



Each Snapshot to Each Space: Space Adaptation for Temporal Knowledge Graph Completion

Yancong Li¹, Xiaoming Zhang²(✉), Bo Zhang¹, and Haiying Ren³

¹ School of Cyber Science and Technology, Beihang University, Beijing, China

² State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, People's Republic of China
yolixs@buaa.edu.cn

³ School of Computer Science and Engineering, Beihang University, Beijing, China

Abstract. Temporal knowledge graphs (TKGs) organize and manage the dynamic relations between entities over time. Inferring missing knowledge in TKGs, known as temporal knowledge graph completion (TKGC), has become an important research topic. Previous models handle all facts with different timestamps in an identical latent space, even though the semantic space of the TKG changes over time. Therefore, they are not effective to reflect the temporality of knowledge. To effectively learn the time-aware information of TKGs, different latent spaces are adapted for temporal snapshots at different timestamps, which yields a novel model, i.e., Space Adaptation Network (SANE). Specifically, we extend convolutional neural networks (CNN) to map the facts with different timestamps into different latent spaces, which can effectively reflect the dynamic variation of knowledge. Meanwhile, a time-aware parameter generator is designed to explore the overlap of latent spaces, which endows CNN with specific parameters in term of the context of timestamps. Therefore, knowledge in adjacent time intervals is efficiently shared to boost the performance of TKGC, which can learn the validity of knowledge over a period of time. Extensive experiments demonstrate that SANE achieves state-of-the-art performance on four well-established benchmark datasets for temporal knowledge graph completion.

Keywords: Temporal knowledge graph · Temporal knowledge graph completion · Space adaptation · Parameter generation

1 Introduction

Knowledge Graphs (KGs) [1, 3] organize and manage knowledge as structured information in the form of fact triples, which are crucial in various downstream tasks [14, 33]. In KGs, nodes represent entities, and directed edges indicate relations between entities. Notably, most KGs are inherently incomplete, which motivates research on Knowledge Graph Completion (KGC). KGC aims to infer new facts from existing facts in KGs and is important to KG field. However, the

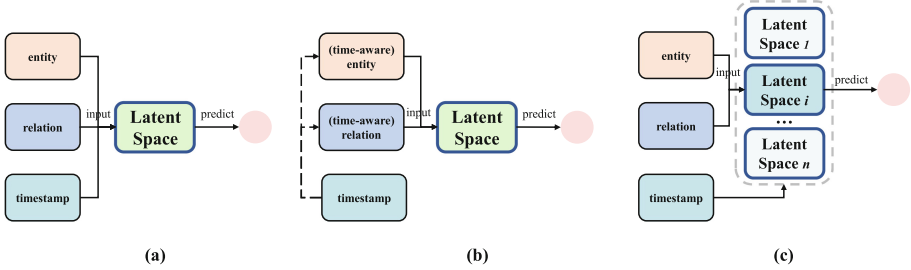


Fig. 1. Existing methods vs. our method. (a) Entities, relations and timestamps are learned to obtain independent representations [21, 23, 32, 42]. (b) Temporal information is implicit in entities or relations to generate time-aware representations of entities or relations [13, 41, 44]. (c) A concise illustration of our model. Entities and relations are adapted into specific latent spaces that are produced based on timestamps.

facts are not always time-invariant, and the validity of triples is often time-aware. Traditional KGC methods are insensitive to temporal information, since they intuitively assume that triples in KGs are universally true. Therefore, these methods are not effective to predict the temporal facts.

Temporal Knowledge Graphs (TKGs), including ICEWS [5], YAGO3 [26], Wikidata [9], etc., are introduced to organize additional temporal aspects of facts. In TKGs, static triples are associated with timestamps, which reflect the temporal dynamics of facts in the form of quadruples. Knowledge in a TKG can be described by the evolution of snapshots over time. However, TKGs also suffer from incompleteness as with static KGs. Therefore, predicting missing knowledge with specific timestamps in TKGs, i.e., Temporal Knowledge Graph Completion (TKGC), has gained growing interest.

Recently, a variety of models have been proposed to handle TKGC. These models significantly outperform traditional KGC models by capturing the latent correlation between knowledge and temporal information. Previous works learn independent representations of entities, relations and timestamps [21, 23, 32, 42] as shown in Fig. 1(a), or obtain time-aware representations by integrating temporal information into entities and relations [13, 41, 44] as shown in Fig. 1(b). These works model variable knowledge in an identical latent space, even though the semantic space of the TKG changes over time. Therefore, these methods are not effective to learn the temporality of knowledge. In practice, TKGs can be decomposed into two components, time-variability and time-stability, which are intrinsic and critical characteristics in TKGs. Time-variability denotes the dynamic knowledge which is varied in different snapshots. For example, the *president of USA* was *George W. Bush* on *2009-01-01*, but became *Barack Obama* on *2010-01-01*. On the other side, time-stability denotes the knowledge which remains unchanged for a period time. For example, (*Barack Obama*, *presidentOf*, *USA*) remains valid for a specific period from *2009-01-20* to *2017-01-20*. One of simple yet generic solution of TKGC is to encode the knowledge in different temporal snapshots into different latent spaces, such that the time-aware information at each snapshots can be captured effectively. Meanwhile, there is a part

of knowledge remains unchanged during a interval. Therefore, knowledge sharing across adjacent snapshots is also required for knowledge accumulation over time. However, it is quite challenging to derive a latent space for each snapshot, since the number of model parameters linearly depend on the number of timestamps. In addition, how to efficiently gather valid knowledge from different spaces is an important problem as well.

