Along the Time: Timeline-traced Embedding for Temporal Knowledge Graph Completion

Fuwei Zhang[†]
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
zhangfuwei20g@ict.ac.cn

Zhao Zhang[*]
Institute of Computing Technology, Chinese Academy of Sciences
zhangzhao2021@ict.ac.cn

Xiang Ao[†]
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
aoxiang@ict.ac.cn

Fuzhen Zhuang
Institute of Artificial Intelligence, Beihang University
SKLSDE, School of Computer Science, Beihang University
zhuangfuzhen@buaa.edu.cn

Yongjun Xu
Institute of Computing Technology, Chinese Academy of Sciences
xyj@ict.ac.cn

Qing He[*†]
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
heqing@ict.ac.cn

ABSTRACT

Recent years have witnessed remarkable progress on knowledge graph embedding (KGE) methods to learn the representations of entities and relations in static knowledge graphs (SKGs). However, knowledge changes over time. In order to represent the facts happening in a specific time, temporal knowledge graph (TKG) embedding approaches are put forward. While most existing models ignore the independence of semantic and temporal information. We empirically find that current models have difficulty distinguishing representations of the same entity or relation at different timestamps. In this regard, we propose a TimeLine-Traced Knowledge Graph Embedding method (TLT-KGE) for temporal knowledge graph completion. TLT-KGE aims to embed the entities and relations with timestamps as a complex vector or a quaternion vector. Specifically, TLT-KGE models semantic information and temporal information as different axes of complex number space or quaternion space. Meanwhile, two specific components carving the relationship between semantic and temporal information are devised to buoy the modeling. In this way, the proposed method can not only distinguish the independence of the semantic and temporal information, but also establish a connection between them. Experimental results on the link prediction task demonstrate that TLT-KGE achieves substantial improvements over state-of-the-art competitors. The source code will be available on https://github.com/zhangfw123/TLT-KGE.

[*]corresponding author
[†]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS). Xiang Ao is also at Institute of Intelligent Computing Technology, Suzhou, China.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557233

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**.

KEYWORDS

Temporal Knowledge Graph; Knowledge Representation Learning; Link Prediction

ACM Reference Format:
Fuwei Zhang, Zhao Zhang, Xiang Ao, Fuzhen Zhuang, Yongjun Xu, and Qing He. 2022. Along the Time: Timeline-traced Embedding for Temporal Knowledge Graph Completion. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557233

1 INTRODUCTION

A lot of large-scale knowledge graphs (KGs), including DBPedia [2], Freebase [4], and etc., have been widely used in many downstream tasks, e.g., recommender systems [7, 15, 35], information retrieval [24, 32, 36] and question answering [38] due to their expressive power and structured knowledge. Generally, a fact in static KGs is represented as a triplet (s, p, o), where s is subject, p is predicate and o is object. Subject and object are composed of entities (e.g., *People:Barack Obama, City:New York, Company:Microsoft, Movie:Titanic* etc.), while the predicate is made up of relations (e.g., *capitalOf, belongTo, presidentOf*, etc.). For instance, (*Paris, capitalOf, France*) means *Paris* is the capital of *France*. *Paris* and *France* are two entities, *capitalOf* is the relation between them.

However, some events are valid only in certain moments or in a range of time. For example, the fact (*Barack Obama, presidentOf, USA*) is true from 2008 to 2016 (a range of time), and the fact (*Barack Obama, Make a visit, South Korea*) is valid on April 18, 2014 (a certain day). The inclusion of time information makes the facts in KGs more specific and accurate. To meet this demand, *temporal knowledge graphs* (TKGs), including ICEWS [6], GDELT [21] and Wikidata [30], are proposed for representing the temporarily valid

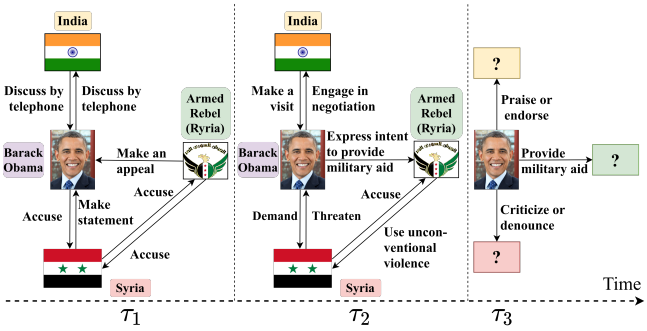


Figure 1: An example of link prediction in a temporal knowledge graph extracted from ICEWS14. The dotted black line with a right arrow at the bottom denotes the timeline ($\tau_1 < \tau_2 < \tau_3$). The solid black lines represent the relationships between two entities (e.g., Barack Obama, India and etc).

