Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.069331

ARTICLE

Learning Time Embedding for Temporal Knowledge Graph Completion

Jinglu Chen[1], **Mengpan Chen**[2], **Wenhao Zhang**[2,*], **Huihui Ren**[2] **and Daniel Dajun Zeng**[1,2]

[1]College of Management and Economics, Tianjin University, Tianjin, 300072, China
[2]The State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

*Corresponding Author: Wenhao Zhang. Email: wenhao.zhang@ia.ac.cn

Received: 20 June 2025; Accepted: 18 September 2025; Published: 09 December 2025

ABSTRACT: Temporal knowledge graph completion (TKGC), which merges temporal information into traditional static knowledge graph completion (SKGC), has garnered increasing attention recently. Among numerous emerging approaches, translation-based embedding models constitute a prominent approach in TKGC research. However, existing translation-based methods typically incorporate timestamps into entities or relations, rather than utilizing them independently. This practice fails to fully exploit the rich semantics inherent in temporal information, thereby weakening the expressive capability of models. To address this limitation, we propose embedding timestamps, like entities and relations, in one or more dedicated semantic spaces. After projecting all embeddings into a shared space, we use the relation-timestamp pair instead of the conventional relation embedding as the translation vector between head and tail entities. Our method elevates timestamps to the same representational significance as entities and relations. Based on this strategy, we introduce two novel translation-based embedding models: TE-TransR and TE-TransT. With the independent representation of timestamps, our method not only enhances capabilities in link prediction but also facilitates a relatively underexplored task, namely time prediction. To further bolster the precision and reliability of time prediction, we introduce a granular, time unit-based timestamp setting and a relation-specific evaluation protocol. Extensive experiments demonstrate that our models achieve strong performance on link prediction benchmarks, with TE-TransR outperforming existing baselines in the time prediction task.

KEYWORDS: Temporal knowledge graph (TKG); TKG embedding model; link prediction; time prediction

1 Introduction

Knowledge Graphs (KGs) are large multilateral relationship networks where nodes depict entities and edges capture the relations among them. In KGs, knowledge is organized into triples (*head entity, relation, tail entity*) which can be expressed as (h, r, t), e.g., (*Obama, presidentOf, USA*). There are numerous well-known open sourced KGs, including WordNet [1], Freebase [2], DBpedia [3], NELL [4], YAGO [5], OpenIE [6], etc. KGs are valuable for a variety of applications, especially in AI-related fields, such as language modeling [7], recommender systems [8,9], question answering [10,11], information retrieval [12,13] and reasoning [14,15].

It should be noted that KG facts are not static and permanently true, but are typically valid within specific temporal bounds. For example, (*Ban Ki-moon, secretaryGeneralOf, the United Nations*) was true between 2007 and 2016, and (*Obama, presidentCandidateOf, USA*) was true in 2008 and 2012. Since temporal characteristics are crucial for tracing the evolution of real-world events, the temporal knowledge graph (TKG) has been put forward and has attracted lots of interest. ICEWS [16], Wikidata [17], GDELT [18], and YAGO [5] are typical TKGs that incorporate a large amount of temporal facts in quadruple format (h, r, t, τ)

where τ is the timestamp, e.g., (*Ban Ki-moon, secretaryGeneralOf, United Nations, [2007, 2016]*), (*Obama, presidentCandidateOf, USA, 2008*).

Given the incompleteness of most TKGs, the capability to infer absent information is essential for the utilization of TKGs. KG embedding is among the most effective methods for completing KGs, aiming to map KGs into continuous vector spaces while retaining their main characteristics [19]. Traditional static knowledge graph (SKG) embedding models are proven to be useful in inferring new facts for SKGs [20–24], but they do not consider the time dimension. In order to handle time-sensitive facts, TKG embedding models integrate temporal information and achieve better performance [25–29]. Among these methods, translation-based methods are a typical and important field, which attempts to minimize the distance between the translated head entity and the tail entity in the embedding space [20–22,26,29].

Despite the demonstrated effectiveness of translation-based methods for TKGC, several challenges remain: (1) Existing methods normally embed timestamps into entities or relations and utilize only the relation as the translation vector, which neglects the important value of temporal information and limits the model's expressive capability. (2) Many methods assume that entities, relations, and timestamps reside in the same semantic space and build the translation without any transformations, leading to the loss of semantics of different elements and the lack of information interaction between them. (3) Previous methods often fail to learn independent time embeddings to fully exploit the semantics inherent in temporal information, thus constraining the model's ability to perform time prediction. While some methods may address certain aspects, a unified solution has yet to emerge.

In this work, to address the above problems comprehensively, we embed timestamps in one or more dedicated embedding spaces to adequately leverage the rich semantics of temporal information. Additionally, we model the relation-timestamp pair as the translation vector instead of modeling only the relation, thus enhancing the flexibility and expressive capability of the model. We propose two novel translation-based models for TKGC, referred to as TE-TransR and TE-TransT. In both models, we learn the embeddings of entities, relations, and timestamps in distinct semantic spaces, respectively. Various embedding vectors can be mapped into a unified space through either relation-specific or time-specific matrices, in which translations are built. Concretely, in TE-TransR, we model an entity space, a time space, and multiple relation spaces. Entity embeddings and time embeddings are mapped into the corresponding relation space through relation-specific matrices. In TE-TransT, we model multiple time spaces, into which entity embeddings and relation embeddings are projected through time-specific matrices. These transformation matrices facilitate information interaction between different embedding spaces, thereby reducing semantic loss. Furthermore, our method can learn independent time embeddings within distinct spaces, which enhances the ability to make precise time predictions.

We also observe that existing methods do not segment timestamps based on fundamental time units, but instead divide time into intervals of varying lengths, based on a threshold number of facts [26]. For simplicity, these time intervals are still denoted as $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_n\}$. With such a timestamp setting, previous methods can only predict vaguely predefined time intervals instead of precise time points for time prediction. To tackle this limitation, we propose a new timestamp setting that segments timestamps according to the original time unit of the dataset (e.g., each timestamp τ corresponds to an exact year in YAGO11k and Wikidata12k [26]), enabling flexible prediction of specific time points for facts. Moreover, we find that the duration of a fact is closely tied to the associated relation. Therefore, we introduce a relation-specific evaluation protocol with a new evaluation metric that allows for a fair assessment of time prediction, tailored to the average duration of the relation involved in each fact.

We conduct a wide range of experiments on several real-world datasets, including link prediction, time prediction, and other analytical experiments. The outcomes of our experiments validate the effectiveness of our method. Our key contributions are summarized below:

- We propose two novel translation-based models to learn embedding vectors for TKGC. In these models, we (1) independently learn entity, relation, and time embeddings in distinct semantic spaces, (2) utilize relation-specific or time-specific matrices for transformations between spaces, and (3) model the combination of relation and timestamp as the translation vector.
- We introduce a novel time prediction scheme, which includes a time unit-based timestamp setting and a relation-specific evaluation protocol, enabling more precise and reliable time prediction for TKGC. Our evaluation protocol provides an evaluation metric that rewards accurate time point predictions that align with the duration of the involved relation.
- We conduct extensive experiments, including comprehensive comparisons with existing models and detailed qualitative analysis, to demonstrate the effectiveness of our method.

2 Related Work

This section reviews the related work on SKG and TKG embedding models.

2.1 SKG Embedding Models

The task of SKGC focuses on inferring missing links within a KG that does not change over time. A diverse range of SKG embedding models has been proposed for SKGC, evolving from classic translation mechanisms to sophisticated neural architectures.

The earliest translation-based model, TransE [20], pioneered the idea of modeling the relation as the translation vector between entities in the embedding space. However, TransE is limited to handling 1-on-1 relations. To address this, several extensions of TransE were proposed [19,22,30]. For instance, TransH [19] allows entity embeddings to vary across relation-specific hyperplanes, while TransR [22] separates entity and relation spaces through projection matrices. RotatE [31] further represents relations as rotations in the complex space to capture a wider variety of relation patterns. Later, BoxE [32] proposes a bounded region representation for relations, enhancing the expressiveness for logical reasoning.

In parallel, semantic matching models gained traction by focusing on scoring functions to evaluate the plausibility of a triple. DistMult [21] applies bilinear scoring to assess triple plausibility, while ComplEx [23] generalizes this formulation to the complex space. Models like SimplE [24] and TuckER [33] further refine these ideas by leveraging tensor decompositions. CapsE [34], built upon capsule networks, captures dimension-wise interactions within entity and relation embeddings. Additionally, methods such as McRL [35] and MLI [36] explore multi-level semantics or conceptual abstraction to improve representation fidelity.

Neural network-based approaches have also been integrated into SKGC. ConvE [37] utilizes 2D convolutions to model entity-relation interactions in a spatial context. R-GCN [38] introduces message-passing mechanisms based on relational graph convolutional networks (GCN). Innovative designs such as M-DCN [39] and TDN [40] further explore adaptive message distribution and dynamic neighborhood encoding to improve representation learning in SKGs.

2.2 TKG Embedding Models

Even though the models listed above can obtain promising results on link prediction of SKGs, they fail to capture the temporal information of TKGs. To address this time-sensitive challenge, various TKG embedding models are proposed for TKGC. Existing approaches are commonly categorized according to their differing strategies in incorporating temporal information.

The first category primarily evolves from SKGC methodologies, directly incorporating the temporal dimension within established embedding frameworks. The translation-based method t-TransE [41] extends classical TransE [20] by learning the temporal order of relations. TTransE [25] expands TransE by embedding relation-timestamp pairs as translation vectors between entity embeddings, while simply assuming that timestamps share the common space with entities and relations. RTS [29] is also an extension of TransE, transforming relation embeddings from the relation space into the entity space through timestamp-specific matrices. Extending complex-space representations, TeRo [42] encodes timestamps as rotations affecting entities within the complex embedding space, capturing dynamic entity changes. Similarly, ChronoR [43], based upon the RotatE [31] framework, represents the interaction of relations and timestamps as rotations connecting head and tail entity embeddings. TComplEx [44] and TuckERTNT [45] expand traditional third-order embeddings into fourth-order tensor formats to encapsulate temporal interactions on the basis of ComplEx [23] and TuckER [33], respectively, while TNTComplEx [44] innovatively differentiates between temporally evolving facts and those that remain stable across timestamps, providing a nuanced representation of temporal dynamics.