In response, we propose a novel model named Space Adaptation Network (SANE) for TKGC as shown in Fig. 1(c). We establish the correlation between latent spaces and snapshots in terms of parameter generation, i.e., a time-specific network is produced for each snapshot, such that the facts with different timestamps are encoded into different spaces. Specifically, to model time-variability, a dynamic convolutional neural network (DCNN) is proposed to deal with the entities and relations with different parameters that are specific to the corresponding timestamps. Therefore, each temporal snapshot, i.e., knowledge graph with the same timestamp, is processed in a specific space. Essentially, TKGC is turn into the static KGC by handling different temporal snapshots in separate spaces. Thereby, this solution alleviates mutual interference of the knowledge with different timestamps. In addition, we explore how to produce the parameters with respect to timestamps to ensure time-stability. Thus, a time-aware parameter generator (TaPG) is designed to constrain the overlap of latent spaces according to the distance of timestamps, which allows adjacent snapshots to share different but similar latent spaces. In this way, valid knowledge across multiple snapshots within a time interval is preserved. The model is experimentally evaluated in detail on several recent standard benchmarks and achieves state-of-the-art performance compared to existing TKGC methods.

To summarize, our contributions are as follows:

- We propose a novel space adaptation network SANE for TKGC, where different latent spaces are adapted for different temporal snapshots. To the best of our knowledge, this is the first work to implement TKG completion from the perspective of space adaptation.
- By constraining the overlap of different spaces in terms of time intervals, the model strikes a balance between learning time-variability and adapting to time-stability.
- Experimental results on four benchmark datasets with rich temporal information demonstrate the superiority of our model¹

2 Related Work

In this section, typical methods for static knowledge graph completion and temporal knowledge graph completion are introduced, and research advances on parameter generation in various fields are briefly reviewed.

Static Knowledge Graph Completion aims to infer missing facts in static KGs. Previous works can be broadly classified into translational, bilinear, and

¹ Our code will be publicly available at <https://github.com/codeofpaper/SANE>.

neural models. TransE [4] is a well-known translation-based model that regards relations as translations from head entities to tail entities. Later, several variants such as TransH [39], TransR [24] and TransD [16] have been proposed to improve the shortcomings of TransE. Bilinear models, such as RESCAL [29], ComplEx [37], and TuckER [2], represent relations as linear transformations acting on entity embeddings, and use bilinear functions to compute plausibility scores for facts. Neural models, such as ConvE [8], InteractE [38], and RGhat [48], complement KGs with nonlinear neural networks and show great effectiveness. The above models have achieved promising results in addressing the incompleteness of KGs. However, they assume that the facts are static and thus cannot model the temporality in TKGs. For example, given two quadruples with timestamps: (*Barack Obama, presidentOf, USA, 2010-01-01*) and (*Barack Obama, presidentOf, USA, 2020-01-01*), the time-insensitive KGC models will output the same plausibility scores for these two quadruples. However, the second quadruple is invalid. To exploit temporal information to further improve the performance of KGC models, several studies have been conducted for temporal knowledge graph completion.

Temporal Knowledge Graph Completion extends KGC to support temporal information. Existing methods for temporal knowledge graph completion generally fall into two categories. The first line of researches models entities, relations, and timestamps independently in an identical latent space. TTransE [23], the variant of TransE [4], incorporates temporal representations into a distance-based scoring function. TComplEx [21] is a temporal extension of ComplEx [37] inspired by the canonical decomposition of order 4 tensors and provides a new regularization scheme. TeLM [42] improves on TComplEx by utilizing a linear temporal regularizer and multi-vector embeddings to perform 4th-order tensor factorization of TKGs. ChronoR [32] is a k-dimensional rotation based model that regards relations with timestamps as temporal rotations from head entities to tail entities. The another line argues that temporal information should be implicit in entities or relations, thus learning time-aware representations. ATiSE [44] incorporates temporal information into entities/relations by using additive time series decomposition and exploits the covariance of Gaussian distributions to represent temporal uncertainty. DE-Simple [13] combines the static KGC model Simple [19] with a diachronic embedding function that provides time-aware representations of entities, and utilizes the same scoring function as Simple for temporal KGC. TIE [41] is a time-aware incremental embedding framework that combines representation learning, experience replay, and temporal regularization to improve model performance.

Parameter Generation has been explored in many research fields. Platanios et al. [31] proposed a neural translation model with a contextual parameter generator to generate parameters used by the encoder and decoder for the current sentence based on the source and target languages. N³ [17] generates network parameters for image classification through natural language descriptions combined with pre-trained models. Nekvinda et al. [28] introduced a multilingual speech synthesis method that uses the meta-learning concept of contextual

parameter generation to produce natural-sounding multilingual speech. According to our investigation, there is also work on parameter generation for static knowledge graph completion. CoPER [35] uses the embeddings of relations to generate model parameters that operate on the embeddings of head entities to allow for more complex interactions between entities and relations. ParamE [6] uses neural network parameters as relation embeddings to make the model more expressive and translational. However, CoPER and ParamE are time-agnostic and thus cannot capture the temporal dependencies of facts in TKGs.

3 Methodology

A temporal knowledge graph can be represented by a set of quadruples $\mathcal{G} = \{(h, r, t, \tau) \mid h, t \in \mathcal{E}, r \in \mathcal{R}, \tau \in \mathcal{T}\}$, where \mathcal{E} , \mathcal{R} , and \mathcal{T} are sets of entities, relations, and timestamps, respectively. Each quadruple represents a time-dependent fact that a head entity h connects to a tail entity t with respect to the relation r at the timestamp τ . Given a query $(h, r, ?, \tau)$ or $(?, r, t, \tau)$, TKGC aims to predict the missing tail entity t or head entity h based on the observed temporal facts. For TKGC, we only focus on predicting missing facts at observed timestamps, i.e., interpolation task [18]. The extrapolation task that predicts future facts is not considered in this paper.

To tackle the challenges of TKGC, we propose a Space Adaptation Network (SANE), in which snapshots with different timestamps are adapted for different latent spaces. As shown in Fig. 2, SANE mainly consists of two modules, i.e., a Dynamic Convolutional Neural Network (DCNN), and a Time-aware Parameter Generator (TaPG). DCNN encodes entities and relations into different latent spaces in terms of convolutional layers equipped with different parameters. These parameters are produced by TaPG according to temporal information. TaPG transforms the timestamps into a set of DCNN parameters, where the timestamps dominate the overlap of multiple latent spaces in DCNN, such that the valid knowledge is shared across adjacent snapshots. Specifically, we denote by the d -dimensional vectors $\mathbf{h} \in \mathbb{R}^d$ and $\mathbf{r} \in \mathbb{R}^d$ the head entity and relation respectively. Given a query $(h, r, ?, \tau)$, DCNN f predicts the correct tail entity t based on generated parameters from TaPG g , i.e.,

$$\mathbf{t} = f(\mathbf{h}, \mathbf{r}; g(\tau)), \quad (1)$$

where $g(\tau)$ is the set of parameters of DCNN f , i.e., $\theta_f = g(\tau)$.