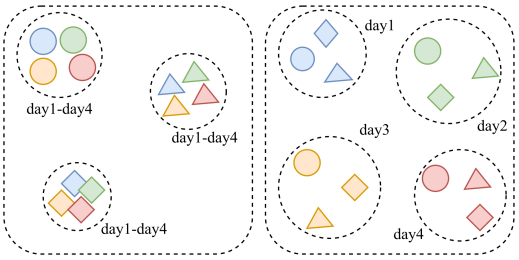


Figure 2: Left: The distribution of entity (relation) embeddings in baseline models when evolving over time. Right: The distribution of entity (relation) embeddings in TLT-KGE when evolving over time. Different colors denote different timestamps. The circle, triangle and diamond represent different entities (relations).

events. Specifically, triples in TKGs are annotated with their corresponding timestamps at the same time. It thus derives a quadruple of a time-sensitive fact in TKGs as (s, p, o, τ), where τ is the time annotation, e.g., 2022-01-01.

Despite there exist plenty of large-scale SKGs and TKGs[1], they remain remarkably incomplete. Link prediction, one of the subtasks in knowledge graph completion, is a significant task that predicts missing entities for incomplete queries in the form of $(s, p, ?, \tau)$ or $(?, p, o, \tau)$ in TKGs. Figure 1 gives an example for the link prediction task in a TKG extracted from ICEWS14. In this example, the link prediction task is to predict the object of (*Barack Obama, Provide military aid / Praise or endorse / Criticize or denounce, ?,* τ_3).

In SKGs, various knowledge graph embedding (KGE) models encode the entities and relations in a low-dimensional space and use some translation-based functions [5, 23, 27, 31], semantic matching functions [28, 34] or even graph neural networks [39] to learn the representations of entities and relations. Following these technologies, some methods have been proposed to tackle the problem of temporal knowledge graph completion [10, 13, 18, 20]. These temporal KGE models also embed the time information into low-dimensional vectors and are shown to perform better than static ones. However, most existing temporal KGE models simply concatenate or multiply the semantic and temporal information together, neglecting the independence of the two kinds of information. Under current settings, the time information can be viewed as temporal disturbances for the representations of entities or relations, as shown in the left part of Figure 2. Therefore, current models may have difficulty distinguishing representations of the same entity or relation at different timestamps and hardly model their changes over time.

To this end, we propose a TimeLine-Traced Knowledge Graph Embedding (TLT-KGE) model for temporal knowledge graph completion, which aims to embed the temporal entities (relations) with a semantic part related to entity (relation) information and a temporal part related to time information. Specifically, TLT-KGE embeds the entity and relation with timestamp as a complex vector or a quaternion vector. And the semantic information and temporal information are modeled as different axes of the complex number space or quaternion space. For instance, in the complex number space, the real part represents semantics, and the imaginary part represents time. The right part of Figure 2 shows an illustrative example of the embedding distribution of the proposed model TLT-KGE. TLT-KGE is able to well distinguish not only different entities (relations) in the same timestamp, but also the same entity (relation) in different timestamps. TLT-KGE fully considers the independence and connection between semantic information and time information, which is beneficial to achieve better performance.

Besides, TLT-KGE is equipped with two extra components that can improve the representations of timestamps for TKGs: a Shared Time Window (STW) module and a Relation-Timestamp Composition (RTC) module. The shared time window module employs a shared embedding for adjacent timestamps to embolden the connection of events in a range of time. To enhance the expression of a relation at specific time, the relation-timestamp composition module further polishes the embeddings for relations with timestamp.

In a summary, we highlight our contributions as follows.

- We propose a novel TimeLine-Traced Knowledge Graph Embedding (TLT-KGE) method for temporal knowledge graph completion, which aims to embed the temporal entities and relations considering both the independence and the connection of the semantic and temporal information.
- TLT-KGE is equipped with two extra components: a Shared Time Window (STW) module and a Relation-Timestamp Composition (RTC) module to strengthen the representations of entities and relations along the time.
- We verify the effectiveness of the proposed TLT-KGE on three widely used benchmarks, and the results exhibit our model can achieve substantial improvements over state-of-the-art baselines.

2 RELATED WORK

2.1 Static Knowledge Graph Embedding

Recently, various approaches have been put forward for knowledge graph completion in static KGs. The majority of these approaches embed entities and relations to low-dimensional vectors. These proposed methods can be divided into two groups: (i) translation-based models and (ii) semantic matching models.

[1]We use SKG and TKG to static knowledge graph and temporal knowledge graph, respectively, for description convenience hereafter.