The second category, differently, employs a modular design where temporal information is independently processed via deep learning or explicitly crafted temporal modeling techniques, subsequently applying conventional SKGC methods for completion. Methods such as TA-TransE and TA-DistMult [27] utilize recurrent neural networks (RNN) to encode temporal evolution dynamically into entity representations. HyTE [26] maps entity and relation embeddings onto distinct hyperplanes determined by timestamps, thus allowing translation-based operations within a temporally specific embedding space. DE-SimplE [46] introduces diachronic embeddings by modeling entity states distinctly at discrete timestamps, subsequently using the SimplE [24] decoder for effective prediction. The ATiSE [47] model further refines temporal representations by decomposing timestamps into additive components—trend, seasonal, and irregular—capturing intricate evolutionary behaviors and uncertainties via multidimensional Gaussian distributions. TeLM [48] employs multivector embeddings combined with temporal regularizers to enhance generalization and expressivity. BoxTE [49] expands upon BoxE [32] by explicitly embedding temporal information within flexible box-shaped embeddings, and TPBoxE [50] achieves better results by incorporating time probability box embeddings. BDME [51] introduces a fine-grained multigranularity embedding framework. QDN [52] extends the triplet distributor in TDN [40] into a quadruplet distributor and applies fourth-order tensor decomposition to facilitate comprehensive interactions. Zhang et al. [53] propose a joint framework that enhances tensor decomposition by integrating temporal and static modules. TeAST [54] aligns each relation with an Archimedean spiral timeline, introducing a temporal spiral regularizer to maintain temporal order while allowing entity embeddings to remain time-invariant. Several methods, including DyERNIE [55], IME [56], and MADE [57], embed TKGs into multi-curvature spaces to better capture complex structures. SANe [58] introduces the space adaptation network to effectively handle temporal variability. ConvTKG [59] and JointDE [60] employ advanced CNN-based frameworks to capture the complex interactions within TKGs. LGRe [61] and EHPR [62] improve the model's expressive ability by learning granularity representations and evolutionary hierarchy perception representations, respectively.

2.3 Differences with Existing Work

The methods most related to our work are translation-based TKG embedding models, including TTransE [25], HyTE [26], and RTS [29]. Similar to our approach, TTransE learns time embeddings for timestamps and employs relation-timestamp pairs as translation vectors. However, a key distinction lies in its simple assumption that entities, relations, and timestamps all reside within the same semantic space, without modeling explicit transformations between them. In contrast, our model learns embeddings for entities, relations, and timestamps in separate vector spaces, and introduces time-specific or relation-specific projection matrices to bridge these spaces. While prior works such as HyTE and RTS also apply temporally specific hyperplanes or matrices to map entities and relations, they neither learn independent time representations nor incorporate time embeddings as components of translation vectors. These distinctions highlight the fundamental differences between our approach and previous methods.

3 Method

In this section, we introduce two innovative translation-based models for TKGC: TE-TransR and TE-TransT. To detail our approach accurately, we first provide the definition of the problem and related terminologies.

3.1 Problem Definition

Given a TKG $\mathcal{G} = \{(h, r, t, [\tau_s, \tau_e])\}$, we have the entity set $\mathcal{E}$, relation set $\mathcal{R}$, and timestamp set $\mathcal{T}$. Each quadruple $(h, r, t, [\tau_s, \tau_e])$ in $\mathcal{G}$ represents a fact in the real world, where $h, t \in \mathcal{E}$ are the head and tail entities, $r \in \mathcal{R}$ is the relation, and $\tau_s, \tau_e \in \mathcal{T}$ define the boundaries of a time scope during which (h, r, t) is valid. Traditionally, a fact $(h, r, t, [\tau_s, \tau_e])$ is represented as a set of quadruples (h, r, t, τ) which indicates that (h, r, t) is true at a specific timestamp τ that is between τ_s and τ_e. We also adopt the format (h, r, t, τ) to articulate our method in the following. The TKGC task aims to complete TKG quadruples by predicting the missing tail entity in a query $(h, r, ?, \tau)$ or the missing head entity in a query $(?, r, t, \tau)$.

3.2 Models

To fully capture the rich semantics inherent in temporal data, we propose TE-TransR and TE-TransT. As illustrated in Fig. 1, our models learn the embeddings of entities, relations, and timestamps in distinct semantic spaces. Different embedding vectors can be mapped into a unified space through either relation-specific or time-specific matrices. Furthermore, our method models the relation-timestamp pair as the translation, instead of modeling only the relation, highlighting the significance of temporal information.

3.2.1 TE-TransR

We first introduce TE-TransR, an extension of TransR [22]—a model that represents entities and relations in separate spaces and projects entity embeddings into relation-specific spaces before translation. While effective for SKGs, TransR lacks the capacity to model temporal dynamics. TE-TransR addresses this by introducing a time dimension and modeling timestamps in a dedicated semantic space. It further projects both entity and time embeddings into the relation space via relation-specific matrices to enable temporally-aware translation. The underlying assumption of TE-TransR is that both entities and time are relation-sensitive. TransR has demonstrated a strong correlation between entities and relations, and recent TKGC approaches have also identified a close link between time and relations [26,29,41]. TE-TransR integrates the strengths of these methods, inheriting the effectiveness of TransR in handling 1-on-N, N-on-1, and N-on-N relations while introducing a time dimension to leverage the interplay between timestamps and relations.

In TE-TransR, given a quadruple (h, r, t, τ), entity embeddings $\mathbf{h}$ and $\mathbf{t}$ take values in $\mathbb{R}^k$, relation embedding $\mathbf{r}$ takes value in $\mathbb{R}^d$, and time embedding $\boldsymbol{\tau}$ takes value in $\mathbb{R}^n$. Since embeddings are in distinct spaces, the dimensions k, d, and n can differ.

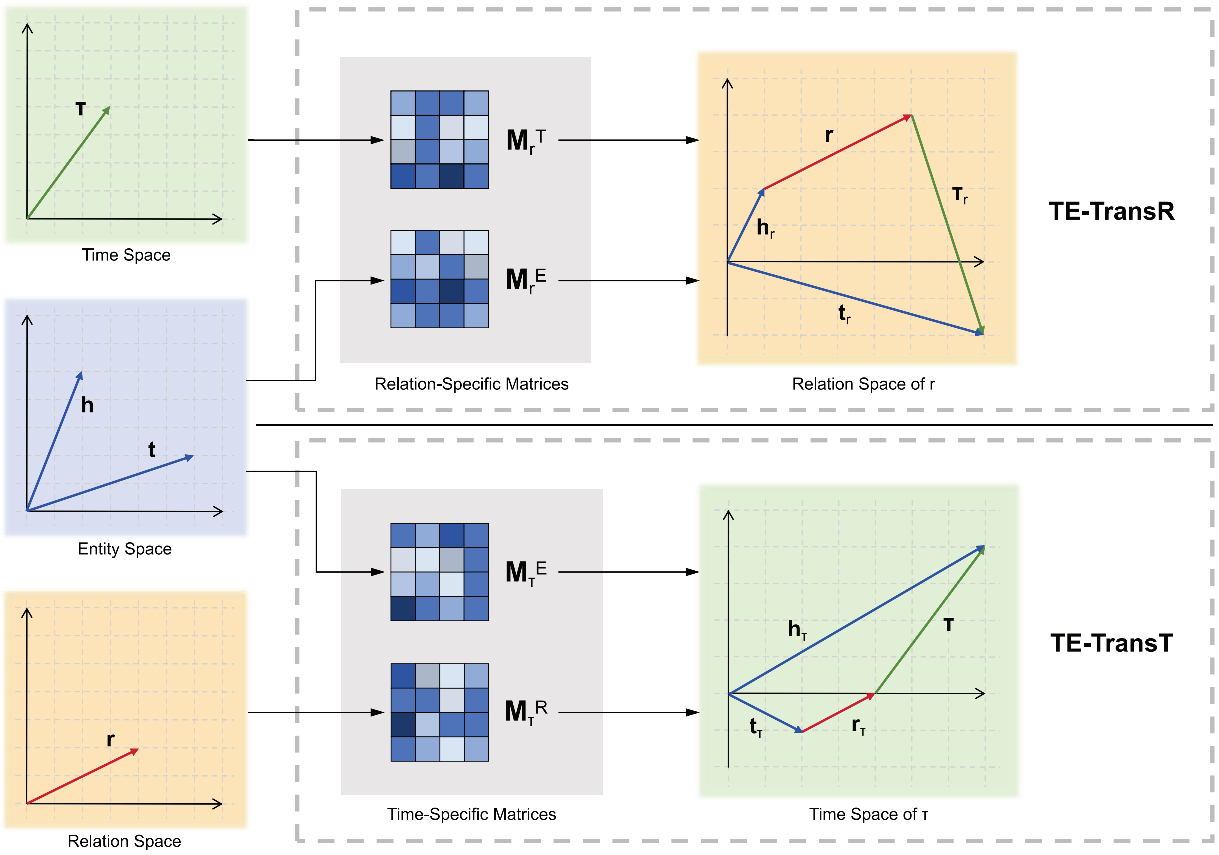


Figure 1: Illustration of TE-TransR and TE-TransT. In both models, we embed entities, relations, and timestamps in distinct semantic spaces. As shown on the top, TE-TransR utilizes relation-specific matrices to project entity and time embeddings to the corresponding relation space. The down panel shows that TE-TransT, differently, maps entity and relation embeddings to the time space with time-specific matrices. The relation-timestamp pair is then employed to build the translation between entities in the unified space

For relation-specific matrices, given a relation r, we have an entity-to-relation projection matrix $\mathbf{M}_r^E \in \mathbb{R}^{k\times d}$ and a time-to-relation projection matrix $\mathbf{M}_r^T \in \mathbb{R}^{n\times d}$. The projected vectors of entities and timestamps are defined as

$$\mathbf{h}_r = \mathbf{h}\mathbf{M}_r^E, \quad \mathbf{t}_r = \mathbf{t}\mathbf{M}_r^E, \quad \boldsymbol{\tau}_r = \boldsymbol{\tau}\mathbf{M}_r^T. \tag{1}$$

The corresponding score function is represented as

$$score(h, r, t, \tau) = \|\mathbf{h}_r + \mathbf{r} + \boldsymbol{\tau}_r - \mathbf{t}_r\|_{l_1/l_2}, \tag{2}$$

where $\|\cdot\|_{l_1/l_2}$ is the l_1 or l_2-norm. Note that the relation embedding $\mathbf{r}$ and the projected time embedding $\boldsymbol{\tau}_r$ together form the translation vector, sharing the same level of importance. In experiments, we impose restrictions on the norms of the embeddings $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$, $\boldsymbol{\tau}$ and the projected vectors as $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{r}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$, $\|\boldsymbol{\tau}\|_2 \leq 1$, $\|\mathbf{h}_r\|_2 \leq 1$, $\|\mathbf{t}_r\|_2 \leq 1$, $\|\boldsymbol{\tau}_r\|_2 \leq 1$.