3.1 Dynamic Convolutional Neural Network

Convolutional neural networks (CNN) have shown expressiveness in static KGC methods [8, 38], but have not been extensively explored in existing TKGC methods. We extend CNN to support TKGC by endowing CNN with specific parameters associated with temporal information. DCNN f consists of several dynamic convolutional layers and batch normalization, followed by a connected linear layer. Dynamic convolutional layer (DCL) is the important backbone of DCNN

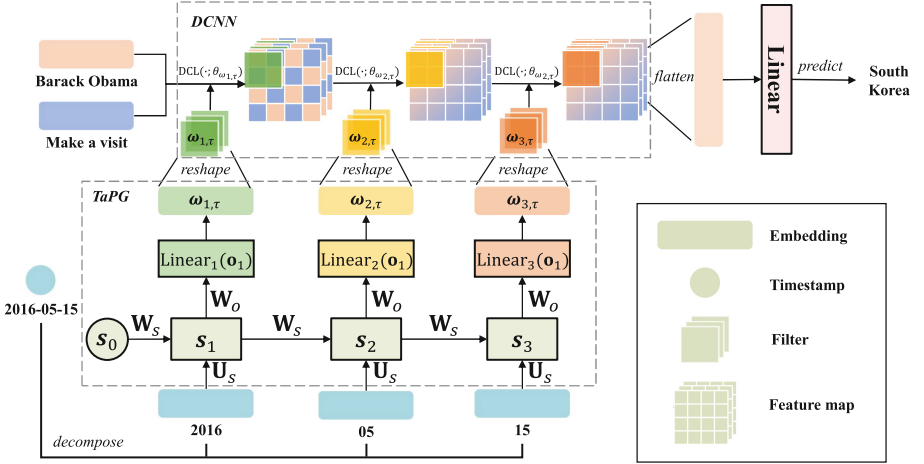


Fig. 2. The framework of our model SANE. DCNN is a multi-layer convolutional neural network for predicting missing entities, and its filter parameters are generated by TaPG based on temporal information.

that identifies the key feature from the inputs based on a filter. It differs from traditional convolutional layer that the parameters of DCL filter is dynamically produced from TaPG instead of fixed. Naturally, TaPG is a tremendous parameter pool and selects appropriate parameters for DCLs when dealing with different temporal facts. DCL pads and filters the input $\mathbf{X} \in \mathbb{R}^{C_i \times H \times W}$ to produce the feature map $\mathbf{X}' \in \mathbb{R}^{C_o \times H \times W}$ based on the filter $\omega_{p,\tau} \in \mathbb{R}^{C_o \times C_i \times k \times k}$ followed by the nonlinear activation function ReLU (i.e., Rectified Linear Unit [12]),

$$\mathbf{X}' = \text{DCL}(\mathbf{X}; \theta_{\omega_{p,\tau}}) = \text{ReLU}(\mathbf{X} \circledast \omega_{p,\tau}), \tag{2}$$

where \circledast is the convolution operator, H and W are height and width, C_o and C_i are the size of input and output channels, and k is the kernel size. The filter $\omega_{p,\tau}$ is produced from TaPG according to the position p of DCL in DCNN and the timestamp τ , i.e., $\theta_{\omega_{p,\tau}} = g(\tau, p)$.

Multiple DCLs are stacked to handle the entities and relations in an effective way. In particular, we first reshape the entity \mathbf{h} and the relation \mathbf{r} into $\tilde{\mathbf{h}} \in \mathbb{R}^{H \times W}$ and $\tilde{\mathbf{r}} \in \mathbb{R}^{H \times W}$, respectively. To enhance the heterogenous interactions between entity $\tilde{\mathbf{h}}$ and relation $\tilde{\mathbf{r}}$ vectors, we perform feature permutation and checkered reshaping operations on the concatenation $\mathbf{X} \in \mathbb{R}^{2H \times W}$ of $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{r}}$ inspired by the work [38]. Feature permutation shuffles each element in $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{r}}$, while checkered reshaping ensures that every two adjacent cells in \mathbf{X} are alternately occupied by elements in \mathbf{h} and \mathbf{r} . The regularized input $\tilde{\mathbf{X}}$ after above operations is fed into P DCLs to produce the feature map \mathbf{M} .

To predict the correct tail entity, a scoring function is introduced to evaluate the score of correlation between the query $(h, r, ?, \tau)$ and candidate tail entity $\mathbf{t} \in \mathbb{R}^d$,

$$\psi_\tau(h, r, t) = \text{Linear}(\text{flatten}(\mathbf{M}))\mathbf{t}, \quad (3)$$

where $\text{Linear}(\cdot)$ is a linear layer activated by ReLU and $\text{flatten}(x)$ flattens x into a 1-dimensional vector.

The sets of filters of DCNN $\{\omega_{1,\tau}, \dots, \omega_{P,\tau}\}$ reflect the delivery and variation of knowledge at different snapshots in consecutive time. The spaces induced by DCNN at different timestamps should be same, overlapped or uncorrelated when the timestamps of facts are the same, adjacent and distant. In other words, the overlap of spaces at different timestamps constrains the range of knowledge sharing. This property ensures that the interfere from early snapshots is alleviated and the missing facts in adjacent snapshots are delivered to accumulate knowledge. Essentially, our SANE model stores the facts in multiple knowledge bases, i.e., multiple sets of parameters, depending on the time range. Thus, it can “index” the knowledge precisely by finding the “records” in parameters according to different timestamps. The next section will introduce the parameter generation of DCNN to preserve the valid knowledge and forget the mistaken in a time-aware way.

