fig. 2, remain true all the time, implying the constant interactions between entities and relations which are disregarded by most of the previous works. Therefore, the challenges of TKG completion problems include exploiting temporal information of distinct granularities, uniformly handling discrete and continuous facts, and considering static facts that are not affected by time. As shown in Fig. 2, these situations are common in TKG and it is necessary to develop a method for them.

A KG can be regarded as a third-order binary tensor, with each element 1 or 0 indicating whether the fact corresponding to the element's index is valid or invalid. Therefore, we consider adopting a powerful tensor decomposition method, the block term decomposition (BTD) (De Lathauwer, 2008), to model complex signal information, including static and temporal properties of facts, as well as discrete and continuous properties of time. In this paper, we propose a novel TKG embedding method utilizing *Block Term Decomposition with distinct time Granularities (BTDG)*, which extracts the information presented by facts under the two different time granularities and takes the constant interactions into account. Specifically, the block term decomposition is employed to explicitly model the interactions among temporal and static properties of entities and relations, with the core tensor and the component matrices to capture coarse-grained and fine-grained time information respectively, in which way discrete and continuous facts are processed in a unified manner. Therefore, our method considers both static and temporal properties of facts in TKG, and focuses on distinct granularity information in time, while providing a method for unified processing of discrete and continuous facts.

Our contributions are summarized as follows:

- We propose to encode the fine-grained and coarse-grained time information of facts under the two distinct granularities through the component matrices and the core tensor in BTD, leading to a more comprehensive evaluation of facts.
- We explicitly model the static properties of entities and relations as well as their interactions to conform to reality.
- The proposed model BTDG unifies the processing in handling discrete and continuous facts without the special preprocessing for certain types of data.
- Experiments conducted on three real-world datasets of different types exhibit that BTDG outperforms most previous state-of-the-art methods.

2. Related work

2.1. SKG embedding methods

Mainstream SKG embedding methods can be classified into three categories based on their loss functions: translational distance models, semantic matching models, and GNN-based models.

Translational distance models

Translational distance models propose that if a fact exists, the subject should be close to the object after the translation of the relation, with a function to measure the distance. The most typical method is TransE (Bordes et al., 2013). Due to the defect of TransE in modeling 1-n, n-1 and $n-n$ relations, many extensions based on TransE are proposed, including TransH (Wang et al., 2014), TransR (Lin et al., 2015), etc. For more powerful generalization, RotatE (Sun et al., 2019) regards the relation as the rotation of the subject embedding in the complex space, and KG2E (He et al., 2015) takes the uncertainties of entities and relations into consideration, modeling their representations as Gaussian distributions. ATransD-NL (Wang et al., 2021) learns attention-based embeddings taking into account the influence of one-hop or potentially multi-hop neighborhood entities and applies the nonlinear dynamic projection to capture nonlinear correlations.

Semantic matching models

Semantic matching models measure the plausibility of triples by matching the representations of entities and relations in the embedding space with a similarity-based scoring function. RESCAL (Nickel et al., 2011) is the representative method that uses the matrix to model the relation. DistMult (Yang et al., 2015) simplifies RESCAL by restricting the relation matrix to be diagonal. However, neither RESCAL nor DistMult can model asymmetric relations. To combine the advantages of simplicity and powerful expressive ability, SimplE (Kazemi & Poole, 2018) and ComplEx (Trouillon et al., 2016) are proposed. MEI (Tran & Takasu, 2020) proposes the multi-partition embedding interaction model with the block term format, which learns the interaction mechanism automatically through the Tucker core tensors.

Since a KG can be regarded as a 0–1 third-order tensor $\mathcal{X}$, common techniques for tensor decomposition are considered by recent works including Candecom/Parafac (CP) decomposition, Tucker decomposition and block term decomposition. CP (Lacroix et al., 2018) decomposes the KG into a sum of component rank-one tensors, which is formulated as $\mathcal{X} \approx \sum_{k=1}^{K} \mathbf{s}^{(k)} \circ \mathbf{r}^{(k)} \circ \mathbf{o}^{(k)}$, where $\circ$ is the outer product operation, $\mathbf{s}^{(k)} \in \mathbb{R}^{d_s}, \mathbf{r}^{(k)} \in \mathbb{R}^{d_r}$ and $\mathbf{o}^{(k)} \in \mathbb{R}^{d_o}$ are vectors composed of the kth dimension elements of the embedding of each subject, relation and object, and d_s, d_r and d_o denote the dimensions of the subjects, relations and objects, respectively. TuckER (Balazevic et al., 2019) decomposes the KG into a core tensor and three component matrices, which is formulated as $\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{S} \times_2 \mathbf{R} \times_3 \mathbf{O}$, where $\mathcal{G} \in \mathbb{R}^{d_s \times d_r \times d_o}$ are core tensors, $\mathbf{S} \in \mathbb{R}^{n_e \times d_s}$, $\mathbf{R} \in \mathbb{R}^{n_r \times d_r}$ and $\mathbf{O} \in \mathbb{R}^{n_e \times d_o}$ are factor matrices in which each row is the embedding of the corresponding entity or relation. The operation $\times_n$ is the n-mode product between the tensor and the matrix. BTD as the generalization of CP decomposition and Tucker decomposition is also employed by Luo et al. (2020). It has been proved that the above bilinear models are the special cases of Tucker decomposition model (Balazevic et al., 2019).

GNN-based models

Some existing works introduce graph neural networks to KGs. ConvE (Dettmers et al., 2018) introduces convolutional layers to model complex interactions between entities and relations. SACN (Shang et al., 2019) utilizes graph connectivity structure by combining weighted graph convolutional network and ConvE. CompGCN (Vashishth et al., 2020) is a graph convolutional based framework that leverages the entity-relation composition operations from KGC models including TransE and DistMult to update entity embeddings. KE-GCN (Yu et al., 2021) updates both entity and relation embeddings by graph convolution operation leveraging various knowledge embedding techniques.

2.2. TKG embedding methods

Most of the existing TKG embedding methods are extensions based on the previous SKG embedding researches, the idea of which is to learn time-specific representations for entities and relations. HyTE (Dasgupta et al., 2018) projects embeddings of entities and relations onto the time-specific hyperplane. TA-TransE (García-Durán et al., 2018) treats relations and timestamps as the predicate sequence which then feeds into LSTM. TTransE (Leblay & Chekol, 2018) explores a variety of extensions of TransE. In DE-TransE (Goel et al., 2020), the timestamp is used to modulate a part of the embeddings of entities and relations. ATiSE (Xu et al., 2019) follows the idea of KG2E and introduces additive time series decomposition. TNTComplEx (Lacroix et al., 2020) extends ComplEx, establishing time as the fourth dimension. TeRo (Xu et al., 2020) regards time as a transformation in complex space. In addition, there are related works that consider the TKG completion task from other different perspectives. CyGNet (Zhu et al., 2021) combines two modes of inference to make predictions based on either the historical vocabulary or the whole entity vocabulary through a copy mechanism. Ding et al. (2021) extends the idea of continuum-depth models to TKG, capturing temporal information through the neural ordinary differential equation and structural information through a GNN.