Translation-based models assume the relation embedding as a translation or rotation from the subject to the object. One of the representative works of translation-based methods is TransE [5]. Because of the weakness to learning N-N, 1-N and N-1 relations, some models extend TransE to tackle this problem, such as TransH [31] and TransR [23]. To further improve the capacity of representation learning, RotatE [27] defines a rotation operation parameterized by the relation from subject to object.

Semantic matching models compute the plausibility of a given fact via matching the semantic information of entities and relations. DistMult [34] uses a bilinear function to calculate the score. ComplEx [28] and QuatE [37] extend DistMult to complex space and quaternion elliptic space, respectively. DualE [8] extends QuatE to dual vector space. With the development of deep neural networks, ConvE [11] and ConvKB [26] utilize convolution neural networks to model the sophisticated interaction between entities and relations. Tucker [3] is a tucker-decomposition-based method for KGE.

2.2 Temporal Knowledge Graph Embedding

In the past several years, some works aim to deal with the problem of link prediction in TKGs. Most of them extend conventional KGE models by considering time information as embeddings.

One of the major differences among these models is how to efficiently model the embeddings of timestamps. By using a long short-term memory (LSTM) network to extract the time series features, TA-TransE [13] and TA-DistMult [13] incorporate the embeddings of time information to TransE and DistMult, respectively. HyTE [10] extends TransH [31] by projecting the entities and relations to a time-specific hyperplane. DE-SimplE [14] introduces a diachronic part for entities and relations. Inspired by RotatE [27], TeRo [33] utilizes a rotation operation on both subject and object to evaluate the semantic scores of a given fact. T(NT)ComplEx [18] incorporates the time embedding into a three-order tensor and uses a novel regularization method to improve link prediction performance. RotateQVS [9] aims to model the temporal changes with rotation operation in quaternion vector space and uses a score function that is similar to TransE. BoxTE [25] is a box embedding model for temporal knowledge graph completion, which is based on the static KGE model BoxE [1].

Illuminated by the expressive power of graph neural network (GNN), some temporal KGE approaches utilize different variants of GNN. RE-NET [17] and RE-GCN [22] learn the evolutional representations of entities and relations at each timestamp by modeling the KG sequence recurrently.

In our work, we regard the time information as a timeline and utilize a novel method that embeds the entities and relations at specific time with a semantic part and a temporal part, keeping the independence of the semantic and temporal information.

3 PRELIMINARIES ON HAMILTON'S QUATERNIONS

In this section, we will introduce the definition of Hamilton's quaternions and some operations used in the methodology section. Hamilton's Quaternions [16] is a representative hypercomplex number that extends the traditional complex number to four-dimension space. And quaternions have been used in static KGE [8, 37].

A quaternion q is consist of one real component and three imaginary components i, j, k, which is defines as $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, where a, b, c, d are real numbers and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are imaginary units. It can also be viewed as a quadruple (a, b, c, d). Here we will provide some operations for quaternions:

- **Conjugate**. The conjugate of a quaternion is similar to that of a complex number. The real component keeps the same while three imaginary components are opposite to the original quaternion:
$$\bar{q} = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}. \tag{1}$$
- **Norm**. The norm of a quaternion is defined as:
$$\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{2}$$
- **Inner Product**. The quaternion inter product between $q_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$ and $q_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$ is calculated by taking the sum of products of each corresponding components:
$$q_1 \cdot q_2 = \langle a_1, a_2 \rangle + \langle b_1, b_2 \rangle + \langle c_1, c_2 \rangle + \langle d_1, d_2 \rangle. \tag{3}$$
- **Hamilton Product**. The hamilton product of two quaternions q_1 and q_2 is composed of the multiplications of every factor and follows the distribution law in quaternion, which is defined as:
$$\begin{aligned} q_1 \otimes q_2 = &(a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) \\ &+ (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)\mathbf{i} \\ &+ (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)\mathbf{j} \\ &+ (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)\mathbf{k}. \end{aligned} \tag{4}$$

4 METHODOLOGY

In this section, we first introduce the overview of our model, including the task of knowledge graph completions and some notations. Then the detailed descriptions of TLT-KGE will be given. Finally, we will introduce regularization terms and loss functions used in our training process.

4.1 Overview

A TKG is viewed as a graph $\mathcal{G} = \{(s, p, o, \tau)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ are the set of entities, relations and timestamps, respectively. Given a graph $\mathcal{G}$ and a quadruple with the missing term $(s, p, ?, \tau)$, we can infer the missing entities by finding the answers from the entity set $\mathcal{E}$. For example, given two events (*India, Make a visit, Barack Obama, 2014/09/30*) and (*India, Engage in negotiation, Barack Obama, 2014/09/30*), if we want to predict the most possible missing entity of a query (*Barack Obama, Praise or endorse, ?, 2014/10/01*) from the set of entities {*Syria, Armed Rebel (Syria), India*} based on the given facts, we can easily infer the most possible answer is *India* because the existing facts indicate the interaction between *India* and *Barack Obama*. This is an simple example of knowledge graph completion.