3.2.2 TE-TransT

TE-TransT also embeds entities, relations, and timestamps in separate semantic spaces, but explores performing translations in time spaces. It utilizes time-specific matrices to map entity embeddings and relation embeddings into the corresponding time space. Different from TE-TransR, TE-TransT assumes that both entities and relations are time-sensitive. Intuitively, the time-specific projection matrices function as temporal adaptation mechanisms, enabling the model to adjust the semantic representations of entities and relations according to the specific timestamp. This design captures the dynamic nature of real-world knowledge, where the meanings and semantics of entities and relations may vary significantly over time. Previous works have attempted to utilize temporal information through time-specific hyperplanes [26] or matrices [29], mapping entities and relations into corresponding hyperplanes or embedding spaces, followed by applying TransE's score function $\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{l_1/l_2}$ to build translation. TE-TransT not only employs time-specific matrices for transformation but also learns the independent representation of timestamps and integrates time embeddings directly into the score function, thereby fully leveraging the rich semantics of the temporal information.

For each timestamp τ, we have an entity-to-time projection matrix $\mathbf{M}_\tau^E \in \mathbb{R}^{k\times n}$ and a relation-to-time projection matrix $\mathbf{M}_\tau^R \in \mathbb{R}^{d\times n}$. Similarly to TE-TransR, the projected vectors of entities and time are defined as

$$\mathbf{h}_\tau = \mathbf{h}\mathbf{M}_\tau^E, \quad \mathbf{t}_\tau = \mathbf{t}\mathbf{M}_\tau^E, \quad \mathbf{r}_\tau = \mathbf{r}\mathbf{M}_\tau^R. \tag{3}$$

The corresponding score function becomes

$$score(h,r,t,\tau) = \|\mathbf{h}_\tau + \mathbf{r}_\tau + \boldsymbol{\tau} - \mathbf{t}_\tau\|_{l_1/l_2}. \tag{4}$$

Both the projected relation embedding $\mathbf{r}_\tau$ and the time embedding $\boldsymbol{\tau}$ act as translation vectors. TE-TransT also enforces constraints on the normalization of embeddings and projected vectors likewise in TE-TransR.

3.2.3 Model Training

We minimize the following margin-based ranking loss to optimize the model's performance on the link prediction task:

$$\mathcal{L} = \sum_{(h,r,t,\tau)\in G^+} \sum_{(h',r,t',\tau)\in G^-} \max(0, score(h,r,t,\tau) - score(h',r,t',\tau) + \gamma), \tag{5}$$

where G^+ represents the set of correct triples, G^- represents the set of corrupted triples, and $\gamma > 0$ denotes a margin hyperparameter.

As in previous works, we utilize two different negative sampling methods for link prediction and time prediction. Here we first introduce the time-agnostic method for the link prediction task, which selects negative samples from the entirety of triples that are absent from the existing KG, without considering timestamps [26]. Specifically, for a given timestamp, we obtain negative samples from G^-, where

$$G^- = \{(h',r,t,\tau) \mid h' \in \mathcal{E}, (h',r,t) \notin G^+\} \cup \{(h,r,t',\tau) \mid t' \in \mathcal{E}, (h,r,t') \notin G^+\}. \tag{6}$$

4 Time Prediction Scheme

Many TKG embedding models [27,43,63–65] treat temporal information only as a supplementary instrument for link prediction and give relatively insufficient attention to time prediction, which is also an important task for TKGC. We identify several limitations with the existing time prediction scheme. To tackle these limitations, here, we introduce a novel time prediction scheme, composed of a time unit-based timestamp setting and a relation-specific evaluation protocol. We also present our time-dependent negative sampling method for the time prediction task.

4.1 Timestamp Setting

A widely adopted timestamp setting is a frequency-based method [26], which handles time in an unintuitive manner. Specifically, it divides time into timestamps, not based on the duration of time, but rather on the occurrence frequency of facts. Each timestamp must contain a number of facts exceeding a certain threshold. As a result, the timestamps are essentially intervals where facts tend to be evenly distributed, but the actual lengths of timestamps vary. This implies that, with such a frequency-based timestamp setting, the predicted timestamp τ in the time prediction task is a predefined imprecise time interval.

We introduce a timestamp setting for time prediction to address the previous vague representation of time, by assigning each timestamp to a precise time unit (e.g., in our experiments, each timestamp τ represents a year on the YAGO11k and Wikidata12k datasets). Based upon our time unit-based timestamp setting, we can precisely and intuitively predict the specific time of a fact, rather than a rough time interval. With a time unit-based methodology, it is crucial to select an appropriate time unit based on the dataset's specific distribution. If the number of facts within each time unit is too small or highly imbalanced across time units, it could negatively impact the training outcomes.

4.2 Evaluation Protocol

We observe a significant correlation between the duration of a fact and its relation type. To improve the validity of time prediction evaluation, we propose linking the evaluation metrics to specific relations. As far as we are aware, it has not been used by previous methods.

4.2.1 Relation-Specific Evaluation Protocol

The process of our relation-specific evaluation protocol is as follows: (1) For each time-missing quadruple $(h, r, t, ?)$, we first calculate the average length l_r of time intervals observed within relation r. (2) Scores are computed for all possible quadruples with different timestamps, which are then ranked in descending order. (3) The top l_r ranked timestamps are adopted to form the prediction set $\mathcal{T}^{pr}$. (4) The Ochiai coefficient is applied to measure the differences between the real set $\mathcal{T}^{re}$ and the prediction set $\mathcal{T}^{pr}$ as

$$O(\mathcal{T}^{re}, \mathcal{T}^{pr}) = \frac{size(\mathcal{T}^{re} \cap \mathcal{T}^{pr})}{\sqrt{size(\mathcal{T}^{re}) \times size(\mathcal{T}^{pr})}}, \tag{7}$$

where $size$ refers to the number of elements in the set, and $\mathcal{T}^{re}$ represents the real time point set while $\mathcal{T}^{pr}$ denotes the predicted time point set. For each quadruple (h, r, t, τ), if τ is a time point, $\mathcal{T}^{re} = \{\tau\}$, and if τ is a time interval, then $\mathcal{T}^{re} = \{\tau_s, \tau_{s+1}, ..., \tau_e\}$. (5) Once we obtain the Ochiai coefficients for all test quadruples, we compute the mean Ochiai coefficient (MOC) as our evaluation metric by averaging the results.

Previous works, relying on frequency-based timestamps, are limited to predicting a vague time interval for a fact, without specific handling of its duration. By leveraging time unit-based timestamps, our new evaluation protocol takes into account the feature of a specific relation type to determine the required number of time points for evaluation and is capable of predicting discrete time points, improving the flexibility and reliability of the evaluation process.

4.2.2 Metric Suitability across Dataset Types

It is important to note that the MOC metric is more suitable for evaluation on interval-based datasets, such as YAGO11k and Wikidata12k, where facts are typically valid over a time interval rather than a single time point. This is because the computation of MOC relies on a relation-specific strategy, selecting top-ranking candidate timestamps based on the average time interval length of a given relation type. In this scenario, MOC effectively captures the extent to which the predicted timestamps match the ground-truth interval. However, in certain datasets such as the event-driven ICEWS14, all facts are only valid at a single timestamp. In this case, both the real set $\mathcal{T}^{re}$ and the prediction set $\mathcal{T}^{pr}$ in the MOC computation always have size one, with $\mathcal{T}^{pr}$ containing only the top-ranked timestamp. As a result, MOC essentially degenerates into the "Hit@1" metric. It merely reflects timestamp prediction accuracy without considering how close the predicted timestamp is to the ground truth. Nonetheless, such proximity clearly matters for the time prediction task: even when the prediction is not exactly correct, a predicted timestamp closer to the true one should be regarded as more reliable. This also highlights a potential direction for improving MOC, specifically by adapting it to ensure robustness under diverse temporal settings.

4.3 Model Training for Time Prediction

For the time prediction task, we propose a time-dependent negative sampling method, which maintains the (h, r, t) triple unchanged, selects another timestamp τ' that has appeared while ensuring that the newly formed quadruple does not belong to the original TKG. The set of time-dependent negative samples can be represented as

$$G^{-} = \{(h, r, t, \tau') \mid \tau' \in \mathcal{T}, (h, r, t, \tau') \notin G^{+}\}. \tag{8}$$

Our proposed time unit-based timestamp setting provides a larger number of timestamps with finer granularity, which can compatibly support the time-dependent negative sampling. With this negative sampling method, the margin-based ranking loss for time prediction is modified to

$$\mathcal{L} = \sum_{(h,r,t,\tau)\in G^{+}} \sum_{(h,r,t,\tau')\in G^{-}} \max(0, score(h, r, t, \tau) - score(h, r, t, \tau') + \gamma). \tag{9}$$

TE-TransR and TE-TransT are both trained for the time prediction task using the training strategy above.

5 Experiments

In this part, we assess the performance of TE-TransR and TE-TransT through various experiments, including link prediction, time prediction, ablation studies, and qualitative analysis. Extensive comparisons and analysis showcase the effectiveness of our method for TKGC.

5.1 Experimental Setup

5.1.1 Datasets

We use three commonly employed benchmark datasets: YAGO11k, Wikidata12k, and ICEWS14 to evaluate our proposed models. The details of these datasets are reported in Table 1. Consistent with prior work, we preserve the original train/valid/test splits for all datasets.

YAGO11k and Wikidata12k are TKG datasets constructed by HyTE [26]. YAGO11k, derived from YAGO3 [66], adheres to the following extraction criteria: (1) inclusion of temporally annotated facts that specify both start and end time points, (2) selection of the ten most frequent temporally rich relations, and

(3) exclusion of edges associated with entities mentioned only once within the subgraph. This process yields a temporal graph comprising 20.5k triples and 10.6k entities. Wikidata12k, sourced from Wikidata [67], follows analogous criteria to those of YAGO11k but expands the selection to the 24 most frequent relations. This results in a temporal graph with 40k triples and 12.5k entities. Both YAGO11k and Wikidata12k feature a temporal granularity of one year. Section 5.4.4 provides a discussion of our strategy for handling missing timestamps.

Table 1: Statistics of datasets.

Datasets	Entity	Relations	Train/Valid/Test	Time span
YAGO11k	10,623	10	16.4k/2k/2k	−453–2844
Wikidata12k	12,554	24	32.5k/4k/4k	1479–2018
ICEWS14	6869	230	72.8k/9k/9k	2014

ICEWS14 [27] is derived from the Integrated Crisis Early Warning System (ICEWS) repository [16], which records structured political and social events. The ICEWS14 subset specifically contains event-based facts occurring within the year 2014. Different from YAGO11k and Wikidata12k, its temporal granularity is one day.

5.1.2 Baselines

To examine the capability of our models, we compare them with translation-based baselines, including SKG embedding models and TKG embedding models.

SKG embedding models: We use four conventional baseline models including TransE [20], TranH [19], TransR [22], and TransD [30]. In these models, TKG quadruples (h, r, t, τ) are treated as (h, r, t) by ignoring time dimension τ.