The above TKG embedding methods are generally designed for discrete facts, and develop different strategies to extend to the continuous case. The several mainstream approaches are as follows. (1) The time axis is divided according to the distribution of facts so that the events are evenly distributed on the divided time period. Then the original time annotation is displaced by the divided time period and treated as a time point. This strategy which HyTE and TeRo adopt, cannot carefully account for the properties of time periods. (2) A continuous fact is discretized into a series of discrete facts in TTransE, obviously leading to a multiplied increase in computational cost. (3) Only the beginning time and end time of the time period are utilized in TA-TransE and TNT-ComplEx, losing the intermediate time information. (4) Some methods like DE-TransE only conducted experiments on discrete fact datasets regardless of continuous facts. In summary, it is a challenge for the existing works to process the discrete facts and continuous facts uniformly and reasonably.

3. Model

A TKG $\mathcal{G}$ is composed of a series of snapshots $\{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(T)}\}$, where $\mathcal{G}^{(t_p)}$ contains all valid facts $\{(s, r, o, t_p)\}$ at time t_p, $s \in \mathcal{E}, r \in \mathcal{R}, o \in \mathcal{E}, t_p \in \mathcal{T}$, with $\mathcal{E}$ denoting the set of entities, $\mathcal{R}$ the set of relations and $\mathcal{T}$ the set of time points. The latent representations learned for each subject, relation and object are denoted as their bold letters $\mathbf{s}$, $\mathbf{r}$ and $\mathbf{o}$ respectively. Our goal is to maximize the score $f(s, r, o, t)$ for all observed facts.

Fig. 3 illustrates the overall framework where BTD with two component tensors is introduced to establish different properties (temporal and static properties) of entities and relations. Further, in the temporal component tensor, the component matrices of entities ($\mathbf{s}_t$ and $\mathbf{o}_t$) focus on the fine-grained time information, while the core tensor ($\mathcal{G}_t$) collects the coarse-grained time information. In this process, the calculation techniques of discrete facts ($t = t_p$) and continuous facts ($t = [t_b, t_e]$) are unified.

3.1. Block term decomposition

Considering the observation that the interaction of certain entities and relations in TKGs does not change over time, the properties of entities and relations are split into two parts, one of which is affected by time while the other is static and remains the same at all times. Representations corresponding to static and temporal properties are trained for entities and relations, which further are used to obtain the plausibility scores of their interactions. The scoring function of a temporal fact (s, r, o, t) consisting of two portions is explored as:

$$f(s, r, o, t) = f_{temp}(s, r, o, t) + f_u(s, r, o), \tag{1}$$

where f_{temp} encodes the interactions among dynamic temporal properties, while f_u encodes the interactions among static properties unchanged over time. Note that $t = t_p$ for discrete facts and $t = [t_b, t_e]$ for continuous facts.

Recall that a KG can be regarded as a third-order binary tensor $\mathcal{X} \in \mathbb{R}^{n_e \times n_r \times n_e}$, in which the element 0 or 1 indicates the fact corresponding to the index is invalid or valid. n_e and n_r denote the number of entities and relations. Several tensor decomposition techniques have been researched by recent works to practice in KG embeddings, including CP (Lacroix et al., 2018) and TuckER (Balazevic et al., 2019).

In this work, another tensor decomposition technique BTD, as the generalization of both CP decomposition and Tucker decomposition, is considered to equip our model with a more powerful expression and generalization capability. BTD admits the modeling of more complex signal components than CP decomposition, and acquires the uniqueness under more restrictive but still fairly natural conditions than TKD (Ma et al., 2019; Ye et al., 2018). From the scene of TKG, BTD introduces multiple core tensors to capture the different aspects of correlations among the three dimensions (i.e., the subject, relation and object). Therefore, BTD fits the scene of the TKG very well, where entities and relations have rich semantic information, including the static property and temporal property of the entities. BTD with two component tensors is adopted to decompose $\mathcal{X}$:

$$\mathcal{X} \approx \mathcal{G}_1 \times_1 \mathbf{S}_1 \times_2 \mathbf{R}_1 \times_3 \mathbf{O}_1 + \mathcal{G}_2 \times_1 \mathbf{S}_2 \times_2 \mathbf{R}_2 \times_3 \mathbf{O}_2, \tag{2}$$

where $\mathcal{G}_1 \in \mathbb{R}^{d_s \times d_r \times d_o}$ and $\mathcal{G}_2 \in \mathbb{R}^{d_s \times d_r \times d_o}$ are core tensors, $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{n_e \times d_s}$, $\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{n_r \times d_r}$ and $\mathbf{O}_1, \mathbf{O}_2 \in \mathbb{R}^{n_e \times d_o}$ are factor matrices, and d_s, d_r and d_o denote the dimensions of the subjects, relations and objects, respectively. The operation $\times_n$ is the n-mode product between