3.2 Time-Aware Parameter Generator

Usually, in the process of searching records by human, the searcher reduces the hunting zone by gradually indexing year, month and day. For example, if a person wants to query a record that are indexed by the timestamp, he needs to split timestamps into year, month and day to locate it. If the record is missing at the timestamp, the similar records around the timestamp should be returned. Based on the observation, the filter parameters $\omega_{1,\tau}$ of the first DCL in DCNN are required to establish a global “catalogue” of the year of τ . The catalogue encodes high-level contextual features with an annual perspective. After that, the second and third of DCLs predict the facts by supplementing more details of month and day information based on the parameters $\omega_{2,\tau}$ and $\omega_{3,\tau}$.

In this part, we introduce a time-aware parameter generator (TaPG) that “store” the knowledge in three sets of parameters that are associated with “year-month-day”. Specifically, we first split and embed the timestamp τ as a fixed-length sequence $\vec{\tau} = (\tau_1, \tau_2, \tau_3)$, where $\tau_1, \tau_2, \tau_3 \in \mathbb{R}^{d_\tau}$ are the embeddings of year, month and day respectively. A recurrent neural network (RNN) is introduced to model the sequence data $\vec{\tau}$ that produces multiple outputs,

$$\{\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3\} = \text{RNN}(\vec{\tau}), \quad (4)$$

$$\mathbf{o}_i = \sigma(\mathbf{W}_o \mathbf{s}_i + \mathbf{b}_o), \quad (5)$$

$$\mathbf{s}_i = \sigma(\mathbf{U}_s \tau_i + \mathbf{W}_s \mathbf{s}_{i-1} + \mathbf{b}_s), \quad (6)$$

where \mathbf{W}_o , \mathbf{W}_s and \mathbf{U}_s are RNN parameters, \mathbf{s}_i and \mathbf{o}_i are the hidden state and output at step i , and σ is the nonlinear activation function.

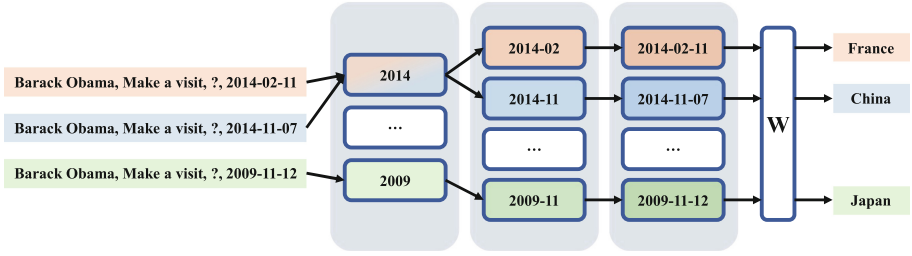


Fig. 3. The process of query prediction by DCNN. Queries with different timestamps are handled by different parameters, while queries with similar timestamps partially share model parameters.

Multiple fully connected layers $\{\text{Linear}_1, \text{Linear}_2, \text{Linear}_3\}$ are employed to transform the outputs of RNN into a set of parameters,

$$g(\tau) = \{\omega_{i,\tau}\} = \{\text{Linear}_i(\mathbf{o}_i)\}. \tag{7}$$

The linear layers $\{\text{Linear}_i\}$ are a parameter pool, which retrieves the parameters according to the context of timestamps. The 1-dimensional vectors produced by $\{\text{Linear}_i\}$ are reshaped into tensors in $\mathbb{R}^{C_o \times C_i \times k \times k}$, since the convolutional operations are involved. The scale of DCNN parameters is obviously irrelevant to the number of timestamps, which only depends on the size of the linear layers $\{\text{Linear}_i\}$.

As show in Fig. 3, the facts with the same year are dealt with the same filters $\omega_{1,\tau}$ and thus the valid knowledge during a interval is shared across adjacent snapshots. Compared to the works [40, 45] that construct sparse snapshots at each timestamps explicitly, our implicit way enable the knowledge delivery at different snapshots. Of course, the facts that have long gap of timestamps are divided into two totally different models that avoids the interfere from early knowledge. Therefore, TaPG enables the ability of DCNN to tackle time-variability and time-stability in an efficient manner. Multiple spaces induced by the parameters output from TaPG are adapted for different temporal snapshots. The knowledge at different time is shared or separated during multiple spaces in term of the context from TaPG.

3.3 Training and Optimization

During training process, the score $\psi_\tau(h, r, t)$ is applied with the logistic sigmoid function $\sigma(\cdot)$ to obtain $p = \sigma(\psi_\tau(h, r, t))$. p indicates the predicted probability that the candidate tail entity t is the answer to query $(h, r, ?, \tau)$. The training objective is to minimize the negative log-likelihood loss as follows,

$$\mathcal{L}(y, p) = -\frac{1}{N} \sum_i (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)), \tag{8}$$

Table 1. Scoring functions of SANE and several existing TKGC methods, and comparison of space complexity.

Model	Scoring Function	Space Complexity
TransE	$\ \mathbf{h} + \mathbf{r} - \mathbf{t}\ $	$\mathcal{O}(n_e d + n_r d)$
TTransE	$\ \mathbf{h} + \mathbf{r} + \tau - \mathbf{t}\ $	$\mathcal{O}(n_e d + n_r d + n_\tau d)$
HyTE	$\ P_\tau(\mathbf{h}) + P_\tau(\mathbf{r}) - P_\tau(\mathbf{t})\ $	$\mathcal{O}(n_e d + n_r d + n_\tau d)$
ATiSE	$\mathcal{D}_{\mathcal{KL}}(\mathbf{P}_{h,\tau} - \mathbf{P}_{t,\tau}, \mathbf{P}_{r,\tau})$	$\mathcal{O}(n_e d + n_r d)$
TeRo	$\ \mathbf{h}_\tau + \mathbf{r} - \bar{\mathbf{t}}_\tau\ $	$\mathcal{O}(n_e d + n_r d)$
SANE	$\text{Linear}(\text{flatten}(\text{CNN}(\mathbf{h}, \mathbf{r})))\mathbf{t}$	$\mathcal{O}(n_e d + n_r d + n_y d_\tau)$

where $y = 1$ for positive samples, i.e., $(h, r, t, \tau) \in \mathcal{G}$, otherwise $y = 0$. N indicates the number of training samples.