TKG embedding models: We also compare our models against recent TKG embedding models. We include TTransE [25], HyTE [26], TA-TransE [27], and RTS [29] in this realm.

To more precisely situate our proposed models within the spectrum of TKGC approaches, we have also conducted extensive comparisons with a range of non-translation-based baselines. A brief description of these baselines and the corresponding experimental results can be found in the Appendix A.

5.1.3 Implementation Details and Complexity Analysis

We implement our models using the PyTorch framework and run the models on four NVIDIA RTX A6000 GPUs. We train our models utilizing stochastic gradient descent (SGD) and pick up the optimal hyper-parameters using Optuna [68], referring to the MRR performance on the validation set. We set the number of training epochs N to 2000 and keep the embedding dimensions of entities, relations, and timestamps the same. The embedding dimension d varies among $\{128, 200, 256, 300, 512\}$, and the batch size b is chosen from $\{512, 1024, 2048, 4096\}$. We tune the learning rate lr in the range of $\{0.1, 0.01, 0.001\}$ and the margin η in the range of $\{4, 6, 8, 10\}$. The optimal hyperparameters and configurations are summarized in Table 2, where the parameter *Dimension* denotes the embedding dimension. All reported experimental results are obtained by averaging over five independent runs. In addition, for link prediction and time prediction results, we also include the standard deviation of each evaluation metric across these runs.

Table 2: Summary of optimal hyperparameters and configurations

Datasets	Models	Epochs	Batch size	Dimension	Margin	Learning rate
YAGO11k	TE-TransT	2000	2048	128	10	0.01
	TE-TransR	2000	2048	200	10	0.01
Wikidata12k	TE-TransT	2000	2048	300	10	0.01
	TE-TransR	2000	2048	300	10	0.01
ICEWS14	TE-TransT	2000	2048	128	10	0.01
	TE-TransR	2000	2048	128	10	0.01

Time and space complexity are key indicators for evaluating the practicality and scalability of TKGC models. As shown in Table 3, we compare our proposed models with several representative translation-based methods in terms of these two aspects.

Table 3: Comparison of time and space complexity of our models with existing translation-based models.

Models	Time complexity	Space complexity
TTransE	$O(Nd)$	$O((\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert + \lvert\mathcal{T}\rvert)d)$
HyTE	$O(Nd)$	$O(\lvert\mathcal{T}\rvert_{\text{freq}}d + (\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert)d)$
RTS	$O(Nd^2)$	$O(\lvert\mathcal{T}\rvert_{\text{freq}}d^2 + (\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert)d)$
TE-TransT	$O(Nd^2)$	$O(\lvert\mathcal{T}\rvert d^2 + (\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert + \lvert\mathcal{T}\rvert)d)$
TE-TransR	$O(Nd^2)$	$O(\lvert\mathcal{R}\rvert d^2 + (\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert + \lvert\mathcal{T}\rvert)d)$

In terms of time complexity, TTransE does not involve any transformation operations and only performs the score function computation, i.e., building the translation, which is an $O(d)$ operation. Under an N-epoch training setup, this results in a total complexity of $O(Nd)$. HyTE, in addition to computing the score function, projects embeddings onto time-specific hyperplanes. This orthogonal projection is also an $O(d)$ operation, so its overall complexity remains $O(Nd)$. When transformation is performed using projection matrices, as in RTS and our models, the time complexity increases to $O(Nd^2)$.

As for space complexity, we present a more detailed comparison. TTransE only stores embeddings, resulting in a complexity of $O(d)$. HyTE does not learn explicit time embeddings but requires storing normal vectors for hyperplane projection, which is also $O(d)$ per timestamp. In contrast, RTS stores time-specific matrices, introducing an $O(d^2)$ term in the complexity. Both HyTE and RTS adopt a frequency-based timestamp setting, where the number of timestamps (denoted $|\mathcal{T}|_{\text{freq}}$) is smaller than the full timestamp set $|\mathcal{T}|$. In our models, we explicitly learn independent representations for timestamps, leading to a $O(|\mathcal{T}|d)$ space term. As with other matrix-based methods, an $O(d^2)$ space is required. For TE-TransT, this term is scaled by $|\mathcal{T}|$, while for TE-TransR, it is scaled by $|\mathcal{R}|$.

Overall, compared to existing projection-matrix-based models, the asymptotic complexity of our methods remains unchanged, with differences only in constant factors. Nonetheless, it should be noted that our time-unit-based timestamp setting can introduce additional computational cost, and the actual memory usage of time-specific and relation-specific matrices may vary depending on dataset characteristics, which should be taken into account in practical applications. In our experiments, TE-TransT occupies approximately 10 GB of memory and converges in 4.4 h/1.9 h/0.3 h on YAGO11k, Wikidata12k, and ICEWS14, respectively. In contrast, TE-TransR uses about 8 GB of memory and converges in 3.8 h/1.2 h/0.2 h on these datasets. This difference may be attributed to the fact that, in these datasets, the number of relations is significantly smaller than the number of timestamps, implying a smaller constant factor in the $O(d^2)$ term, since $|\mathcal{R}| < |\mathcal{T}|$.

5.2 Link Prediction

5.2.1 Evaluation Protocol

Link prediction aims to identify the missing head or tail entity in a given fact. Link prediction has been widely explored since being proposed in [69] and has become a critical measurement for KG completion techniques.

In this task, we utilize the method of Eq. (6) to generate negative samples for training. For every test quadruple (h, r, t, τ), we substitute the head or tail entity with every potential entity from $\mathcal{E}$ to construct corrupted triples. Rather than using the static filtered setting [20], we adopt a time-aware filtered setting proposed by [70] which only filters out the duplicate triples that occur at τ from the corrupted set. This ensures the corrupted triples are not coarsely filtered out without considering temporal information. We calculate scores for all the quadruples, including the test quadruple and corrupted quadruples, and then rank them. For a test fact that is valid during a time interval, we split the time interval into time points, e.g., $(h, r, t, [\tau_s, \tau_e])$ is split to $\{(h, r, t, \tau_s), (h, r, t, \tau_{s+1}), \ldots, (h, r, t, \tau_e)\}$. The score of this test fact is the average score of all split quadruples.

We employ two standard evaluation metrics to measure the prediction results: MRR (mean reciprocal rank) and Hit@k, where $k \in \{1, 3, 10\}$. We follow the setting described in [46]. Concretely, for a test quadruple $(h, r, t, \tau) \in test$, two queries 1-$(?, r, t, \tau)$, 2-$(h, r, ?, \tau)$ are created. All the entities in $h \cup \mathcal{C}_h$ for query 1 and entities in $t \cup \mathcal{C}_t$ for query 2 are ranked, where $\mathcal{C}_h = \{h' : h' \in \mathcal{E}, (h', r, t, \tau) \notin \mathcal{G}\}$, $\mathcal{C}_t = \{t' : t' \in \mathcal{E}, (h, r, t', \tau) \notin \mathcal{G}\}$. Let k_h, k_t denote the rank positions of h and t with respect to the two queries. We calculate MRR as $\frac{1}{2*|test|} \sum_{f=(h,r,t,\tau)\in test} (\frac{1}{k_h} + \frac{1}{k_t})$, and Hits@$k$ as $\frac{1}{2*|test|} \sum_{f=(h,r,t,\tau)\in test} (I_{k_h \leq k} + I_{k_t \leq k})$, where $I_{conditon}$ is 1 when $condition$ holds and 0 otherwise.

5.2.2 Evaluation Results

As reported in Table 4, TE-TransT and TE-TransR outperform baseline models across all metrics on the Wikidata12k dataset and ICEWS14 dataset. Although our models fail to beat all baselines in terms of the Hit@10 metric on the YAGO11k dataset, they tie with the second best-performing model, RTS, and surpass all baselines in terms of other metrics. The results suggest that our approach is capable of effectively embedding temporal information and utilizing it for link prediction. In comparisons within our models, TE-TransT slightly outperforms TE-TransR on the YAGO11k dataset, while TE-TransR slightly edges out TE-TransT on the Wikidata12k dataset and ICEWS14 dataset.

Table 4: The results of link prediction.

Models	**YAGO11k**				**Wikidata12k**				**ICEWS14**			
	MRR	**Hit@1**	**Hit@3**	**Hit@10**	**MRR**	**Hit@1**	**Hit@3**	**Hit@10**	**MRR**	**Hit@1**	**Hit@3**	**Hit@10**
TransE	0.100	0.015	0.138	0.244	0.178	0.100	0.192	0.339	0.280	0.094	–	0.637
TransH†	0.099	0.010	0.142	0.240	0.178	0.097	0.195	0.353	0.278	0.095	0.399	0.633
TransR†	0.112	0.028	0.155	0.235	0.177	0.095	0.191	0.358	0.291	0.106	0.408	0.627
TransD†	0.096	0.014	0.134	0.233	0.175	0.093	0.191	0.355	0.282	0.099	0.401	0.641

(Continued)

Table 4 (continued)

Models	YAGO11k				Wikidata12k				ICEWS14			
	MRR	**Hit@1**	**Hit@3**	**Hit@10**	**MRR**	**Hit@1**	**Hit@3**	**Hit@10**	**MRR**	**Hit@1**	**Hit@3**	**Hit@10**
TTransE	0.108	0.020	0.150	0.251	0.172	0.096	0.184	0.329	0.255	0.074	–	0.601
HyTE	0.105	0.015	0.143	0.272	0.180	0.098	0.197	0.333	0.297	0.108	0.416	0.655
TA-TransE	0.127	0.027	0.160	**0.326**	0.178	0.030	0.267	0.429	0.275	0.095	–	0.625
RTS‡	–	–	–	0.304	–	–	–	0.448	–	–	–	–
TE-TransT	**0.142**	**0.043**	**0.181**	0.304	0.262	0.162	0.285	0.472	0.349	0.148	0.473	0.657
std	±0.001	±0.001	±0.001	±0.002	±0.001	±0.000	±0.001	±0.000	±0.001	±0.001	±0.001	±0.001
TE-TransR	0.136	0.037	0.178	0.304	**0.264**	**0.164**	**0.286**	**0.477**	**0.361**	**0.155**	**0.487**	**0.672**
std	±0.002	±0.001	±0.003	±0.002	±0.002	±0.002	±0.003	±0.002	±0.001	±0.001	±0.001	±0.001

Note: The top-performing results are denoted in bold, and the second-best results are indicated with underlining. †: We obtain results by utilizing publicly accessible codes shared by authors. ‡: We take results reported in the original paper. Other results are taken from [64]. - means the unavailable result.