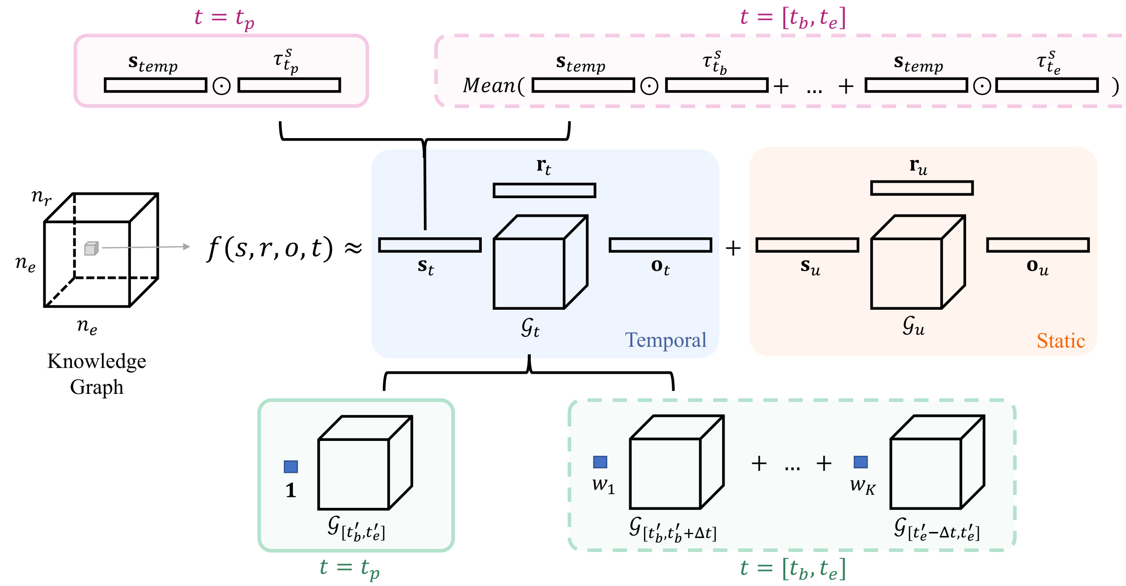


Fig. 3. Illustration of BTDG. Based on the block term decomposition, our model establishes temporal and static properties of entities and relations, focusing on the fine-grained and coarse-grained time information for both discrete and continuous facts uniformly. BTD with two component tensors is applied to model the score of a fact in the TKG. The blue and the orange block represent the temporal and the static portion respectively (corresponding to Eq. (3)). In the first component tensor, the capture of fine-grained time information for the subject corresponds to the pink boxes (corresponding to Eq. (4) and (5), and the relevant part for the object corresponding to Eqs. (6) and (7) is omitted in the figure), while the capture of coarse-grained time information corresponds to the green boxes (corresponding to Eq. (8)). The solid-line boxes and the dashed-line boxes illustrate the case of discrete facts and continuous facts.

the tensor and the matrix. $\mathbf{S}_1$ and $\mathbf{S}_2$ are the two embedding matrices of subjects, where each row of each matrix is the embedding of the corresponding subject. Similarly, $\mathbf{R}_1$ and $\mathbf{R}_2$ as well as $\mathbf{O}_1$ and $\mathbf{O}_2$ are the embedding matrices of relations and objects. After the embeddings of these entities and relations are calculated with the core tensor $\mathcal{G}_1$ and $\mathcal{G}_2$, the credibility score of the corresponding fact can be obtained.

Naturally, two terms in BTD are used to model the interactions among different types of properties, which attributes to a more comprehensive evaluation of facts. For a temporal fact (s,r,o,t), the scoring function in Eq. (1) is reformulated as:

$$f(s,r,o,t) = \mathcal{G}_t \times_1 \mathbf{s}_t \times_2 \mathbf{r}_t \times_3 \mathbf{o}_t + \mathcal{G}_u \times_1 \mathbf{s}_u \times_2 \mathbf{r}_u \times_3 \mathbf{o}_u, \tag{3}$$

where $\mathbf{s}_t \in \mathbb{R}^{d_s}, \mathbf{r}_t \in \mathbb{R}^{d_r}, \mathbf{o}_t \in \mathbb{R}^{d_o}$ denote the temporal representations, with $\mathcal{G}_t \in \mathbb{R}^{d_s \times d_r \times d_o}$ capturing the temporal property interactions, and $\mathbf{s}_u \in \mathbb{R}^{d_s}, \mathbf{r}_u \in \mathbb{R}^{d_r}, \mathbf{o}_u \in \mathbb{R}^{d_o}$ denote the unchanged static representations, with $\mathcal{G}_u \in \mathbb{R}^{d_s \times d_r \times d_o}$ capturing their interactions. The form of Eq. (3) is inspired by Eq. (2). If only one fact (containing one subject, one relation, and one object) is considered in Eq. (2), Eq. (3) can be obtained.

3.2. Fine-grained and coarse-grained

This section describes how f_t collects information presented by facts under two distinct time granularities.

Fine-grained

Note that $\mathbf{s}_t, \mathbf{o}_t$ are latent representations of s and o corresponding to time t to study fine-grained time information. Taking $\mathbf{s}_t$ with $t = [t_b, t_e]$ as an example, representations of s at each time point included in $[t_b, t_e]$ are taken into consideration, namely $\{\mathbf{s}_{t_b}, \mathbf{s}_{t_b+1}, \ldots, \mathbf{s}_{t_e}\}$, which are defined as:

$$\mathbf{s}_{t_p} = \mathbf{s}_{temp} \odot \tau^s_{t_p}, \tag{4}$$

for $t_p \in [t_b, t_e]$, where $\tau^s_{t_p} \in \mathbb{R}^{d_s}$ is the representation of the time point t_p for subjects, which learns the time characteristics and establishes a connection between subjects since it is shared by $\mathbf{s}_{t_p}$ for all s. $\mathbf{s}_{temp}$ denotes the temporal part of the embedding of s acting as its temporal properties. The operator $\odot$ is Hadamard product. Then $\mathbf{s}_t$ is calculated as their average:

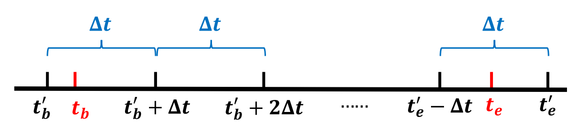


Fig. 4. The meanings of notations on the timeline.

$$\mathbf{s}_t = Mean(\mathbf{s}_{t_p}), t_p \in [t_b, t_e]. \tag{5}$$

Obviously, for the discrete case of $t = t_p$, $\mathbf{s}_t$ can be directly obtained by Eq. (4), while Eq. (5) is omitted since there exists only one time point in t. Therefore, the calculation of discrete facts and continuous facts is essentially the same.

With the notations of similar meanings, the object representation $\mathbf{o}_t$ with $t = [t_b, t_e]$ or $t = t_p$ is obtained as follows:

$$\mathbf{o}_{t_p} = \mathbf{o}_{temp} \odot \tau^o_{t_p}, \tag{6}$$

$$\mathbf{o}_t = Mean(\mathbf{o}_{t_p}), t_p \in [t_b, t_e]. \tag{7}$$

By focusing on each included time point in the time period, facts are evaluated on a fine-grained time scale, and in this process, discrete and continuous facts are processed uniformly.

Coarse-grained

The core tensor $\mathcal{G}_t$ models the interaction of entities and relations in time t and studies the coarse-grained time information. A rule is firstly explored to construct the coarse-grained time periods. For a TKG, a granularity Δt is selected so that multiple fine-grained time points are included in a coarse-grained time period corresponding to a specific $\mathcal{G}$.