In Table 1, we summarize the scoring functions and space complexity of several TKGC methods. n_e , n_r , and n_τ are the number of entities, relations, and timestamps, respectively. d and d_τ are the dimensions of feature vectors. n_y is the number of years. CNN refers to the three-layer convolutional neural network in DCNN. In terms of space complexity, SANE is comparable to several existing methods.

4 Experiments

In this section, four TKGC benchmark datasets are used to demonstrate the effectiveness of SANE. The experimental setup is first explained in detail. Then, the experimental results are discussed. Ablation studies are also conducted to evaluate the importance of different components in SANE.

4.1 Experimental Setup

Datasets. The proposed model is evaluated on four public benchmarks, ICEWS14 [10], ICEWS05-15 [10], YAGO11k [7], and Wikidata12k [7]. ICEWS14 and ICEWS05-15 are subsets of the Integrated Crisis Early Warning System (ICEWS) [5] dataset, where ICEWS14 includes events that occurred in 2014, and ICEWS05-15 includes events that occurred in the period 2005 to 2015. ICEWS contains discrete time-annotated sociopolitical events, e.g. (*Barack Obama, Make a visit, South Korea, 2014-03-15*). YAGO11k and Wikidata12k are subsets of YAGO3 [26] and Wikidata [9], respectively. Facts in both YAGO11k and Wikidata12k contain time annotations, and each fact is formatted as a time interval. Following Dasgupta et al. [7], facts with time intervals are discretized into multiple quadruplets with a single timestamp. Meanwhile, month and day information is dropped, and year-level granularity is preserved. To process such datasets, timestamps are appended with constant fabricated months and days, e.g., 2015-00-00. Statistics for these four benchmarks are summarized in Table 2.

Table 2. Statistics of TKGC benchmark datasets. The unit of the time span is year.

Datasets	#Entities	#Relations	Time span	#Train	#Valid	#Test
ICEWS14	6,869	230	2014	72,826	8,941	8,963
ICEWS05-15	10,094	251	2005-2015	386,962	46,275	46,092
YAGO11k	10,623	10	-453-2844	16,408	2,050	2,051
Wikidata12k	12,554	24	1709-2018	32,497	4,062	4,062

Baselines. We compare with a wide selection of static and temporal KGC models: (1) static KGC models, including TransE [4], DistMult [46], ComplEx-N3 [22], RotatE [36], and QuatE² [47]; (2) temporal KGC models, including TTransE [23], HyTE [7], TA-TransE [10], TA-DistMult [10], DE-Simple [13], ATiSE [44], TeRo [43], ChronoR [32], TimePlex [15], TComplEx [21], TeLM [42], and BoxTE [27]. Among them, ChronoR and BoxTE are not compared with SANE on YAGO11k and Wikidata12k, because their results are unobtainable.

Evaluation Protocols. For each quadruple (h, r, t, τ) in the test set, two queries $(h, r, ?, \tau)$ and $(?, r, t, \tau)$ are leveraged to optimize the model simultaneously. Note that in practice, each quadruple (h, r, t, τ) is added with a reciprocal relation (t, r^{-1}, h, τ) . Thus, the query $(?, r, t, \tau)$ is replaced by $(t, r^{-1}, ?, \tau)$. Such operations do not result in a loss of generality [15, 42]. MRR (Mean Reciprocal Rank, the average of the reciprocal values of all computed ranks) and Hits@N (the percentage of times that the true entity candidate appears in the top N of ranked candidates, where $N \in \{1, 3, 10\}$) are reported as evaluation metrics. Among them, MRR is an important evaluation index, which is less susceptible to outliers [10]. Higher MRR and Hits@N indicate better model performance. All evaluations are performed under the time-wise filtering setting widely adopted in previous work [43, 44].

Implementation Details. The proposed model is implemented using PyTorch [30] and trained using a single NVIDIA GeForce RTX 3090 GPU. The values of the hyperparameters are determined based on the MRR performance on each validation set. The model parameters are initialized using Xavier initialization [11] and optimized by the Adam optimizer [20] with a learning rate of 0.001. During training, 256 mini-batches are created for each epoch. The negative sampling ratio is set to 1000, i.e., 1000 negative samples are created for each quadruple in the training set. The embedding dimension is set to $d = 200$ for all datasets except ICEWS05-15 which is set to $d = 300$. The number of convolution filters is fixed to 64. The kernel size is chosen from $k \in \{3, 5, 7\}$.

4.2 Main Results

The MRR and Hits@N results on ICEWS dataset, i.e., ICEWS14 and ICEWS05-15, are reported in Table 3. Some observations and analysis are listed as follows.

(1) Most of TKGC models achieve significantly better results than static KGC

Table 3. Link prediction results on ICEWS14 and ICEWS05-15. *: results are taken from [10]. †: results are taken from [43]. ∘: results are taken from [42]. Dashes: results are unobtainable. Other results are taken from the original papers. The best results are marked in **bold**.