5.3 *Time Prediction*

5.3.1 *Evaluation Protocol*

Time prediction involves predicting the missing time point or time interval of a fact, which is also an important task for TKGC. In the time prediction task, the incorporation of time embeddings in our method allows for an explicit and direct replacement of the time dimension during negative sampling as in Eq. (8). The remainder of the training process aligns with the procedures of the link prediction task. We utilize the identical training set and the same best parameters that are employed in link prediction for model training. Following the evaluation protocol introduced in Section 4, we apply relation-specific evaluation to all time-missing triples $(h, r, t, ?)$ in the test set and obtain the final MOC.

TTransE [25] and HyTE [26] serve as baselines for the time prediction task. Since static KG embedding methods do not consider the time dimension and other TKG embedding methods are not tailored for time prediction, they are not suitable for this task. TTransE incorporates timestamps within the same embedding space as entities and relations, thus allowing direct compatibility with our setup for the time prediction task. Regarding HyTE, to ensure fairness, we adapt its original timestamp setting based on the frequency of facts to our time unit-based setting and then produce the results.

5.3.2 *Evaluation Results*

Evaluation results of time prediction are shown in Table 5. The models exhibit close performance on the YAGO11k dataset, with TE-TransT and TE-TransR marginally outperforming TTransE and HyTE. On the Wikidata12k dataset, HyTE performs significantly worse than other models, highlighting the critical flaw of not learning an independent representation for timestamps. TE-TransT's performance falls below both TTransE and TE-TransR, indicating that representing each timestamp in a different embedding space could be detrimental to the time prediction task in certain settings. This might be due to the sparsity of training data for individual timestamps. TE-TransR outperforms TTransE, suggesting that building translation in the relation space effectively facilitates information interaction among different embeddings. As for the ICEWS14 dataset, all methods yield relatively low MOC scores. This is because MOC is better suited for datasets like YAGO11k and Wikidata12k, where facts are valid over a time interval. In such cases, MOC effectively measures the overlap between the predicted timestamps and the ground-truth interval. However, in datasets like ICEWS14, where each fact is only valid at a single timestamp, correct predictions of the exact timestamp are required to achieve a high MOC score, which is much more difficult. Despite the overall low values, our models still outperform TTransE and HyTE, further demonstrating the generalizability of our approach across different types of TKGs.

Table 5: The results of time prediction.

Models	**YAGO11k**	**Wikidata12k**	**ICEWS14**
	MOC	**MOC**	**MOC**
TTransE	0.289	<u>0.127</u>	0.004
HyTE	0.273	0.050	0.002
TE-TransT	<u>0.304</u>	0.102	**0.007**
std	±0.001	±0.000	±0.000
TE-TransR	**0.306**	**0.142**	<u>0.005</u>
std	±0.001	±0.001	±0.000

Note: The top-performing results are denoted in bold, and the second-best results are indicated with underlining. All results are produced with our proposed time prediction scheme.

5.4 Analysis

5.4.1 Temporal Sensitivity of Relations

Previous works have made significant progress in demonstrating that relations are often time-sensitive. To verify that our model has captured the close correlation between relations and timestamps, we use TE-TransT as an example and present t-SNE visualizations in Fig. 2, showing the relation embeddings of some randomly selected relations projected into different time spaces. As observed, the embeddings of the same relation exhibit a clear and continuous ordering over time, indicating that the model has learned the temporal sensitivity of relations. Additionally, it is evident that the influence of time varies across different relations.

To move beyond qualitative plots, we quantify how well time-specific transformations preserve relation identity while allowing timestamp-specific variation. For each relation r and timestamp τ, we obtain its time-specific embedding $\mathbf{r}_\tau$ via the learned transformation and pool all $\{\mathbf{r}_\tau : r \in \mathcal{R},\ \tau \in \mathcal{T}\}$. We then run k-means on this pooled set with $k = |\mathcal{R}|$, so that each cluster is intended to group the embeddings of one relation across different timestamps.

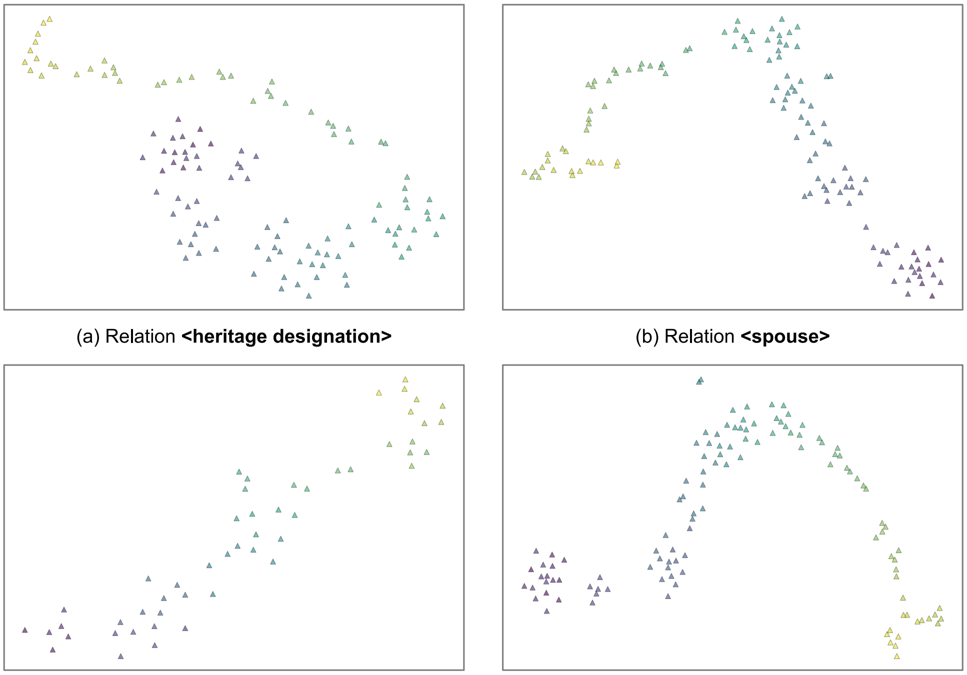
For any cluster C, we define its coherence as the average pairwise cosine similarity among its members:

$$\text{Coherence}(C) = \frac{2}{|C|(|C|-1)} \sum_{i<j,\, \mathbf{x}_i, \mathbf{x}_j \in C} \cos(\mathbf{x}_i, \mathbf{x}_j), \tag{10}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity. The **Average Cluster Coherence** is then defined as:

$$\text{Average Cluster Coherence} = \frac{1}{k} \sum_{j=1}^{k} \text{Coherence}(C_j). \tag{11}$$

Based on the value reported in Table 6, the relatively high Average Cluster Coherence indicates that, after time-specific transformations, embeddings of the same relation remain tightly grouped while avoiding collapse to a single point—consistent with the t-SNE trajectories—since embeddings within each cluster still carry timestamp-specific characteristics.



(a) Relation **<heritage designation>**

(b) Relation **<spouse>**

(c) Relation **<IMA status and/or rank>**

(d) Relation **<capital of>**

Figure 2: t-SNE visualization of relation embeddings. Scatter points in each subplot represent the embeddings of the same relation with different timestamps. Warmer and brighter colors indicate later times. Other relations exhibit similar results

Table 6: Quantitative analysis of relation and time embeddings.

Model	**Average cluster coherence (Relation embeddings)**	**Temporal correlation (Time embeddings)**
TE-TransT	0.74	0.86

5.4.2 Evolution of Time Embeddings

Figure 3 illustrates the 2D PCA projection of the 128-dimensional time embeddings from the TE-TransT model, trained for the time prediction. We can observe that, even without the addition of any time-related constraints, the time embeddings naturally exhibit discernible patterns. Similar time points form clusters, and an ordering is observed as time progresses, indicating that our independent representation for timestamps successfully captures the rich semantics of temporal information through time-dependent negative sampling.

To complement the qualitative observations from Fig. 3, we further introduce a **Temporal Correlation** metric to quantitatively assess whether the learned time embeddings preserve the chronological ordering of timestamps. Let $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{|\mathcal{T}|}\}$ denote the embeddings of timestamps sorted by their true temporal order. For each timestamp i, we compute its one-dimensional coordinate p_i obtained by projecting $\mathbf{t}_i$ onto the first principal component, which captures the dominant variance in the temporal embedding space. We then calculate the Spearman's rank correlation coefficient between $\{p_i\}$ and their corresponding ground-truth temporal indices $\{i\}$:

$$\text{Temporal Correlation} = \frac{\operatorname{cov}(\operatorname{rank}(p), \operatorname{rank}(i))}{\sigma_{\operatorname{rank}(p)}\, \sigma_{\operatorname{rank}(i)}}, \tag{12}$$

where $\operatorname{cov}(\cdot, \cdot)$ denotes the covariance, and σ is the standard deviation. As shown in Table 6, the TE-TransT model achieves a high Temporal Correlation, indicating a strong positive alignment and quantitatively confirming that the learned time embeddings preserve temporal continuity in the embedding space.

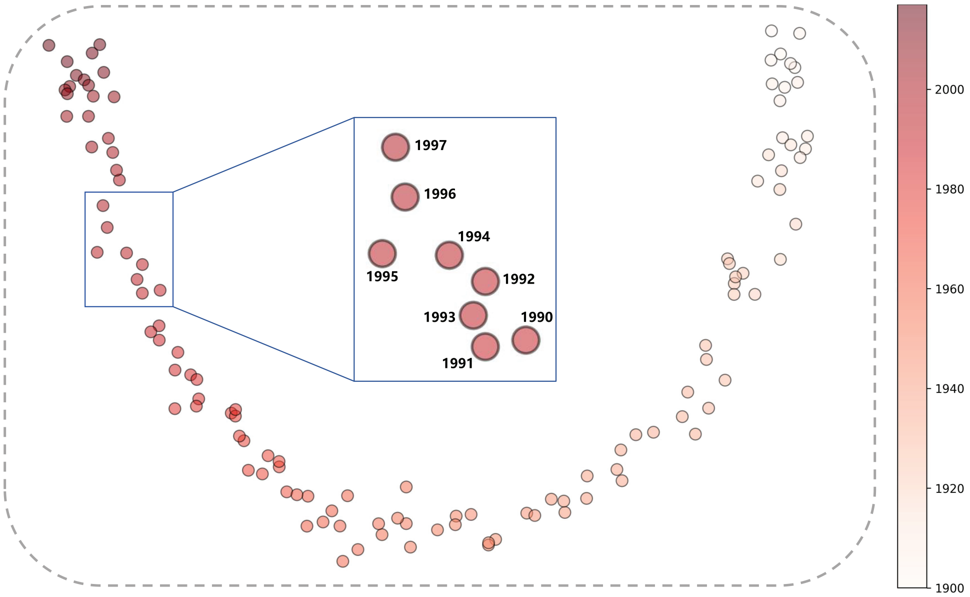


Figure 3: 2D PCA visualization of time embeddings. Colors become darker as time progresses. A zoomed-in region is marked by a blue box, providing a closer look at the clustering pattern in greater detail. Only timestamps later than 1900 are retained for better visual presentation

5.4.3 Ablation Study

To investigate the individual contributions of each component of our proposed method to the final outcomes, we conduct a series of ablation studies. Link prediction results on the YAGO11k dataset for the ablation study are reported in Table 7. As demonstrated in Table 7, two existing methods, TransR [22] and TTransE [25], are included for comparisons. TTransE can be seen as our models stripped of all embedding transformations between different spaces, while TransR represents TE-TransR without considering the time dimension. Additionally, as the ablation methods of TE-TransR, "TE-TransR w/o RE" denotes TE-TransR with the relation embedding removed from the score function, "TE-TransR w/o TRT" indicates TE-TransR without the transformation from time space to relation space, and "TE-TransR w/o RST" represents TE-TransR without relation-specific transformations, meaning that the projection operations are performed using shared matrices across all relations. Similarly, ablation methods of TE-TransT include "TE-TransT w/o TE", which denotes TE-TransT with the time embedding removed from the score function, "TE-TransT w/o RTT", which represents TE-TransT without the transformation from relation space to time space, and "TE-TransT w/o TST", which indicates TE-TransT without time-specific transformations by using shared projection matrices for all timestamps.