Taking $\mathcal{G}_t$ with $t = [t_b, t_e]$ as an example, the time period $[t_b, t_e]$ spans one or several coarse-grained time periods. Considering a general case of spanning K time periods, the corresponding core tensor is defined as:

$$\mathcal{G}_t = \mathcal{G}_{[t_b,t_e]} = \sum_{k=1}^{K} w_k \, \mathcal{G}_{[t_1(k),t_2(k)]}, \; K = (t'_e - t'_b)/\Delta t \tag{8}$$

with $t_1(k) = t'_b + (k-1)\Delta t$ and $t_2(k) = t'_b + k\Delta t$. t'_b and t'_e are the time points closest to t_b and t_e under the division of Δt and satisfy $[t_b, t_e] \subseteq [t'_b, t'_e]$. $w_k = \frac{t_2(k)-t_1(k)}{t_e-t_b}$ is a weight coefficient equal to the proportion of the time period represented by the corresponding $\mathcal{G}$ in $[t_b, t_e]$, thus $\sum_{k=1}^{K} w_k = 1$. Fig. 4 describes the meanings of these notations on the time axis. For example, if the selected granularity is 10 years and the valid time period of the fact is set to $[2006, 2021]$, the core tensor $\mathcal{G}$ is calculated as $\mathcal{G}_{[2006,2021]} = \frac{4}{16}\mathcal{G}_{[2006,2009]} + \frac{10}{16}\mathcal{G}_{[2010,2019]} + \frac{2}{16}\mathcal{G}_{[2020,2021]}$. The idea of providing independent embedding for each coarse-grained time slice is similar to the position encoding technique in Transformer (Vaswani et al., 2017). But unlike Transformer, which encodes position information of tokens in the sequence, our model is not equipped with this feature, and we plan to address this in the future.

As for the case of $t = t_p$, the discrete fact can be regarded as a special form of the continuous fact with $t_b = t_e = t_p$. Hence, Eq. (8) can be directly adopted for the calculation of $\mathcal{G}_{t_p}$ with $K = 1$, but applying a different definition of $w = 1$. The core tensor $\mathcal{G}_{t_p}$ for discrete facts provides the context of time period containing t_p of interest, acting as background information.

In summary, $\mathbf{s}_{[t_b,t_e]}$ and $\mathbf{o}_{[t_b,t_e]}$ capture the properties of entities and relations at each intermediate time point from t_b to t_e, while $\mathcal{G}_{[t_b,t_e]}$ captures the complex interaction on the coarse-grained time scale. Therefore, the model can capture two time-granularity information and evaluate the plausibility of facts from distinct scales in a unified manner for both discrete and continuous cases.

3.3. Time smoothness

In addition, we assume that the evolution of real-world KGs is generally smooth without sudden changes, and the adjacent time points exhibit similar properties. Therefore, a time smoothness term is added to the final loss function for the penalty of mutation:

$$\mathcal{L}_{smooth} = Mean((\tau_i^s - \tau_{i-1}^s) + (\tau_i^o - \tau_{i-1}^o)),\ i \in [t_B + 1, t_E], \tag{9}$$

where t_B and t_E refer to the earliest time and the latest time in the whole TKG.

3.4. Training

The predicted probability of a fact is defined as:

$$p(s, r, o, t) = \sigma(f(s, r, o, t)), \tag{10}$$

where $\sigma(\cdot)$ represents the sigmoid function. The binary cross entropy for 1-N scoring (Dettmers et al., 2018) is employed to train BTDG, and the final loss function is:

$$\mathcal{L} = -\frac{1}{n_q}\sum_i (y^i \log(p^i) + (1-y^i)\log(1-p^i)) + \alpha\mathcal{L}_{smooth}, \tag{11}$$

where p^i represents the predicted probability of the ith fact, and y^i represents its true label. α is a hyperparameter weighing the importance of time smoothness.

4. Experiments

4.1. Experimental setup

Datasets

Two discrete fact datasets and a continuous dataset are used to verify the performance of BTDG.

- ICEWS14 and ICEWS05-15 are two discrete fact datasets created by García-Durán et al. (2018). They are both the subset of ICEWS (Lautenschlager et al., 2015), containing the political events with timestamps in 2014 and 2005–2015 respectively.
- Wikidata12k (Dasgupta et al., 2018) is a continuous fact dataset. The start time and end time of the time period in facts are in the form of (year-month-day). Since most of the records of months and dates are missing, we only utilize the years as total time annotations ignoring months and dates.

Table 1
Statistics of datasets.

Dataset	#Entities	#Relations	#Time steps	Time span
ICEWS14	7,128	230	365	2014.01.01–2014.12.31
ICEWS05–15	10,488	251	4,017	2005.01.01–2015.12.31
Wikidata12k	12,255	24	–	19–2020

The statistics of the datasets are summarized in Table 1.

Baselines

The performance of BTDG is compared against both SKG and TKG embedding methods.

(a) SKG embedding methods

SKG embedding methods ignore the time information and treat the TKG as a collection of snapshots at each time point, i.e., $\mathcal{G} = \cup_t \mathcal{G}^{(t)}$. We select typical representative methods mentioned before, including: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016) and RotatE (Sun et al., 2019).

TransE is a classic translational distance model with scoring function $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$. DistMult is a semantic matching model with scoring function $\|\mathbf{h}\cdot\mathbf{r}\cdot\mathbf{t}\|$. ComplEx models entities and relations in the complex space, and RotatE models the relations as rotations in the complex plane.

(b) TKG embedding methods

HyTE (Dasgupta et al., 2018), TTransE (Leblay & Chekol, 2018), TA-TransE, TA-DistMult (García-Durán et al., 2018), DE-SimplE (Goel et al., 2020), ATiSE (Xu et al., 2019) and TeRo (Xu et al., 2020) are selected as TKG embedding baselines.

HyTE projects embeddings of entities and relations onto the time-specific hyperplane. TTransE explores a variety of extensions of TransE. TA-TransE and TA-DistMult treat relations and timestamps as the predicate sequence which then feeds into LSTM. In DE-TransE, the timestamp is used to modulate a part of the embeddings of entities and relations. ATiSE follows the idea of KG2E and introduces additive time series decomposition. TeRo regards time as a transformation in complex space.

Since DE-SimplE is designed for the discrete facts and does not provide an approach to extend to continuous ones, we only use it as the baseline on the discrete fact dataset.

Evaluation metrics

For each fact (s, r, o, t) in the test set, we make predictions from both sides for each test fact, i.e., complete $(s, r, ?, t)$ and $(?, r, o, t)$. Refer to the test phase in Bordes et al. (2013), for each query $(s, r, ?, t)$ ($(?, r, o, t)$ similarly), we first calculate the scores of each candidate fact in $\{(s, r, o_c, t)|o_c \in \mathcal{E}\}$, and then report the rank of the score of the correct one (s, r, o, t) among all scores. In order to ensure all candidate facts except the test fact are corrupted negative samples, it is necessary to remove the facts in the train set, validation set and test set (except for the test fact of interest) in advance which actually acts as the correct ones from the candidate fact set. The result after such processing is called "filtered".