The proposed method TLT-KGE has a crucial point: we divide the embedding of entities or relations into two independent parts. One is the semantic meaning of entities or relations (semantic part), and the other is the timestamps (temporal part). Under this setting, we can regard the time embeddings as a timeline acted on the entities or relations. Then we incorporate the semantic part and the temporal part to a complex vector or a quaternion vector. In order to strengthen the expression of timestamps and relations, TLT-KGE is equipped with two additional components: (i) Shared

time window module, which utilizes shared embeddings for time windows to strengthen the connection of events that happen within a time scope; (ii) Relation-timestamp composition module, which is able to enhance the expression of a relation at specific time.

4.2 Proposed Method: TLT-KGE

4.2.1 Timeline-traced Embedding. The timeline-traced embedding consists of two individual components: the semantic part and the temporal part. The semantic part represents the meaning of entities or relations and it can be regarded as the static part of timeline-traced embedding. And temporal part is designed for expressing the time information. For a given TKG $\mathcal{G} = \{(s, p, o, \tau)\} \subseteq \mathcal{E}\times\mathcal{R}\times\mathcal{E}\times\mathcal{T}$, the semantic embedding matrix of all entities and relations is denoted as $\mathbf{W}_E \in \mathbb{R}^{|\mathcal{E}|\times d}$ and $\mathbf{W}_R \in \mathbb{R}^{|\mathcal{R}|\times d}$, where $|\mathcal{E}|$ and $|\mathcal{R}|$ is the size of entity set $\mathcal{E}$ and relation set $\mathcal{R}$, and the dimension of embeddings is d. Each row of $\mathbf{W}_E$ or $\mathbf{W}_R$ is the semantic embedding of an entity or relation in $\mathcal{G}$. For the sake of providing comprehensive representations of entities and relations in current timestamp. We design temporal embeddings for entities and relations, respectively. The temporal embedding matrix of all timestamps is viewed as $\mathbf{W}_T^e \in \mathbb{R}^{|\mathcal{T}|\times d}$ and $\mathbf{W}_T^r \in \mathbb{R}^{|\mathcal{T}|\times d}$, where $|\mathcal{T}|$ is the size of timestamp set $\mathcal{T}$ and each row of $\mathbf{W}_T^e$ or $\mathbf{W}_T^r$ is one of the timestamp embeddings representing an exact time.

Semantic Part. For a given fact $(s, r, o, \tau) \in \mathcal{G}$ in temporal knowledge graph, the semantic embeddings of entity s and o is defined as $\mathbf{e}_s \in \mathbb{R}^d$ and $\mathbf{e}_o \in \mathbb{R}^d$. The semantic embeddings of relation p is defined as $\mathbf{r}_p \in \mathbb{R}^d$. Note that $\mathbf{e}_s, \mathbf{e}_o$ and $\mathbf{r}_p$ are rows in matrix $\mathbf{W}_E$, $\mathbf{W}_E$ and $\mathbf{W}_R$, respectively.

Temporal Part. The temporal embeddings of timestamp τ for entities and relations is defined as $\mathbf{t}_\tau^e \in \mathbb{R}^d$ and $\mathbf{t}_\tau^r \in \mathbb{R}^d$, respectively. And $\mathbf{t}_\tau^e$, $\mathbf{t}_\tau^r$ are rows in matrix $\mathbf{W}_T^e$ and $\mathbf{W}_T^r$, respectively.

4.2.2 Combine to a Uniform Embedding. In the proposed model, we aim to combine the semantic and temporal information to represent entities and relations in KG. Inspired by the powerful ability of the complex vector space and hypercomplex vector space, we design two ways to combine the semantic and temporal embeddings together. One is combining them in complex vector space, and the other is combining them in quaternion vector space.