Datasets Metrics	ICEWS14				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE* [4]	.280	.094	–	.637	.294	.090	–	.663
DistMult* [46]	.439	.323	–	.672	.456	.337	–	.691
ComplEx-N3† [22]	.467	.347	.527	.716	.481	.362	.535	.729
RotatE† [36]	.418	.291	.478	.690	.304	.164	.355	.595
QuatE2† [47]	.471	.353	.530	.712	.482	.370	.529	.727
TTransE† [23]	.255	.074	–	.601	.271	.084	–	.616
HyTE† [7]	.297	.108	.416	.655	.316	.116	.445	.681
TA-TransE* [10]	.275	.095	–	.625	.299	.096	–	.668
TA-DistMult* [10]	.477	.363	–	.686	.474	.346	–	.728
DE-SimpleE† [13]	.526	.418	.592	.725	.513	.392	.578	.748
ATiSE [44]	.545	.423	.632	.757	.533	.394	.623	.803
TeRo [43]	.562	.468	.621	.732	.586	.469	.668	.795
ChronoR [32]	.625	.547	.669	.773	.675	.596	.723	.820
TimePlex [15]	.604	.515	–	.771	.640	.545	–	.818
TComplEx [∘] [21]	.610	.530	.660	.770	.660	.590	.710	.800
TeLM [∘] [42]	.625	.545	.673	.774	.678	.599	.728	.823
BoxTE [27]	.613	.528	.664	.763	.667	.582	.719	.820
SANe	.638	.558	.688	.782	.683	.605	.734	.823

methods. TKGC models leverage temporal information to constrain the similarity of facts, such that similar facts with different timestamps are separate efficiently. (2) SANe achieves the best performance for all metrics on link prediction, which suggests the effectiveness of adapting snapshots with different timestamps to different latent spaces. The facts are implicitly assigned to different CNN modules, and thus each snapshot at different timestamps is handled in term of a specific latent space. The results indicate that the parameter generation plays an important role in alleviating mutual interference of the knowledge across snapshots with different timestamps. (3) Facts in ICEWS are transient events, which usually happen and end in a moment. Compared to other TKGC methods, SANe is capable of remembering and inferring instant facts by recovering the CNN model from the parameter pool according to timestamps. The result in Table 3 further certifies that SANe is more effective to enable time-variability that inherent in TKGs.

Table 4 shows the prediction performance over Wikipedia-based datasets, i.e., YAGO11k and Wikidata12k. SANe achieves superior performance over previous methods by a large margin compared to the result on ICEWS. On MRR, a main

Table 4. Link prediction results on YAGO11k and Wikidata12k. *: results are taken from [44]. †: results are taken from [43]. ◊: results are taken from [42]. Dashes: results are unobtainable. Other results are taken from the original papers. The best results are marked in **bold**.

Datasets	YAGO11k				Wikidata12k			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE* [4]	.100	.015	.138	.244	.178	.100	.192	.339
DistMult* [46]	.158	.107	.161	.268	.222	.119	.238	.460
CompLex-N3* [22]	.167	.106	.154	.282	.233	.123	.253	.436
RotatE* [36]	.167	.103	.167	.305	.221	.116	.236	.461
QuatE ^{2*} [47]	.164	.107	.148	.270	.230	.125	.243	.416
TTransE [†] [23]	.108	.020	.150	.251	.172	.096	.184	.329
HyTE [†] [7]	.105	.015	.143	.272	.180	.098	.197	.333
TA-TransE [†] [10]	.127	.027	.160	.326	.178	.030	.267	.429
TA-DistMult [†] [10]	.161	.103	.171	.292	.218	.122	.232	.447
ATiSE [44]	.185	.126	.189	.301	.252	.148	.288	.462
TeRo [†] [43]	.187	.121	.197	.319	.299	.198	.329	.507
TimePlex [15]	.236	.169	–	.367	.334	.228	–	.532
TCompLex [◊] [21]	.185	.127	.183	.307	.331	.233	.357	.539
TeLM [◊] [42]	.191	.129	.194	.321	.332	.231	.360	.542
SANe	.250	.180	.266	.401	.432	.331	.483	.640

metric for the TKGC task, SANe outperforms by 6% and 29% dramatically compared with the state-of-the-art methods across the YAGO11k and Wikidata12k, respectively. The facts in Wikipedia-based datasets spans a period of hundreds of years, even around 3,000 years, while ICEWS only covers several years. The plenty of facts usually last for a long period of time different from ICEWS that events happen and end in a moment. The superior result of SANe reveals the necessity of designing a more principled parameter generation approach to produce multiple latent spaces that constrains the range of knowledge sharing based on timestamp distance. Multiple sets of parameters encode the context of timestamps that the knowledge in adjacent snapshots is delivered to accumulate knowledge. Therefore, the valid knowledge during a period can be preserved and shared efficiently. The models of learning independent representations [15, 21, 23, 32, 42] or incorporating timestamp into entities and relations [7, 10, 13, 27, 43, 44] suffer from the interfere across snapshots particularly when the knowledge last for a long period. This is mainly because they handle all the facts in an identical latent space, and thus inevitably misremember and forget knowledge. The result in Table 4 further certifies that SANe is more effective to enable time-stability inherent in TKGs.

4.3 Analysis

Ablation Study. To better verify the effectiveness of the proposed model, several variants of SANE are investigated on ICEWS14. The results are shown in Table 5. The \checkmark is used to indicate a component used in the experiment, and the \times is used to indicate the absence of the corresponding component. The SANE without time information means that TaPG returns a fixed set of parameters regardless of the timestamps. Based on the SANE without time information, the SANE without parameter generation also incorporates the temporal information into the entities, i.e., $\hat{\mathbf{h}} = \mathbf{h} \odot \boldsymbol{\tau}$, where \odot is Hadamard product as the work [34] does. The TaPG of SANE without time granularity produces the parameters directly in term of timestamp $\boldsymbol{\tau}$ without RNN, i.e., $\{\text{Linear}_i(\boldsymbol{\tau})\}$. It is found that (1) when the time information is not used, the model achieves the worst results, which reflects the importance of time information to SANE. (2) The parameter generator has a great influence on the model performance, which verifies the effectiveness of the time-aware parameter generator. (3) Decomposing timestamps into different time granularities is beneficial to the improvement of model performance. (4) Even if the timestamps are not decomposed into different granularities, the model achieves better results than previous TKGC methods, which confirms the superiority of the model.

Table 5. Results for different model variations on ICEWS14.

Time Information	Time Granularity	Parameter Generation	MRR	Hits@1	Hits@3	Hits@10
\times	\times	\times	.469	.350	.529	.703
\checkmark	\times	\times	.608	.527	.656	.760
\checkmark	\checkmark	\times	.622	.536	.679	.778
\checkmark	\times	\checkmark	.630	.548	.683	.780
\checkmark	\checkmark	\checkmark	.638	.558	.688	.782

Table 6. Generalization performance for queries with unseen timestamps on the ICEWS14 dataset.