The evaluation metrics are Hits@n defined as the proportion of the test facts whose rank is in the top n, and mean reciprocal rank (MRR) defined as the average of the reciprocal ranks of all test facts. We report Hits@1, Hits@3, Hits@10 and MRR in filtered settings for each dataset.

Table 2
Experimental results of comparative study.

	ICEWS14				ICEWS05-15				Wikidata12k			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
TransE	.094	–	.637	.280	.090	–	.663	.294	.100	.192	.339	.178
DistMult	.323	–	.672	.439	.337	–	.691	.456	.119	.238	.460	.222
ComplEx	.347	.527	.716	.467	.362	.535	.729	.481	.123	.253	.436	.233
RotatE	.291	.478	.690	.418	.164	.355	.595	.304	.116	.236	.461	.221
TTransE	.074	–	.601	.255	.084	–	.616	.271	.096	.184	.329	.172
HyTE	.108	.416	.655	.297	.116	.445	.681	.316	.098	.197	.333	.180
TA-TransE	.095	–	.625	.275	.096	–	.668	.299	.030	.267	.429	.178
TA-DistMult	.363	–	.686	.477	.346	–	.728	.474	.122	.232	.447	.218
DE-SimplE	.418	.592	.725	.526	.392	.578	.748	.513	–	–	–	–
ATiSE	.436	<u>.629</u>	<u>.750</u>	.550	.378	.606	.794	.519	.175	.317	.481	.280
TeRo	<u>.468</u>	.621	.732	<u>.562</u>	<u>.469</u>	<u>.668</u>	<u>.795</u>	<u>.586</u>	<u>.198</u>	<u>.320</u>	<u>.507</u>	<u>.299</u>
BTDG	**.516** (4.8%)	**.656** (2.7%)	**.753** (0.3%)	**.601** (3.9%)	**.534** (6.5%)	**.687** (1.9%)	**.798** (0.3%)	**.627** (4.1%)	**.214** (1.6%)	**.351** (3.1%)	**.523** (1.6%)	**.314** (1.5%)

Table 3
Experimental results of ablation study.

	ICEWS14				Wikidata12k			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
BTDG	**.516**	**.656**	**.753**	**.601**	**.214**	**.351**	**.523**	**.314**
BTDG w/o temporal	.361	.533	.698	.474	.160	.291	.498	.264
BTDG w/o static	.482	.605	.681	.554	.185	.282	.426	.264
BTDG w/o fined	.434	.610	.745	.542	.207	.349	.508	.307
BTDG w/o coarse	.509	.648	.746	.593	.197	.331	.520	.299
BTDG w/o smoothness	.502	.614	.709	.573	.212	.333	.477	.301

Implementation

All the experiments are implemented by PyTorch and trained with Adam optimizer. The codes of our paper are available online: https://github.com/JaneYul/BTDG.

For ICEWS14 and ICEWS05-15, the coarse time granularity Δt is selected from {year, quarter, month}, the learning rate lr from {0.0005, 0.001, 0.005, 0.001}, the dimension of entity embedding d_e from {30, 50, 100, 150, 200} (let $d_s = d_o = d_e$), the dimension of relation embedding d_r from {20, 30, 50, 100}, and the time smoothing weight α from {1, 0.1, 0.01, 0.001}. For Wikidata12k, the coarse time granularity Δt is selected from {century, decade, 5 years}, and the set of candidate values for lr, d_e, d_r, and α are the same as that in ICEWS.

In our implementation, the parameters used in ICEWS14 are: Δt = month, $lr = 0.001$, $d_e = 200$, $d_r = 50$, $\alpha = 0.01$. The parameters used in ICEWS05-15 are: Δt = quarter, $lr = 0.001$, $d_e = 100$, $d_r = 50$, $\alpha = 0.01$. The parameters used in Wikidata12k are: Δt = 5 years, $lr = 0.001$, $d_e = 30$, $d_r = 20$, $\alpha = 0.1$. We observe the best parameter d_e and d_r in Wikidata12k are smaller than that in ICEWS, which may be due to the less relation in Wikidata12k. The values of more detailed parameters are listed in the code link.

4.2. Comparative study

The experimental results of BTDG and baselines are shown in Table 2, where the best results are marked in bold and the second best results are underlined. All results of baselines are taken from Xu et al. (2020). BTDG outperforms all the competitors on the three datasets and we report the increases of BTDG results relative to the suboptimal ones in parentheses.

Performance and analysis on the discrete fact datasets

On both ICEWS14 and ICEWS05-15, the performance improvement of BTDG relative to the second best baseline on Hits@1 and Hits@3 is greater than that on Hits@10, which means the development of BTDG is mainly reflected in the capacity to improve the ranking of the correct answer to the top three and the first. All these baselines lack in extracting information from different time scales, and the deficiency of their performance indicates the effectivity of the coarse and fine time scales BTDG explores.

Performance and analysis on the continuous fact dataset

On the continuous fact dataset, BTDG also outperforms all the baselines, which can be attributed to the more detailed modeling of the continuous fact with its time period. Comparing the experimental results of ATiSE and TeRo on different types of datasets, it is observed that the performance of ATiSE on the continuous fact dataset is worse than TeRo, while Hits@3 and Hits@10 reported on the discrete fact dataset ICEWS14 by ATiSE achieve better results on the contrary, which indicates these two methods lack a unified focus on different types of data. In contrast, BTDG provides a unified processing architecture for two types of temporal facts, and obtains the best performance on both types of datasets.

4.3. Ablation study

In order to verify the effectiveness of the various components in BTDG, we conduct a series of ablation experiments on two different types of datasets, including ICEWS14 and Wikidata12k, to explore how each part in BTDG works for different types of facts.

Specifically, we report the experimental results after the following five components of the model are removed respectively.