Table 7: The ablation study results on the YAGO11k dataset.

Models	YAGO11k			
	MRR	**Hit@1**	**Hit@3**	**Hit@10**
TransR	0.112	0.028	0.155	0.235
TTransE	0.108	0.020	0.150	0.251
TE-TransR w/o RE	0.127	0.031	0.166	0.276
TE-TransR w/o TRT	0.134	0.035	0.173	0.289
TE-TransR w/o RST	0.130	0.032	0.169	0.281
TE-TransR	0.136	0.037	0.178	**0.304**
TE-TransT w/o TE	0.131	0.033	0.171	0.280
TE-TransT w/o RTT	0.132	0.034	0.173	0.283
TE-TransT w/o TST	0.129	0.032	0.168	0.279
TE-TransT	**0.142**	**0.043**	**0.181**	**0.304**

Note: The top-performing results are denoted in bold, and the second-best results are indicated with underlining.

Firstly, the proposed models outperform the baseline models, TransR and TTransE. This is because TransR lacks the leverage of temporal information, while TTransE crudely represents entities, relations, and timestamps all within a common space, thereby limiting the model's expressive capability. Secondly, our models perform better than "TE-TransR w/o RE" and "TE-TransT w/o TE". These two ablation methods share a commonality: the removal of either relation embedding or time embedding from the score function, which means keeping only one of them as the translation vector. This simplification leads to diminished model performance, thus demonstrating the effectiveness of modeling the relation-timestamp pair as the translation. Thirdly, TE-TransR and TE-TransT also surpass "TE-TransR w/o TRT" and "TE-TransT w/o RTT", both of which eliminate the transformation between time space and relation space. The results indicate that this alteration leads to poorer outcomes, suggesting that representing timestamps and relations in the same space leads to the loss of semantics. This further substantiates the notion that relations evolve over time. Finally, our models outperform both "TE-TransR w/o RST" and "TE-TransT w/o TST", highlighting the importance of relation-specific and time-specific transformations. These results demonstrate that such tailored transformations could significantly enhance the expressiveness of the models.

5.4.4 Impact of Timestamp Settings

We also design experiments to explore the impact of different timestamp settings on the link prediction task. The corresponding results are summarized in Figs. 4 and 5.

1. "HyTE" utilizes the frequency-based timestamp setting of HyTE [26], where timestamps are essentially variable-length time intervals. Missing values are replaced with the earliest or latest time across the entire dataset.
2. "TU-based #1" adopts the time unit-based timestamp setting proposed in this paper, where timestamps correspond to the dataset's inherent time units. Missing values are also replaced with the earliest or latest time across the dataset.
3. "TU-based #2" employs the proposed time unit-based timestamp setting but uses a different strategy for replacing missing values, substituting them with the earliest or latest time associated with the specific relation.

4. “TU-based #3” also uses the proposed time unit-based timestamp setting, where missing values are replaced with the fact’s start or end time that exists. If both start and end times are missing, that fact is discarded.

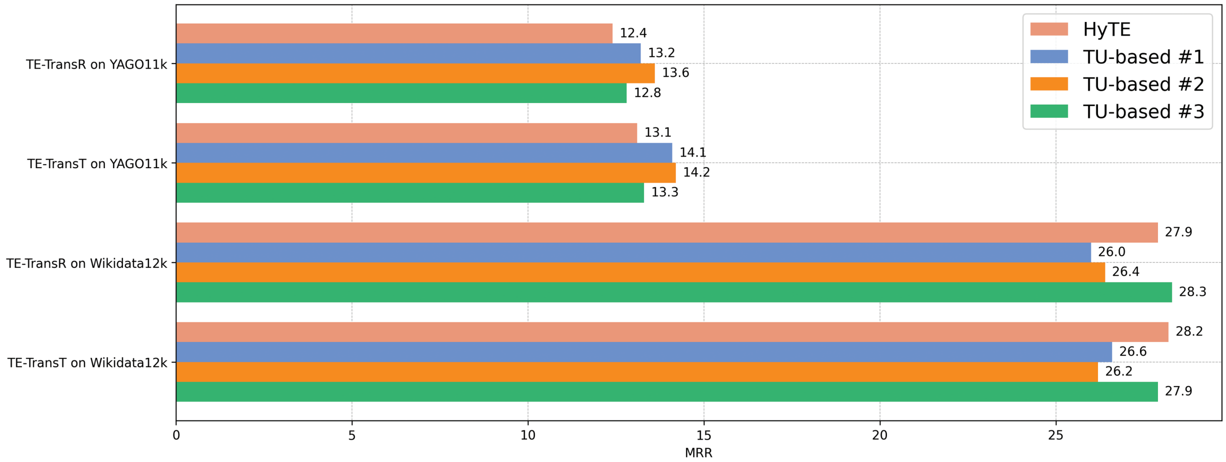


Figure 4: Impact of different timestamp settings on MRR

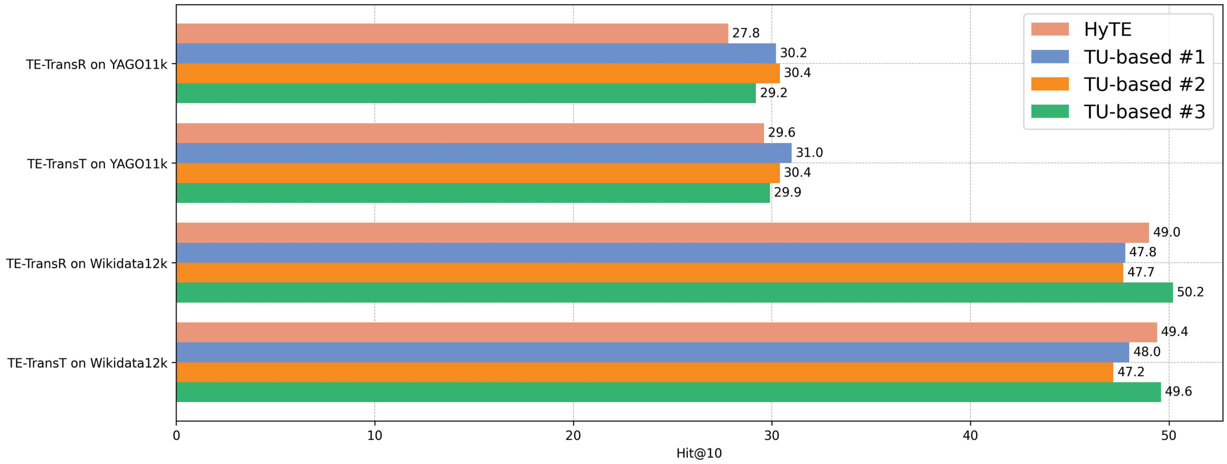


Figure 5: Impact of different timestamp settings on Hit@10. The results of Hit@1 and Hit@3 show similar patterns

We can observe that different timestamp settings have a discernible influence on the evaluation results, with a consistent effect on both MRR and Hit@10. The results obtained from “TU-based #1” and “TU-based #2” are relatively higher on the YAGO11k dataset, while lower on the Wikidata12k dataset. In contrast, “HyTE” and “TU-based #3” show the opposite trend. The results of link prediction and time prediction presented in this paper are all based on “TU-based #2”. These observations suggest that while the proposed time unit-based timestamp setting enables more fine-grained predictions in the time prediction task, it is not necessarily optimal for link prediction. Due to the inherent differences in dataset distributions, the time unit-based setting may suffer from data sparsity in certain cases. This is likely a contributing factor to the superior performance of the frequency-based setting on the Wikidata12k dataset.

Time granularity has a significant impact on the data sparsity of TKGs. Coarser granularity can help alleviate data sparsity issues; however, it may also limit the model’s ability to capture fine-grained temporal patterns, thereby reducing its expressiveness. In Table 8, we report the sensitivity of TE-TransT to different levels of time granularity, including daily, weekly, and monthly settings, on the ICEWS14 dataset. We observe that increasing the granularity from one day to one week leads to improvements across all metrics. However, when the granularity is further increased to one month, the performance drops sharply and falls below that of the daily setting. These results indicate a trade-off between mitigating data sparsity and preserving temporal expressiveness, which should be carefully considered based on the characteristics of each dataset.

Table 8: Sensitivity to time granularity.

Models	**ICEWS14**			
	MRR	**Hit@1**	**Hit@3**	**Hit@10**
TE-TransT (daily)	<u>0.349</u>	<u>0.148</u>	<u>0.473</u>	<u>0.657</u>
TE-TransT (weekly)	**0.357**	**0.152**	**0.482**	**0.667**
TE-TransT (monthly)	0.338	0.144	0.454	0.642

Note: The top-performing results are denoted in bold, and the second-best results are indicated with underlining.