Complex Vector Space. The first way is to regard the entire embedding as a complex vector. The semantic embedding is the real part and the temporal embedding is the imaginary part of the complex vector. Both of them make up a full vector in complex vector space, and the entire embedding of entities s, o and relations p at the current timestamp $\tau \in \{1, 2, \ldots, |\mathcal{T}|\}$ can be represented as Equation 5,

$$\mathbf{e}_{s,\tau}^c = \mathbf{e}_s + \mathbf{t}_\tau^e \mathbf{i}, \quad \mathbf{e}_{o,\tau}^c = \mathbf{e}_o + \mathbf{t}_\tau^e \mathbf{i}, \quad \mathbf{r}_{p,\tau}^c = \mathbf{r}_p + \mathbf{t}_\tau^r \mathbf{i}, \tag{5}$$

where $\mathbf{e}_{s,\tau}^c$, $\mathbf{e}_{o,\tau}^c \in \mathbb{C}^d$ and $\mathbf{r}_{p,\tau}^c \in \mathbb{C}^d$ denote the embeddings of subject, object and predicate at current timestamp τ. $\mathbb{C}$ denotes the algebra of complex number. i is the complex unit. The former embeddings (i.e., $\mathbf{e}_s$, $\mathbf{e}_o$ and $\mathbf{r}_p$)) are the real part and the latter embeddings (i.e., $\mathbf{t}_\tau^e$, $\mathbf{t}_\tau^e$ and $\mathbf{t}_\tau^r$) are the imaginary part.

Quaternion Vector Space. The powerful representation ability of the hypercomplex number system has been proved in several works before [8, 37]. Thus, we extend the complex vector to the quaternion vector. In order to do so, we first divide both the semantic embedding and the temporal embedding into two parts. The embeddings $\mathbf{e}_s, \mathbf{e}_o, \mathbf{r}_p, \mathbf{t}_\tau^e, \mathbf{t}_\tau^r$ are equally divided into $(\mathbf{e}_{s,a}, \mathbf{e}_{s,b})$, $(\mathbf{e}_{o,a}, \mathbf{e}_{o,b})$, $(\mathbf{r}_{p,a}, \mathbf{r}_{p,b})$, $(\mathbf{t}_{\tau,c}^e, \mathbf{t}_{\tau,d}^e)$, $(\mathbf{t}_{\tau,c}^r, \mathbf{t}_{\tau,d}^r)$, respectively. Note that the embedding size of divided embeddings is $\frac{d}{2}$. Then, the processed embeddings are combined as follow,

$$\begin{aligned}
\mathbf{e}_{s,\tau}^q &= \mathbf{e}_{s,a} + \mathbf{e}_{s,b}\mathbf{i} + \mathbf{t}_{\tau,c}^e\mathbf{j} + \mathbf{t}_{\tau,d}^e\mathbf{k} \\
\mathbf{e}_{o,\tau}^q &= \mathbf{e}_{o,a} + \mathbf{e}_{o,b}\mathbf{i} + \mathbf{t}_{\tau,c}^e\mathbf{j} + \mathbf{t}_{\tau,d}^e\mathbf{k} \\
\mathbf{r}_{p,\tau}^q &= \mathbf{r}_{p,a} + \mathbf{r}_{p,b}\mathbf{i} + \mathbf{t}_{\tau,c}^r\mathbf{j} + \mathbf{t}_{\tau,d}^r\mathbf{k},
\end{aligned} \tag{6}$$

where $\mathbf{e}_{s,\tau}^q$, $\mathbf{e}_{o,\tau}^q \in \mathbb{H}^{\frac{d}{2}}$ and $\mathbf{r}_{p,\tau}^q \in \mathbb{H}^{\frac{d}{2}}$ denote the embeddings of subject, object and predicate in current timestamp τ. $\mathbb{H}$ denotes the algebra of quaternions. $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the unit vectors representing the three Cartesian axes. The first embedding in the right-side of Equation 6 is the real part and the other three embeddings are the imaginary parts in quaternion number space.

Figure 3 shows an illustration of two method that combines the semantic part and temporal part. For the TLT-KGE (Complex), the entities and relations are along the real axis, and timestamps are along the imaginary axis. For the TLT-KGE (Quaternion), we simplify the illustration via setting the imaginary part $\mathbf{k}$ of time information to zero so that we can describe our method in a 3D figure. The X-axis represents the real part of the quaternion, i.e., semantic information. The Y-axis and Z-axis represent the imaginary parts $\mathbf{i}$ and $\mathbf{j}$, i.e., temporal information.

From Figure 3, we can see that different from most existing models, which simply concatenate or multiply the semantic and temporal information together, TLT-KGE models the semantic information and temporal information as different axes of complex number space or quaternion space. This setting can not only distinguish the independence of the two kind of information, but also establish a connection between them.

4.2.3 Score Functions. Because there are two ways to combine the embeddings above, we utilize two different score functions to calculate the probability of a given quadruple (s, p, o, τ). The first score function is based on the Hamilton product of two complex numbers and the second one uses the Hamilton product between two quaternions.