(a) Temporal information

Any parameters related to time are removed, and the first term in BTD is replaced with the same form as the second term, but with different parameters:

$$f(s,r,o,t) = \mathcal{G}_u^{(1)} \times_1 \mathbf{s}_u^{(1)} \times_2 \mathbf{r}_u^{(1)} \times_3 \mathbf{o}_u^{(1)} + \mathcal{G}_u^{(2)} \times_1 \mathbf{s}_u^{(2)} \times_2 \mathbf{r}_u^{(2)} \times_3 \mathbf{o}_u^{(2)},$$

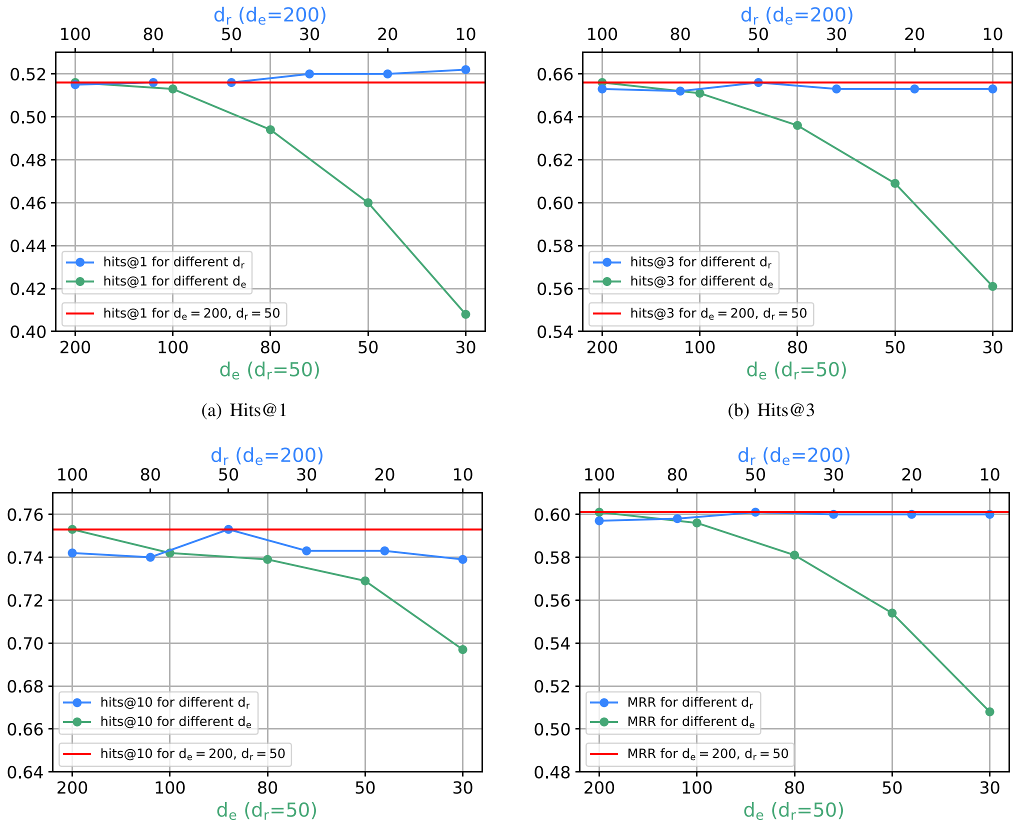
This model is denoted as “BTDG w/o temporal”.

(b) Static properties

Exclude the static properties, i.e., remove the first term $f_u(s,r,o)$ in the scoring function and only reserve the second term $f_{temp}(s,r,o,t)$. This model is denoted as “BTDG w/o static”.

(c) Fine time granularity

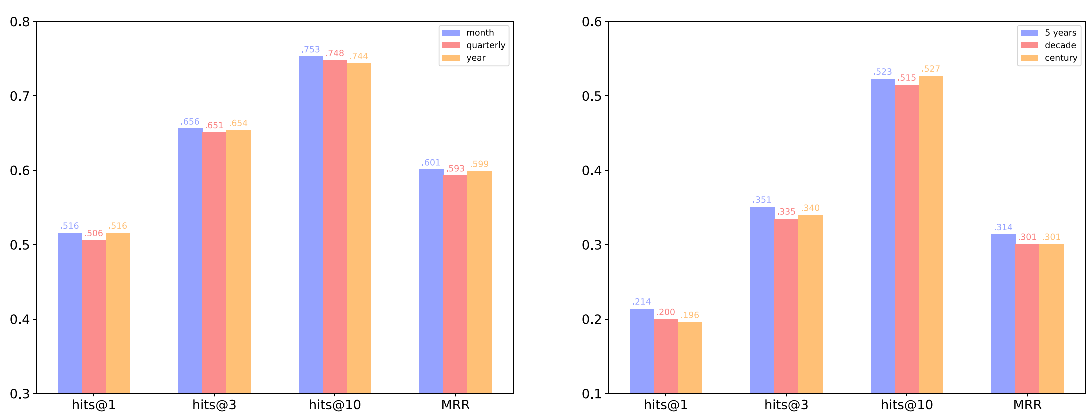
Remove the design of fine time granularity. This model is denoted as “BTDG w/o fined”.



(a) Hits@1 (b) Hits@3

(c) Hits@10 (d) MRR

Fig. 5. The impact of distinct values of d_e and d_r on ICEWS14.



(a) On ICEWS14 (b) On Wikidata12k

Fig. 6. The impact of distinct coarse time granularity.

(d) Coarse time granularity

Remove the design of coarse time granularity. This model is denoted as "BTDG w/o coarse".

(e) Time smoothness

The time smoothing penalty term is removed in the loss function, i.e., the coefficient α is set to 0. This model is denoted as "BTDG w/o smoothness".

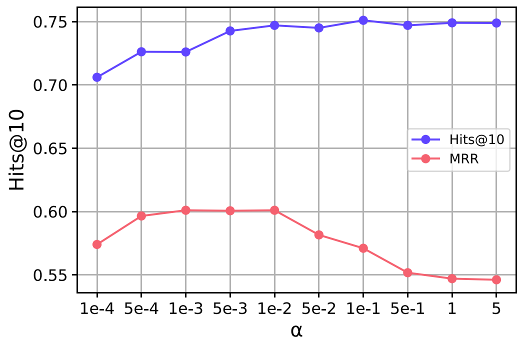
All experimental results are shown in Table 3. When some components of the model are removed, the performance decreases to varying degrees, which implies they all work. By analyzing the difference in the decline of model performance, the specific role played by each component of the model can be inferred. In detail, we obtain the following conclusions.