5.4.5 Discussion of Dataset-Induced Variations

Due to the varying data distributions and temporal characteristics across datasets, the performance improvements of our method are not entirely consistent. In the link prediction task, our proposed models generally achieve solid gains over existing translation-based models across nearly all datasets and metrics. However, the imbalance in performance becomes more pronounced in the time prediction task. For example, TE-TransR achieves a relatively modest improvement of 5.9% over the strongest baseline on YAGO11k, whereas the improvement rises to 11.8% on Wikidata12k. TE-TransT exhibits even greater fluctuations: while its performance on YAGO11k and Wikidata12k falls short of TE-TransR—and on Wikidata12k it even underperforms a baseline model—it achieves the best result on ICEWS14. We attribute this discrepancy to the influence of data distribution and sparsity. In our approach, TE-TransR and TE-TransT employ relation-specific and time-specific transformations, whose projection matrices depend heavily on whether sufficient data are available for the corresponding relation or timestamp. In YAGO11k and Wikidata12k, the temporal span is wide but the distribution is uneven, with facts concentrated toward later timestamps. As a result, the time-specific matrices for certain timestamps may not be adequately trained, leading to unstable predictions. By contrast, ICEWS14 covers only a single year, where data are relatively evenly distributed across days, enabling TE-TransT to learn more effectively. Regarding sparsity, we have already discussed in Section 5.4.4 the importance of selecting an appropriate time granularity to avoid poor training for underrepresented timestamps. In principle, TE-TransR may also suffer from similar issues, since different relation types are not equally represented, though the effect is less pronounced than in TE-TransT. Because our method does not explicitly address data imbalance, the performance can inevitably be affected by the inherent distribution of the datasets.

6 Limitations and Future Work

Our proposed method enhances model expressiveness by embedding temporal information into dedicated semantic spaces and employing time-specific and relation-specific projection matrices to facilitate transformations across different spaces. While such fine-grained modeling improves temporal reasoning, it introduces additional runtime and memory costs, which may affect the scalability of our models. We leave the exploration of more cost-efficient alternatives for future work. For the time prediction task, we introduce the MOC metric, which is particularly suitable for datasets where facts remain valid over a time interval, as it effectively captures the overlap between predicted timestamps and ground-truth intervals. However, for datasets where each fact is only valid at a single timestamp, MOC has certain limitations. Future work may involve designing more unified and robust evaluation metrics that are applicable across different temporal settings. Moreover, when learning time embeddings and time-specific matrices for the link prediction task, our method requires careful calibration of time granularity to balance the trade-off between mitigating data sparsity and preserving temporal expressiveness. Adopting the finest possible granularity may not always yield optimal results, and choosing an appropriate granularity should depend on the characteristics of the dataset.

While translation-based models offer the advantages of structural simplicity, fewer parameters, and high inference efficiency, neural-based approaches—such as those built on GNNs, RNNs, or Transformers—are typically more expressive due to their ability to model complex temporal interactions, capture long-term dependencies, and attend to critical contextual signals. In future work, we aim to explore hybrid architectures that combine the strengths of translation-based modeling with attention mechanisms.

Currently, our work is confined to incorporating independent time embeddings for TKGC tasks. Moving forward, we intend to extend this method to additional challenges, including event graph prediction.

7 Conclusion

We have presented two novel translation-based TKG embedding models: TE-TransR and TE-TransT. As the key to our method, we have explored learning embeddings of timestamps in different spaces from entities and relations, and modeling the relation-timestamp pair as the translation. Concretely, we embed entities, relations, and timestamps in multiple distinct embedding spaces, respectively. Different types of embeddings are projected into a common space through either relation-specific or time-specific matrices. We utilize the relation-timestamp pair as translation vectors, improving the flexibility and expressive capability of models. Furthermore, we propose a novel time prediction scheme, comprising a time unit-based timestamp setting and a relation-specific evaluation protocol, allowing for more precise and reliable time prediction. Extensive quantitative and qualitative evaluations show that our models, particularly TE-TransR, surpass previous translation-based approaches, offering a more accurate method for TKGC.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Natural Science Foundation of China under Grant No. 72293575.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jinglu Chen and Daniel Dajun Zeng; methodology, Jinglu Chen, Mengpan Chen and Wenhao Zhang; software, Mengpan Chen and Huihui Ren; validation, Mengpan Chen, Wenhao Zhang and Huihui Ren; formal analysis, Jinglu Chen and Wenhao Zhang; investigation, Jinglu Chen, Mengpan Chen and Wenhao Zhang; resources, Jinglu Chen and Daniel Dajun Zeng; data curation, Mengpan Chen and Huihui Ren; writing—original draft preparation, Jinglu Chen, Mengpan Chen and Wenhao Zhang; writing—review and editing, Jinglu Chen, Wenhao Zhang and Daniel Dajun Zeng; visualization, Wenhao Zhang; supervision, Jinglu Chen and Daniel Dajun Zeng; project administration, Jinglu Chen and Daniel Dajun Zeng; funding acquisition, Daniel Dajun Zeng. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available on the website: https://github.com/cmp11/tkgc_trans (accessed on 17 September 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A Comparison of Link Prediction Results with Non-Translation-Based Methods

Appendix A1 Baselines

To better position our proposed models within the broader TKGC landscape, we also conduct comparisons with non-translation-based baselines. These baselines include both SKG embedding models and TKG embedding models.

SKG embedding models: Our SKG embedding baselines include DistMult [21], QuatE [71], and HousE [72], encompassing bilinear, quaternion-based, and reflection-based modeling paradigms.

TKG embedding models: For TKG embedding baselines, we consider TA-DistMult [27], DE-SimplE [46], ATiSE [47], DyERNIE [55], HERCULES [73], and DuaTHPR [74], covering a diverse range of temporal modeling strategies from RNN-based models to Transformer-based architectures.

Appendix A2 Evaluation Results

The experimental results reported in Table A1 reveal mixed outcomes. For YAGO11k, our models achieve superior performance over all non-translation-based baselines on Hit@3 and Hit@10, but fall behind neural-based models ATiSE and DuaTHPR on MRR and Hit@1. On Wikidata12k, our models consistently perform slightly below ATiSE yet slightly above DuaTHPR across all metrics. However, on ICEWS14, the baselines generally achieve better performance than our models. This discrepancy may be related to the substantially higher relation diversity in ICEWS14, which contains 230 relation types—23 times more than YAGO11k and 9.6 times more than Wikidata12k—thereby making temporal modeling more challenging. These findings align with our discussion in Section 6, where we acknowledge that neural-based models, through their ability to capture complex temporal interactions, tend to achieve advantageous performance in such highly heterogeneous environments.

Table A1: Comparison of link prediction results with non-translation-based models

Models	YAGO11k				Wikidata12k				ICEWS14			
	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
DistMult	0.158	0.107	0.161	0.268	0.222	0.119	0.238	0.460	0.439	0.323	–	0.672
QuatE	0.164	0.107	0.148	0.270	0.230	0.125	0.243	0.416	0.471	0.353	0.530	0.712
HousE	0.158	0.089	0.124	0.264	0.269	0.147	0.271	0.416	0.427	0.367	0.537	**0.769**
TA-DistMult	0.161	0.103	0.171	0.292	0.218	0.122	0.232	0.447	0.477	0.363	–	0.686
DE-SimplE	–	–	–	–	–	–	–	–	0.526	0.418	0.592	0.725
ATiSE	**0.170**	0.110	0.171	0.288	**0.280**	**0.175**	**0.317**	**0.481**	0.550	0.436	0.629	0.750
DyERNIE	–	–	–	–	–	–	–	–	0.588	0.498	0.638	0.761
HERCULES	–	–	–	–	–	–	–	–	**0.612**	**0.543**	**0.647**	0.741
DuaTHPR	0.169	**0.115**	0.152	0.294	0.246	0.141	0.262	0.453	0.495	0.412	0.547	0.729
TE-TransT	0.142	0.043	**0.181**	**0.304**	0.262	0.162	0.285	0.472	0.349	0.148	0.473	0.657
TE-TransR	0.136	0.037	0.178	**0.304**	0.264	0.164	0.286	0.477	0.361	0.155	0.487	0.672

Note: The top-performing results are denoted in bold, and the second-best results are indicated with underlining. Results of baselines are taken from [74]. - means the unavailable result.

References

1. Miller GA. WordNet: a lexical database for English. Commun ACM. 1995;38(11):39–41.
2. 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; 2008 Jun 10–12; Vancouver, BC, Canada. p. 1247–50.

3. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: a nucleus for a web of open data. In: International Semantic Web Conference. Cham, Switzerland: Springer; 2007. p. 722–35.
4. Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka E, Mitchell T. Toward an architecture for never-ending language learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2010 Jul 11–15; Atlanta, GA, USA. p. 1306–13.
5. Suchanek FM, Kasneci G, Weikum G. YAGO: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web; 2007 May 8–12; Banff, AB, Canada. p. 697–706.
6. Yates A, Banko M, Broadhead M, Cafarella MJ, Etzioni O, Soderland S. TextRunner: open information extraction on the web. In: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT). Rochester, NY, USA: Association for Computational Linguistics; 2007. p. 25–6.
7. Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. ERNIE: enhanced language representation with informative entities. arXiv:1905.07129. 2019.
8. Wang H, Zhang F, Wang J, Zhao M, Li W, Xie X, et al. RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; 2018 Oct 22–26; Turin, Italy. p. 417–26.
9. Gao M, Li JY, Chen CH, Li Y, Zhang J, Zhan ZH. Enhanced multi-task learning and knowledge graph-based recommender system. IEEE Trans Knowl Data Eng. 2023;35(10):10281–94. doi:10.1109/tkde.2023.3251897.
10. Bao J, Duan N, Yan Z, Zhou M, Zhao T. Constraint-based question answering with knowledge graph. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: The COLING 2016 Organizing Committee; 2016. p. 2503–14.
11. Huang X, Zhang J, Li D, Li P. Knowledge graph embedding based question answering. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining; 2019 Feb 11–15; Melbourne, VIC, Australia. p. 105–13.
12. Whan Kim Y, Kim JH. A model of knowledge based information retrieval with hierarchical concept graph. J Doc. 1990;46(2):113–36. doi:10.1108/eb026857.
13. Bounhas I, Soudani N, Slimani Y. Building a morpho-semantic knowledge graph for Arabic information retrieval. Inf Process Manag. 2020;57(6):102124.
14. Liu Y, Wan Y, He L, Peng H, Philip SY. KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence; 2021 Feb 2–9; Online. p. 6418–25.
15. Wan G, Pan S, Gong C, Zhou C, Haffari G. Reasoning like human: hierarchical reinforcement learning for knowledge graph reasoning. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20); 2021 Jan 7–15; Online.
16. Boschee E, Lautenschlager J, O'Brien S, Shellman S, Starz J, Ward M. ICEWS coded event data. Harv Dataverse. 2015:V37.
17. Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Commun ACM. 2014;57(10):78–85.
18. Leetaru K, Schrodt PA. GDELT: global data on events, location, and tone, 1979–2012. In: ISA annual convention. Vol. 2. Easton, PA, USA: Citeseer; 2013. p. 1–49.
19. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2014 Jul 27–31; Québec City, QC, Canada. Vol. 28. p. 1112–9.
20. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst. 2013;26:2787–95.
21. Yang B, Wt Yih, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. arXiv:1412.6575. 2014.
22. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2015 Jan 25–30; Austin, TX, USA. Vol. 29. p. 2181–7.

23. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. In: International Conference on Machine Learning. Westminster, UK: PMLR; 2016. p. 2071–80.
24. Kazemi SM, Poole D. Simple embedding for link prediction in knowledge graphs. Adv Neural Inf Process Syst. 2018;31:4289–4300.
25. Leblay J, Chekol MW. Deriving validity time in knowledge graph. In: WWW '18: Companion Proceedings of the the Web Conference 2018; 2018 Apr 23–27; Lyon, France. p. 1771–6.
26. Dasgupta SS, Ray SN, Talukdar P. HYTE: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018 Oct 31–Nov 4; Brussels, Belgium. p. 2001–11.
27. García-Durán A, Dumančić S, Niepert M. Learning sequence encoders for temporal knowledge graph completion. arXiv:1809.03202. 2018.
28. Wu J, Cao M, Cheung JCK, Hamilton WL. Temp: temporal message passing for temporal knowledge graph completion. arXiv:2010.03526. 2020.
29. Wang Z, Li L, Zeng DD. Time-aware representation learning of knowledge graphs. In: 2021 International Joint Conference on Neural Networks (IJCNN); 2021 Jul 18–22; Shenzhen, China. p. 1–8.
30. 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. Vol. 1: Long Papers. Stroudsburg, PA, USA: Association for Computational Linguistics; 2015. p. 687–96.
31. Sun Z, Deng ZH, Nie JY, Tang J. Rotate: knowledge graph embedding by relational rotation in complex space. arXiv:1902.10197. 2019.
32. Abboud R, Ceylan I, Lukasiewicz T, Salvatori T. Boxe: a box embedding model for knowledge base completion. Advances in Neural Information Processing Systems. 2020;33:9649–61.
33. Balažević I, Allen C, Hospedales TM. Tucker: tensor factorization for knowledge graph completion. arXiv:1901.09590. 2019.
34. Nguyen DQ, Vu T, Nguyen TD, Nguyen DQ, Phung D. A capsule network-based embedding model for knowledge graph completion and search personalization. arXiv:1808.04122. 2018.
35. Wang J, Wang B, Gao J, Hu Y, Yin B. Multi-concept representation learning for knowledge graph completion. ACM Trans Knowl Discov Data. 2023;17(1):1–19. doi:10.1145/3533017.
36. Wang J, Wang B, Gao J, Hu S, Hu Y, Yin B. Multi-level interaction based knowledge graph completion. IEEE/ACM Trans Audio Speech Lang Process. 2023;32:386–96. doi:10.1109/taslp.2023.3331121.
37. Dettmers T, Minervini P, Stenetorp P, Riedel S. Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2018 Feb 2–7; New Orleans, LA, USA. Vol. 32. p. 1811–8.
38. Schlichtkrull M, Kipf TN, Bloem P, Van Den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018; 2018 Jun 3–7; Crete, Greece. 2018. p. 593–607.
39. Zhang Z, Li Z, Liu H, Xiong NN. Multi-scale dynamic convolutional network for knowledge graph embedding. IEEE Trans Knowl Data Eng. 2020;34(5):2335–47. doi:10.1109/tkde.2020.3005952.
40. Wang J, Wang B, Gao J, Li X, Hu Y, Yin B. TDN: triplet distributor network for knowledge graph completion. IEEE Trans Knowl Data Eng. 2023;35(12):13002–14. doi:10.1109/tkde.2023.3272568.
41. Jiang T, Liu T, Ge T, Sha L, Li S, Chang B, et al. Encoding temporal information for time-aware link prediction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics; 2016. p. 2350–4.
42. Xu C, Nayyeri M, Alkhoury F, Yazdi HS, Lehmann J. TeRo: a time-aware knowledge graph embedding via temporal rotation. arXiv:2010.01029. 2020.
43. Sadeghian A, Armandpour M, Colas A, Wang DZ. ChronoR: rotation based temporal knowledge graph embedding. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence; 2021 Feb 2–9; Online. Vol. 35. p. 6471–9.

44. Lacroix T, Obozinski G, Usunier N. Tensor decompositions for temporal knowledge base completion. arXiv:2004.04926. 2020.
45. Shao P, Zhang D, Yang G, Tao J, Che F, Liu T. Tucker decomposition-based temporal knowledge graph completion. Knowl Based Syst. 2022;238:107841. doi:10.1016/j.knosys.2021.107841.
46. Goel R, Kazemi SM, Brubaker M, Poupart P. Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence; 2020 Feb 7–1; New York, NY, USA. Vol. 35. p. 3988–95.
47. Xu C, Nayyeri M, Alkhoury F, Yazdi H, Lehmann J. Temporal knowledge graph completion based on time series Gaussian embedding. In: The Semantic Web–ISWC 2020: 19th International Semantic Web Conference; 2020 Nov 2–6; Athens, Greece. Cham, Switzerland: Springer; 2020. p. 654–71.
48. Xu C, Chen YY, 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. Stroudsburg, PA, USA: Association for Computational Linguistics; 2021. p. 2569–78.
49. Messner J, Abboud R, Ceylan II. Temporal knowledge graph completion using box embeddings. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence; 2022 Feb 22–Mar 1; Online. Vol. 36. p. 7779–87.
50. Li S, Wang Q, Li Z, Zhang L. TPBoxE: temporal knowledge graph completion based on time probability box embedding. Comput Sci Inf Syst. 2025;22(1):153–80.
51. Yue L, Ren Y, Zeng Y, Zhang J, Zeng K, Wan J. Block decomposition with multi-granularity embedding for temporal knowledge graph completion. In: International Conference on Database Systems for Advanced Applications. Cham, Switzerland: Springer; 2023. p. 706–15.
52. Wang J, Wang B, Gao J, Li X, Hu Y, Yin B. QDN: a quadruplet distributor network for temporal knowledge graph completion. IEEE Trans Neural Netw Learn Syst. 2024;35(10):14018–30. doi:10.1109/tnnls.2023.3274230.
53. Zhang F, Chen H, Shi Y, Cheng J, Lin J. Joint framework for tensor decomposition-based temporal knowledge graph completion. Inf Sci. 2024;654(2):119853. doi:10.1016/j.ins.2023.119853.
54. Li J, Su X, Gao G. TeAST: temporal knowledge graph embedding via Archimedean spiral timeline. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. Vol. 1: Long Papers. p. 15460–74.
55. Han Z, Ma Y, Chen P, Tresp V. DyERNIE: dynamic evolution of Riemannian manifold embeddings for temporal knowledge graph completion. arXiv:2011.03984. 2020.
56. Wang J, Cui Z, Wang B, Pan S, Gao J, Yin B, et al. Integrating multi-curvature shared and specific embedding for temporal knowledge graph completion. In: Proceedings of the ACM on Web Conference 2024; 2024 May 13–17; Singapore. p. 1954–62.
57. Wang J, Wang B, Gao J, Pan S, Liu T, Yin B, et al. MADE: multicurvature adaptive embedding for temporal knowledge graph completion. IEEE Trans Cybern. 2024;54(10):5818–31. doi:10.1109/tcyb.2024.3392957.
58. Li Y, Zhang X, Zhang B, Huang F, Chen X, Lu M, et al. SANe: space adaptation network for temporal knowledge graph completion. Inf Sci. 2024;667:120430. doi:10.1016/j.ins.2024.120430.
59. He M, Zhu L, Bai L. ConvTKG: a query-aware convolutional neural network-based embedding model for temporal knowledge graph completion. Neurocomputing. 2024;588(5):127680. doi:10.1016/j.neucom.2024.127680.
60. He M, Zhu L, Bai L. Jointly leveraging 1D and 2D convolution on diachronic entity embedding for temporal knowledge graph completion. Appl Soft Comput. 2025;176(2):113144. doi:10.1016/j.asoc.2025.113144.
61. Zhang J, Wan T, Mu C, Lu G, Tian L. Learning granularity representation for temporal knowledge graph completion. In: International Conference on Neural Information Processing. Cham, Switzerland: Springer; 2024. p. 91–105.
62. Guo J, Zhao M, Yu J, Yu R, Song J, Wang Q, et al. EHPR: learning evolutionary hierarchy perception representation based on quaternion for temporal knowledge graph completion. Inf Sci. 2025;688:121409. doi:10.1016/j.ins.2024.121409.

63. Li Z, Jin X, Li W, Guan S, Guo J, Shen H, et al. Temporal knowledge graph reasoning based on evolutional representation learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; Online. p. 408–17.
64. Xie Z, Zhu R, Liu J, Zhou G, Huang JX. TARGAT: a time-aware relational graph attention model for temporal knowledge graph embedding. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2023;31:2246–58. doi:10.1109/taslp.2023.3282101.
65. Yang F, Bai J, Li L, Zeng D. A continual learning framework for event prediction with temporal knowledge graphs. In: 2023 IEEE International Conference on Intelligence and Security Informatics (ISI); 2023 Oct 2–3; Charlotte, NC, USA. p. 1–6.
66. Mahdisoltani F, Biega J, Suchanek FM. YAGO3: a knowledge base from multilingual Wikipedias. In: 7th Biennial Conference on Innovative Data Systems Research (CIDR 2015); 2015 Jan 4–7; Asilomar, CA, USA. 2013.
67. Erxleben F, Günther M, Krötzsch M, Mendez J, Vrandečić D. Introducing Wikidata to the linked data web. In: The Semantic Web–ISWC 2014: 13th International Semantic Web Conference; 2014 Oct 19–23, Riva del Garda, Italy. Cham, Switzerland: Springer; 2014. p. 50–65.
68. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019 Aug 4–8; Anchorage, AK, USA. p. 2623–31.
69. Bordes A, Weston J, Collobert R, Bengio Y. Learning structured embeddings of knowledge bases. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence; 2011 Aug 7–11; San Francisco, CA, USA. Vol. 25. p. 301–6.
70. Jain P, Rathi S, Mausam, Chakrabarti S. Temporal knowledge base completion: new algorithms and evaluation protocols. arXiv:2005.05035. 2020.
71. Zhang S, Tay Y, Yao L, Liu Q. Quaternion knowledge graph embeddings. Adv Neural Inf Process Syst. 2019;32:2735–45.
72. Li R, Zhao J, Li C, He D, Wang Y, Liu Y, et al. House: knowledge graph embedding with householder parameterization. In: International Conference on Machine Learning. Westminster, UK: PMLR; 2022. p. 13209–24.
73. Montella S, Barahona LMR, Heinecke J. Hyperbolic temporal knowledge graph embeddings with relational and time curvatures. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Stroudsburg, PA, USA: Association for Computational Linguistics; 2021. p. 3296–308.
74. Chen Y, Li X, Liu Y, Hu T. Integrating transformer architecture and householder transformations for enhanced temporal knowledge graph embedding in DuaTHP. Symmetry. 2025;17(2):173. doi:10.3390/sym17020173.