Score Function of TLT-KGE (Complex). We first use the Hamilton product between two complex vector $\mathbf{e}_{s,\tau}^c$ and $\mathbf{r}_{p,\tau}^c$. The operation is shown as Equation 7.

$$\begin{aligned}
\mathbf{C}(s, p, \tau) &= \mathbf{e}_{s,\tau}^c \odot \mathbf{r}_{p,\tau}^c = (\mathbf{e}_s \circ \mathbf{r}_p - \mathbf{t}_\tau^e \circ \mathbf{t}_\tau^r) + (\mathbf{e}_s \circ \mathbf{t}_\tau^r + \mathbf{t}_\tau^e \circ \mathbf{r}_p)\mathbf{i} \\
&= \mathbf{c}_0 + \mathbf{c}_1\mathbf{i},
\end{aligned} \tag{7}$$

where $\odot$ denotes the Hamilton product of two complex numbers and $\circ$ represents the element-wise multiplication. $\mathbf{C}(s, p, \tau) \in \mathbb{C}^d$ represents the combination of s and p at timestamp τ. $\mathbf{c}_0$ and $\mathbf{c}_1$ is the real and imaginary part of $\mathbf{C}(s, p, \tau)$.

For providing more interactions between entities, relations and timestamps, we introduce another complex vector $\mathbf{C}'(s, p, \tau) = \mathbf{c}_1 + \mathbf{c}_0\mathbf{i}$, which exchanges the real and imaginary part of $\mathbf{C}(s, p, \tau)$. Then we utilize inner products of $(\mathbf{C}(s, p, \tau), \mathbf{e}_{o,\tau}^c)$ and $(\mathbf{C}'(s, p, \tau), \mathbf{e}_{o,\tau}^c)$ to calculate the score for TLT-KGE (Complex),

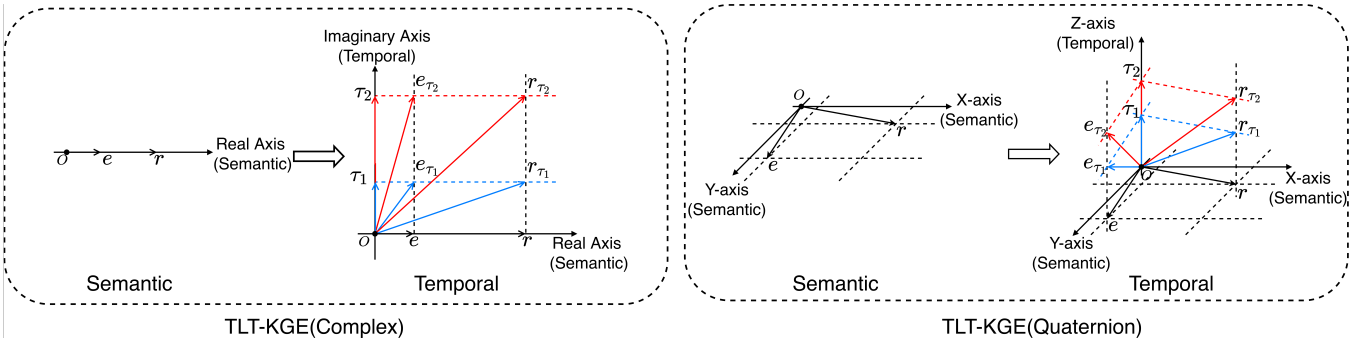


Figure 3: Illustration of the full representation combining semantic information and temporal information in TLT-KGE(Complex) and TLT-KGE(Quaternion). e and r are the semantic parts of entity and relation, respectively. τ_1 and τ_2 are two timestamps. Left: In TLT-KGE(Complex), the semantic information is modeled as the real part and the temporal information is regarded as the imaginary part of a complex vector. Right: The X-axis and Y-axis constitute a plane for the semantic information. The Z-axis represents the temporal information.

$$\begin{aligned}\phi^c(s,p,o,\tau) &= \langle \mathbf{C}(s,p,\tau), \mathbf{e}^c_{o,\tau}\rangle + \langle \mathbf{C}'(s,p,\tau), \mathbf{e}^c_{o,\tau}\rangle \\ &= \langle \mathbf{c}_0, \mathbf{e}_o\rangle + \langle \mathbf{c}_1, \mathbf{t}^e_\tau\rangle + \langle \mathbf{c}_0, \mathbf{t}^e_\tau\rangle + \langle \mathbf{c}_1, \mathbf{e}_o\rangle,\end{aligned} \tag{8}$$

where $\langle\cdot,\cdot\rangle$ denotes the sum of the element-wise product between two vector.