Metrics	MRR	Hits@1	Hits@3	Hits@10
DistMult [46]	.410	.302	.462	.620
DE-Simple [13]	.434	.333	.492	.624
TComplex [21]	.443	.348	.492	.625
SANE	.503	.394	.569	.709

Generalizing to Unseen Timestamps. Since timestamps are decomposed at different time granularities in SANE, this allows queries with similar timestamps to share a part of filter parameters and temporal information. Therefore, SANE

is expected to perform well on queries with unseen timestamps. Following Goel et al. [13], we re-split ICEWS14, taking all quadruplets except the 5th, 15th, and 25th day of each month as the training set, and using the excluded quadruplets to randomly split into validation and test sets. The obtained results in Table 6 indicate that SANE gains almost 14% MRR improvement over TComplex [21], thus showing the effectiveness of our model to generalize to unseen timestamps.

Performance on Different Relations. Most of the time annotations in YAGO11k are time intervals, and the relations between entities may change after a period of time. We evaluate SANE on several relations (*worksAt*, *hasWonPrize*, *graduatedFrom*, and *isAffiliatedTo*) in YAGO11k, and reproduce ATiSE [44] and TimPlex [15] based on their given hyperparameters. These relations are usually created or disappeared between some entities at a certain point in time, and maintained for a period of time [44]. For example, a person may switch to another company after working for one company for a few months. SANE is expected to perform well in such relations. The comparisons in Fig. 4 show that SANE is superior in almost all metrics. This confirms our hypothesis that adapting different temporal snapshots to different latent spaces via parameter generation is beneficial for capturing the time-variability of knowledge. Likewise,

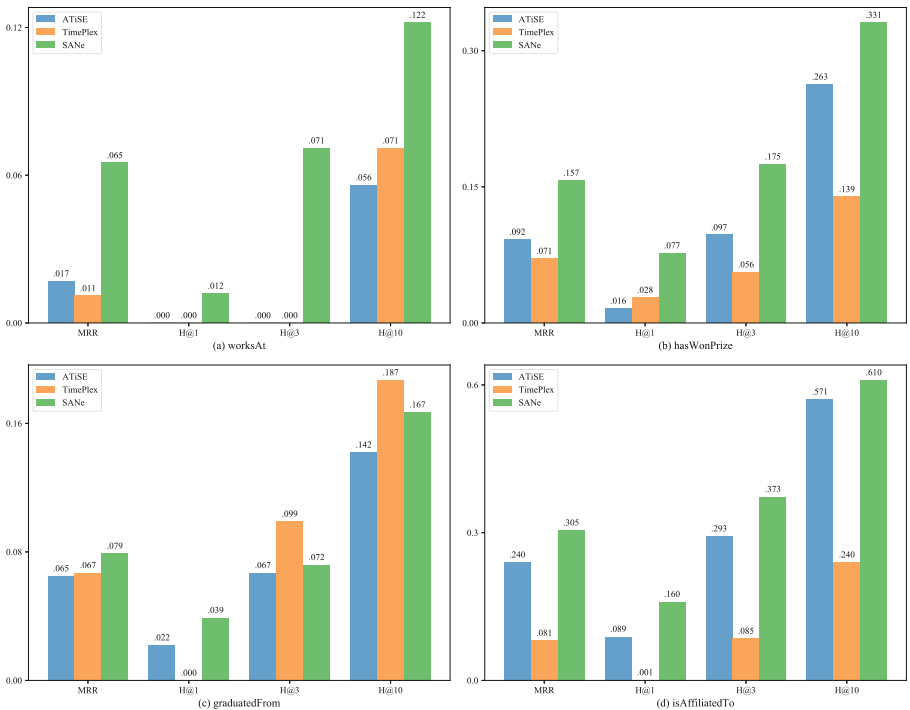


Fig. 4. Results obtained by ATiSE [44], TimePlex [15], and SANE on several relations in YAGO11k.

overlapping latent spaces by decomposing timestamps into different granularities facilitates knowledge sharing across adjacent snapshots, which is beneficial for modeling the time-stability of knowledge.

Visualization of Temporal Embeddings. Figure 5 shows a t-SNE [25] visualization of the temporal embeddings learned by SANE and its variant. Figure 5(a) visualizes the temporal embeddings learned by the variant of SANE. Timestamps are modeled independently by the variant of SANE rather than decomposed into different granularities. Figure 5(b) visualizes the temporal embeddings with granularity of 1 day learned by SANE. By comparison, it can be found that the temporal embeddings learned by SANE form good clusters in chronological order. In general, SANE effectively preserves time series information by decomposing timestamps into different granularities and processed by the time series model, which provides good geometric meanings for temporal embeddings, thus improving the model performance.

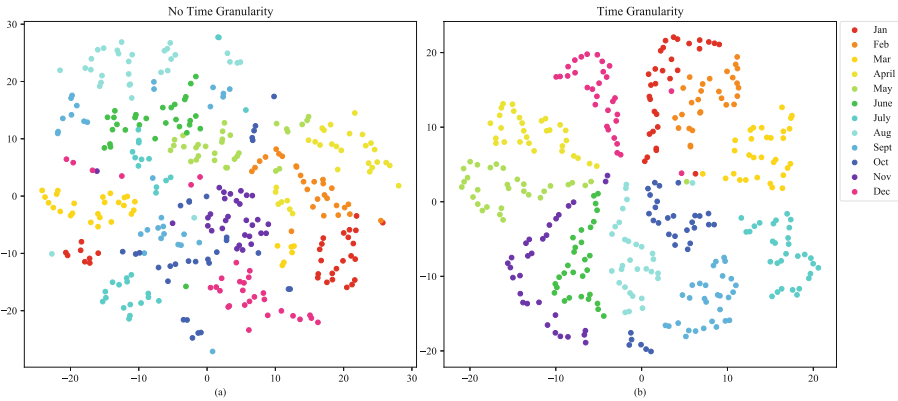


Fig. 5. The figure illustrates the t-SNE visualization of the temporal embeddings obtained by SANE and its variant after training on ICEWS14. Time points in different months are represented by different colors.