Time information and static attributes

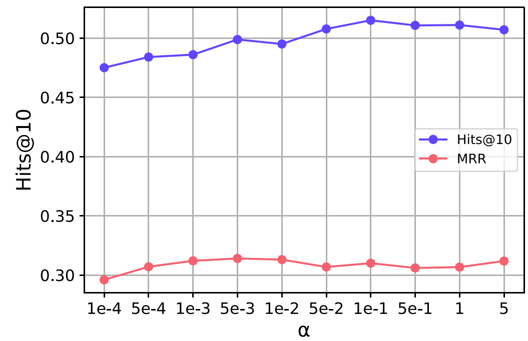
On ICEWS14, the experimental result of "BTDG w/o temporal" is worse than that of "BTDG w/o static", indicating that the time information in BTDG plays a more significant role, which is consistent with our intuition. On Wikidata12k, the MRRs of the two models are similar. We speculated that it may be due to the long-tail characteristic of Wikidata12k which makes it more difficult for the model to collect time information. The time points in ICEWS14 are evenly distributed, while the facts in Wikidata12k are mainly concentrated after 1900.

Fine time granularity and coarse time granularity

On ICEWS14, Hits@1 reported by "BTDG w/o fined" is significantly higher than that of "BTDG w/o coarse". However, the superiority of the



(a) On ICEWS14



(b) On Wikidata12k

Fig. 7. The impact of distinct values of α.

Table 4
Statistics of time overhead. (IC14, IC15 and Wiki are shorthand for ICEWS14, ICEWS05-15 and Wikidata12k).

	BTDG			TeRo			ATiSE		
	IC14	IC15	Wiki	IC14	IC15	Wiki	IC14	IC15	Wiki
Training time per epoch (s)	93	489	66	4.34	22.12	3.53	6.98	35.68	28.04
Num of epochs	150	100	150	5000	4000	900	5000	2250	400
Total training time (s)	13 950	48 900	9900	21 700	88 480	3177	34 750	80 280	11 216

"BTDG w/o fined" on Hits@3 is not as obvious as that on Hits@1, and Hits@10s reported by the two methods are similar. It can be inferred that the fine-grained time information is more critical than coarse-grained time information on ICEWS14, because in discrete fact datasets, the form of time representation is originally a fine-grained time point with coarse-grained time as auxiliary information. On Wikidata12k, the experiment shows the opposite results. For the time period in the continuous facts, it is more effective to capture time information through a coarse time scale.

Time smoothness

For the two datasets, after removing the time smoothing term, the performance degradation of the model on Hits@10 is more obvious, while the impact on Hits@1 is less. It can be seen that the time smoothing term promotes the ranking of the correct entity into the top 10, and the assumption that the representations of adjacent time points are similar to each other in the latent space is reasonable.

4.4. Parameter study

Dimension of entity and relation

In the comparative study on ICEWS14, the values of d_e and d_r we selected for BTDG are 200 and 50. Now we examine the effect of distinct values of d_e and d_r on the performance of the model when the other parameter are fixed respectively.

Fig. 5 illustrates the experimental results (including four evaluation metrics) obtained when d_e is fixed at 200 and d_r is set to various values (corresponding to the blue line), and when d_r is fixed at 50 and d_e is set to various values (corresponding to the green line). The red line in the figure is the result when $d_e = 200$ and $d_r = 50$, which is adopted in the comparative study.

It can be observed from the experimental results that different values of d_e have a large impact on the performance of our model. As d_e decreases, the four indicators all have different degrees of obvious decline. On the other hand, when d_e is fixed and d_r changes, the reported values of the four indicators have little change (except for hits@10, which fluctuates relatively large). The performance of the model remains at a relatively stable level even when d_r takes a small value (such as 10 and 20).

Based on the above observations, it can be concluded that larger dimensions are required to store information of entities due to their rich semantics, while the semantics of the relations are relatively clear and single, and the relations need to model the interaction between numerous various entities, so satisfying results can be obtained with small dimensions.

Coarse time granularity

We also investigated the influence of the selection of different coarse time granularities on the model's performance. Fig. 6 shows the experimental results of ICEWS14 and Wikidata12k. We find that too coarse granularity is not conducive to model performance.

Time smoothness coefficient

In addition, we explore the impact of the hyperparameter α in the time smoothness term on the performance of BTDG. Fig. 7 exhibits Hits@10s and MRRs reported by ICEWS14 and Wikidata12k under different values of α. The proposed method reaches a satisfactory and stable performance when α is between 0.005 and 0.01 on ICEWS14, or between 0.05 to 1 on Wikidata12k.

4.5. Efficiency study

Time overhead

We compared the time overhead of our method and the two competitive baselines, TeRo (Xu et al., 2020) and ATiSE (Xu et al., 2019) on the three datasets. The metrics include the training time per epoch, the number of epochs required for the model to converge, and the total training time. The results are listed in Table 4.

Experimental results indicate that our method converges faster, although the training time for each epoch is longer. When the epoch is small, the results on the validation sets are stable and the results on the test sets are better than the two baselines. On ICEWS14 and ICEWS05-15, the total training time of BTDG is obviously shorter than that of TeRo and ATISE.

Memory costing

Similarly, we compared the space complexity as well as the memory costing of our method and TeRo and ATiSE on the three datasets. The results are listed in Table 5.

Our method requires more memory on ICEWS14 and ICEWS05-15, and requires less on Wikidata12k. We need to explain although the space complexity of our method is $O(n_{\Delta t} * d_e * d_r * d_e)$, the parameters

Table 5
Statistics of memory costing. (IC14, IC15 and Wiki are shorthand for ICEWS14, ICEWS05-15 and Wikidata12k).

	BTDG			TeRo			ATiSE		
	IC14	IC15	Wiki	IC14	IC15	Wiki	IC14	IC15	Wiki
Space complexity	$O(n_{\Delta t} * d_e * d_r * d_e)$			$O(n_e * d + n_r * d + n_r * d)$			$O(n_e * d + n_r * d)$		
$d_e(= d_{time})$	200	100	30	500	500	500	500	500	500
d_r	50	50	20	500	500	500	500	500	500
Memory cost (MB)	2217	2115	1621	1451	1695	1719	1773	1987	2067

Table 6
Experiments of BTDG with different parameters.

	d_e	d_r	Hits@1	Hits@3	Hits@10	MRR	Memory cost (M)	Training time (s)
BTDG	200	50	0.516	0.656	0.753	0.601	–	–
	100	50	0.506	0.643	0.741	0.590	–	–
	80	50	0.494	0.636	0.739	0.581	–	–
	50	20	0.457	0.609	0.735	0.554	1439	10 230
TeRo	500	500	0.468	0.621	0.732	0.562	1451	21 700
ATiSE	500	500	0.436	0.629	0.75	0.55	1773	34 750

$d_e(= d_{time})$ and d_r on the three datasets are set to 200/100/30 and 50/50/20, while TeRo and ATiSE both set these parameters to 500. Note that the number of entities of three datasets is all around 10000. Therefore, our method uses fewer parameters for entity and relation embeddings, and has additional core tensor parameters.