Score Function of TLT-KGE (Quaternion). We use Hamilton product between two quaternion vectors $\mathbf{e}^q_{s,\tau}$ and $\mathbf{r}^q_{p,\tau}$ to obtain the middle embedding $Q(s,p,\tau) \in \mathbb{H}^{\frac{d}{2}}$ as follow,

$$\begin{aligned}Q(s,p,\tau) &= \mathbf{e}^q_{s,\tau} \otimes \mathbf{r}^q_{p,\tau} \\ &= (\mathbf{e}_{s,a} \circ \mathbf{r}_{p,a} - \mathbf{e}_{s,b} \circ \mathbf{r}_{p,b} - \mathbf{t}^e_{\tau,c} \circ \mathbf{t}^r_{\tau,c} - \mathbf{t}^e_{\tau,d} \circ \mathbf{t}^r_{\tau,d}) \\ &\quad + (\mathbf{e}_{s,a} \circ \mathbf{r}_{p,b} + \mathbf{e}_{s,b} \circ \mathbf{r}_{p,a} + \mathbf{t}^e_{\tau,c} \circ \mathbf{t}^r_{\tau,d} - \mathbf{t}^e_{\tau,d} \circ \mathbf{t}^r_{\tau,c})\mathbf{i} \\ &\quad + (\mathbf{e}_{s,a} \circ \mathbf{t}^r_{\tau,c} - \mathbf{e}_{s,b} \circ \mathbf{t}^r_{\tau,d} + \mathbf{t}^e_{\tau,c} \circ \mathbf{r}_{p,a} + \mathbf{t}^e_{\tau,d} \circ \mathbf{r}_{p,b})\mathbf{j} \\ &\quad + (\mathbf{e}_{s,a} \circ \mathbf{t}^r_{\tau,d} + \mathbf{e}_{s,b} \circ \mathbf{t}^r_{\tau,c} - \mathbf{t}^e_{\tau,c} \circ \mathbf{r}_{p,b} + \mathbf{t}^e_{\tau,d} \circ \mathbf{r}_{p,a})\mathbf{k} \\ &= \mathbf{q}_0 + \mathbf{q}_1\mathbf{i} + \mathbf{q}_2\mathbf{j} + \mathbf{q}_3\mathbf{k},\end{aligned} \tag{9}$$

where $\mathbf{q}_0$ is the real part of the quaternion vector $\mathbf{Q}(s,p,\tau)$. $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ are the imaginary parts.

Similar to TLT-KGE (Complex), we introduce another quaternion vector that exchanges the $\mathbf{q}_0$, $\mathbf{q}_1$ and $\mathbf{q}_2$, $\mathbf{q}_3$, which is represented as $\mathbf{Q}'(s,p,\tau) = \mathbf{q}_2 + \mathbf{q}_3\mathbf{i} + \mathbf{q}_0\mathbf{j} + \mathbf{q}_1\mathbf{k}$. The score is calculated by the quaternion inner products of $(\mathbf{Q}(s,p,\tau), \mathbf{e}^q_{o,\tau})$ and $(\mathbf{Q}'(s,p,\tau), \mathbf{e}^q_{o,\tau})$,

$$\begin{aligned}\phi^q(s,p,o,\tau) &= \langle \mathbf{Q}(s,p,\tau), \mathbf{e}^q_{o,\tau}\rangle + \langle \mathbf{Q}'(s,p,\tau), \mathbf{e}^q_{o,\tau}\rangle \\ &= \langle \mathbf{q}_0, \mathbf{e}_{o,a}\rangle + \langle \mathbf{q}_1, \mathbf{e}_{o,b}\rangle + \langle \mathbf{q}_2, \mathbf{t}^e_{\tau,c}\rangle + \langle \mathbf{q}_3, \mathbf{t}^e_{\tau,d}\rangle \\ &\quad + \langle \mathbf{q}_0, \mathbf{t}^e_{\tau,c}\rangle + \langle \mathbf{q}_1, \mathbf{t}^e_{\tau,d}\rangle + \langle \mathbf{q}_2, \mathbf{e}_{o,a}\rangle + \langle \mathbf{q}_3, \mathbf{e}_{o,b}\rangle.\end{aligned} \tag{10}$$

4.2.4 Additional Modules. In this section, we introduce two additional modules to learn better representations.

Shared Time Window (STW). Existing TKGE models ignore the connections between the adjacent timestamps while embedding the entities or relations in the ccurrent time. Indeed, events that happened within a time scope are often full of connections. E.g. for the two facts (*COVID-19, infects, Alice, 2021/11/01*), (*Alice, meets, Bob, 2021/11/03*), most existing temporal KGE models cannot take advantage of the correlations between them. To this end, we propose the Shared Time Window (STW) module to strengthen the connections of events within a range of time.