5 Conclusion and Future Work

In this paper, we shed a new light on the challenges of TKGC. For the first time, we proposed to investigate the problem of TKGC by adapting different latent spaces for snapshots at different timestamps. Specifically, we provided a novel model named SANE to process the entities and relations using a dynamic convolutional neural network equipped with different parameters, which are produced by TaPG according to temporal information. TaPG endowed with contextual timestamps gathers valid knowledge from multiple sets of parameters by constraining the overlap of spaces. Our model is different from existing works which learn the temporal KGs all the time in the same latent space. The experimental results demonstrate the benefits of constructing parameter-independent model implicitly for each temporal snapshot.

Supplemental Material Statement: Source code to reproduce the full experimental results is already available on the Easychair system and will be published on <https://github.com/codeofpaper/SANe>.

Acknowledgments. This work is supported by the Fund of the State Key Laboratory of Software Development Environment.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Balažević, I., Allen, C., Hospedales, T.: Tucker: tensor factorization for knowledge graph completion. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5185–5194 (2019)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250 (2008)
4. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2, pp. 2787–2795 (2013)
5. Boschee, E., Lautenschlager, J., O’Brien, S., Shellman, S., Starz, J., Ward, M.: Icews coded event data. Harvard Dataverse 12 (2015)
6. Che, F., Zhang, D., Tao, J., Niu, M., Zhao, B.: Parame: regarding neural network parameters as relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2774–2781 (2020)
7. Dasgupta, S.S., Ray, S.N., Talukdar, P.: Hyte: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2001–2011 (2018)
8. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
9. Erxleben, F., Günther, M., Kröttsch, M., Mendez, J., Vrandečić, D.: Introducing wikidata to the linked data web. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 50–65. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_4
10. Garcia-Duran, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4816–4821 (2018)
11. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)

12. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 315–323. JMLR Workshop and Conference Proceedings (2011)
13. Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3988–3995 (2020)
14. Guo, Q., et al.: A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data Eng.* (2020)
15. Jain, P., Rath, S., Chakrabarti, S., et al.: Temporal knowledge base completion: New algorithms and evaluation protocols. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3733–3747 (2020)
16. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (volume 1: Long papers), pp. 687–696 (2015)
17. Jin, T., Liu, Z., Yan, S., Eichenberger, A., Morency, L.P.: Language to network: Conditional parameter adaptation with natural language descriptions. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
18. Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: autoregressive structure inference over temporal knowledge graphs. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6669–6683 (2020)
19. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 4289–4300 (2018)
20. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of 3rd International Conference on Learning Representations (2015)
21. Lacroix, T., Obozinski, G., Usunier, N.: Tensor decompositions for temporal knowledge base completion. In: International Conference on Learning Representations (2020)
22. Lacroix, T., Usunier, N., Obozinski, G.: Canonical tensor decomposition for knowledge base completion. In: International Conference on Machine Learning, pp. 2863–2872. PMLR (2018)
23. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion Proceedings of the The Web Conference 2018, pp. 1771–1776 (2018)
24. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
25. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**(11) (2008)
26. Mahdisoltani, F., Biega, J., Suchanek, F.: Yago3: a knowledge base from multilingual wikipe-dias. In: 7th Biennial Conference on Innovative Data Systems Research. CIDR Conference (2014)
27. Messner, J., Abboud, R., Ceylan, I.I.: Temporal knowledge graph completion using box embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence (2022)
28. Nekvinda, T., Dušek, O.: One model, many languages: meta-learning for multilingual text-to-speech. *Proc. Interspeech* **2020**, 2972–2976 (2020)

29. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning, pp. 809–816 (2011)
30. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 8026–8037 (2019)
31. Platanios, E.A., Sachan, M., Neubig, G., Mitchell, T.: Contextual parameter generation for universal neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 425–435 (2018)
32. Sadeghian, A., Armandpour, M., Colas, A., Wang, D.Z.: Chronor: rotation based temporal knowledge graph embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 6471–6479 (2021)
33. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4498–4507 (2020)
34. Shao, P., Zhang, D., Yang, G., Tao, J., Che, F., Liu, T.: Tucker decomposition-based temporal knowledge graph completion. *Knowl.-Based Syst.* **238**, 107841 (2022)
35. Stoica, G., Stretcu, O., Platanios, E.A., Mitchell, T., Póczos, B.: Contextual parameter generation for knowledge graph link prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3000–3008 (2020)
36. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2019)
37. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48, pp. 2071–2080 (2016)
38. Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., Talukdar, P.: Interact: improving convolution-based knowledge graph embeddings by increasing feature interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3009–3016 (2020)
39. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
40. Wu, J., Cao, M., Cheung, J.C.K., Hamilton, W.L.: Temp: temporal message passing for temporal knowledge graph completion. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5730–5746 (2020)
41. Wu, J., Xu, Y., Zhang, Y., Ma, C., Coates, M., Cheung, J.C.K.: Tie: a framework for embedding-based incremental temporal knowledge graph completion. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 428–437 (2021)
42. Xu, C., Chen, Y.Y., Nayyeri, M., Lehmann, J.: Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2569–2578 (2021)
43. Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H.S., Lehmann, J.: Tero: a time-aware knowledge graph embedding via temporal rotation. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1583–1593 (2020)

44. Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H., Lehmann, J.: Temporal knowledge graph completion based on time series gaussian embedding. In: Pan, J.Z., et al. (eds.) ISWC 2020. LNCS, vol. 12506, pp. 654–671. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62419-4_37
45. Xu, Y., et al.: Rtfte: a recursive temporal fact embedding framework for temporal knowledge graph completion. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5671–5681 (2021)
46. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015 (2015)
47. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 2735–2745 (2019)
48. Zhang, Z., Zhuang, F., Zhu, H., Shi, Z., Xiong, H., He, Q.: Relational graph neural network with hierarchical attention for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9612–9619 (2020)