Experiments with equal parameters

To further explore the performance of our model, we compare BTDG to the other models (ATiSE and TeRo) on ICEWS14 while the number of parameters is kept to be equal. We conduct experiments by phasing down d_e and d_r, and report the experimental results as well as the memory costing and the training time in Table 6.

From the experimental results, we find that in the same level of parameter size, BTDG still obtains a competitive result in less training time.

5. Conclusion

In this paper, we innovatively propose to model the facts from two different time scales and to overcome the shortcomings of the existing TKG embedding methods in dealing with different types of facts. We propose in this paper that the block term decomposition is adopted to design the scores of facts, which has the following advantages. First, modeling the facts from two different time scales overcomes the shortcomings of the existing TKG embedding methods in dealing with different types of facts. Second, the proposed method also provides a unified framework for discrete facts and continuous facts. Last, the proposed model can converge quickly and achieve satisfactory results without setting too large entity and relation dimensions.

However, there still exist some deficiencies in the proposed model. The model treats each coarse-grained time slice in the time period equally for simplicity's sake, while their contributions to the fact inference may vary. This problem might be solved by reweighting the time slice, and how to obtain the appropriate weight needs to be carefully designed. In future work, the concept of distinct time granularities can also be extended to existing TKG methods based on other semantic matching models or translational distance models.

CRediT authorship contribution statement

Yujing Lai: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Data curation, Visualization. **Chuan Chen:** Validation, Writing – review & editing, Supervision. **Zibin Zheng:** Project administration. **Yangqing Zhang:** Project administration.

Declaration of competing interest

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

Acknowledgments

The research is supported by the Key-Area Research and Development Program of Guangdong Province (2020B010165003), the National Natural Science Foundation of China (62176269, U1911202), the Guangdong Basic and Applied Basic Research Foundation, China (2019A1515011043).

References

Balazevic, I., Allen, C., & Hospedales, T. M. (2019). Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 5184–5193).

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Neural information processing systems* (pp. 1–9).

Dasgupta, S. S., Ray, S. N., & Talukdar, P. (2018). Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2001–2011).

De Lathauwer, L. (2008). Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness. *SIAM Journal on Matrix Analysis and Applications*, *30*(3), 1033–1066.

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1*.

Ding, Z., Han, Z., Ma, Y., & Tresp, V. (2021). Temporal knowledge graph forecasting with neural ODE. arXiv:2101.05151.

García-Durán, A., Dumancic, S., & Niepert, M. (2018). Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4816–4821).

Goel, R., Kazemi, S. M., Brubaker, M., & Poupart, P. (2020). Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 04* (pp. 3988–3995).

He, S., Liu, K., Ji, G., & Zhao, J. (2015). Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 623–632).

Kacupaj, E., Plepi, J., Singh, K., Thakkar, H., Lehmann, J., & Maleshkova, M. (2021). Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main volume* (pp. 850–862).

Kazemi, S. M., & Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018* (pp. 4289–4300).

Lacroix, T., Obozinski, G., & Usunier, N. (2020). Tensor decompositions for temporal knowledge base completion. In *8th International conference on learning representations*.

Lacroix, T., Usunier, N., & Obozinski, G. (2018). Canonical tensor decomposition for knowledge base completion. In *International conference on machine learning* (pp. 2863–2872). PMLR.

Lautenschlager, J., Shellman, S., & Ward, M. (2015). Icews event aggregations. Harvard Dataverse 3.

Leblay, J., & Chekol, M. W. (2018). Deriving validity time in knowledge graph. In *Companion proceedings of the the web conference 2018* (pp. 1771–1776).

Lee, D., Oh, B., Seo, S., & Lee, K. H. (2020). News recommendation with topic-enriched knowledge graphs. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 695–704).

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence, vol. 29, no. 1*.

Luo, T., Wei, Y., Yu, M., Li, X., Zhao, M., Xu, T., Yu, J., Gao, J., & Yu, R. (2020). BTDE: Block term decomposition embedding for link prediction in knowledge graph. In *ECAI 2020* (pp. 817–824). IOS Press.

Ma, X., Zhang, P., Zhang, S., Duan, N., Hou, Y., Zhou, M., & Song, D. (2019). A tensorized transformer for language modeling. In *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019* (pp. 2229–2239).

Molokwu, B. C., Shuvo, S. B., Kar, N. C., & Kobti, Z. (2020). Node classification in complex social graphs via knowledge-graph embeddings and convolutional neural network. In *International conference on computational science* (pp. 183–198). Springer.

Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Icml*.

Shang, C., Tang, Y., Huang, J., Bi, J., He, X., & Zhou, B. (2019). End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 3060–3067).

Sun, Z., Deng, Z., Nie, J., & Tang, J. (2019). RotatE: Knowledge graph embedding by relational rotation in complex space. In *7th International conference on learning representations*. OpenReview.net.

Tran, H. N., & Takasu, A. (2020). Multi-partition embedding interaction with block term format for knowledge graph completion. In *ECAI 2020 - 24th european conference on artificial intelligence, - including 10th conference on prestigious applications of artificial intelligence, vol. 325* (pp. 833–840).

Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning* (pp. 2071–2080). PMLR.

Vashishth, S., Sanyal, S., Nitin, V., & Talukdar, P. P. (2020). Composition-based multi-relational graph convolutional networks. In *8th International conference on learning representations*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017* (pp. 5998–6008).

Wang, Z., Xu, H., Li, X., & Deng, Y. (2021). Learning attention-based translational knowledge graph embedding via nonlinear dynamic mapping. In *Advances in knowledge discovery and data mining - 25th Pacific-Asia conference* (pp. 141–154).

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence, vol. 28, no. 1*.

Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H. S., & Lehmann, J. (2019). Temporal knowledge graph embedding model based on additive time series decomposition. arXiv preprint arXiv:1911.07893.

Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H. S., & Lehmann, J. (2020). Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th international conference on computational linguistics* (pp. 1583–1593).

Yang, B., Yih, W., He, X., Gao, J., & Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In *3rd International conference on learning representations, ICLR 2015, conference track proceedings*.

Ye, J., Wang, L., Li, G., Chen, D., Zhe, S., Chu, X., & Xu, Z. (2018). Learning compact recurrent neural networks with block-term tensor decomposition. In *2018 IEEE conference on computer vision and pattern recognition* (pp. 9378–9387).

Yu, D., Yang, Y., Zhang, R., & Wu, Y. (2021). Knowledge embedding based graph convolutional network. In *WWW '21: The web conference 2021, virtual event / Ljubljana* (pp. 1619–1628).

Zhu, C., Chen, M., Fan, C., Cheng, G., & Zhang, Y. (2021). Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence*.