Specifically, we define the size of time windows W, which is a hyper-parameter. In each time window, we utilize a shared embedding as an indicator to learn the interactions of events. Formally, for each timestamp τ, the embeddings with STW of entities $\mathbf{t}^{e'}_\tau \in \mathbb{R}^d$ and relations $\mathbf{t}^{r'}_\tau \in \mathbb{R}^d$ is composed of two parts, and they are calculated as follows,

$$\mathbf{t}^{e'}_\tau = \mathbf{t}^e_\tau + \mathbf{t}^e_{s,\tau'}, \quad \mathbf{t}^{r'}_\tau = \mathbf{t}^r_\tau + \mathbf{t}^r_{s,\tau'}, \tag{11}$$

where $\mathbf{t}^{e/r}_\tau \in \mathbb{R}^d$ is the temporal embedding that represents the special characteristics of each timestamps. $\mathbf{t}^{e/r}_{s,\tau'} \in \mathbb{R}^d$ is the shared embedding that encodes the rich correlations within a time window. Here we take the temporal embeddings of entities as an example to explain how to calculate the number of shared time windows. For $|\mathcal{T}|$ timestamps, the temporal embedding set of individual timestamps is $\{\mathbf{t}^e_1, \mathbf{t}^e_2, \ldots, \mathbf{t}^e_{|\mathcal{T}|}\}$ and the embedding set of shared timestamps is $\{\mathbf{t}^e_{s,1}, \mathbf{t}^e_{s,2}, \ldots, \mathbf{t}^e_{s,m}\}$, where $m = \lceil \frac{|\mathcal{T}|}{W} \rceil$ is the number of shared embeddings and W is the window size. Thus, the index τ' is calculated as $\lfloor \frac{\tau}{W} \rfloor$, where τ is the current timestamp.

In order to apply STW module to TLT-KGE, we can replace $\mathbf{t}^e_\tau$, $\mathbf{t}^r_\tau$ in Equation 5 or Equation 6 with $\mathbf{t}^{e'}_\tau$, $\mathbf{t}^{r'}_\tau$, respectively.

Relation-Timestamp Composition (RTC). The relation embeddings are crucial in temporal knowledge graph completion since each relation is related to multiple facts in KG. To enhance the expression of a relation at specific time, we introduce an relation-timestamp composition operation, which combines both the semantic information and temporal information with element-wise product. We empirically find this setting is beneficial to learn better relation representations and achieve better performance.

The operation are shown as below,

$$\begin{aligned}\mathbf{r}'_p &= \mathbf{r}_p + \mathbf{r}_{comp_\tau} \\ \mathbf{r}_{comp_\tau} &= \mathbf{r}_p \circ \mathbf{t}_{comp_\tau},\end{aligned} \tag{12}$$

where $\mathbf{r}_{comp_\tau} \in \mathbb{R}^d$ is a relation-timestamp composition embedding and $\mathbf{t}_{comp_\tau} \in \mathbb{R}^d$ is a learnable parameter related to timestamps. We can replace $\mathbf{r}_p$ in Equation 5 or Equation 6 with $\mathbf{r}'_p$ to utilize RTC for TLT-KGE (Complex) or TLT-KGE (Quaternion), respectively.

4.3 Regularization

The regularization term is set to prevent the model from overfitting and improve the generalization performance. In this subsection, we detail the regularization terms that we use.

In our experiments, the training parameters of TLT-KGE can be divided into two parts: (i) the semantic embeddings of entities and relations; (ii) the temporal embeddings of timestamps. Therefore, we utilize the same regularization terms as [18] for the semantic embeddings. The first regularization term Ω for entities and relations is shown as Equal (13).

$$\Omega = \sum_{(s,p,o,\tau)\in\mathcal{G}} (\|\mathbf{e}_s\|_3^3 + \|\mathbf{e}_o\|_3^3 + \|\mathbf{r}_p\|_3^3), \tag{13}$$

where $\|\cdot\|_3$ denotes the 3-norm operation for vectors.

In addition, the temporal embeddings of timestamps are supposed to gradually change over time for learning better representations, i.e., the embeddings from adjacent timestamps should be close. To achieve this, we add the following regularization term Λ,

$$\begin{aligned}\Lambda = \frac{1}{|\mathcal{T}|-1}\sum_{i=1}^{|\mathcal{T}|-1} (&\|\mathbf{t}^e_{i+1} - \mathbf{t}^e_i\|_4^4 + \|\mathbf{t}^r_{i+1} - \mathbf{t}^r_i\|_4^4 \\ &+ \|\mathbf{t}_{comp_{i+1}} - \mathbf{t}_{comp_i}\|_4^4) + \frac{1}{\lceil \frac{|\mathcal{T}|}{W} \rceil - 1}\sum_{i=1}^{\lceil \frac{|\mathcal{T}|}{W} \rceil - 1} (\|\mathbf{t}^e_{s,i+1} - \mathbf{t}^e_{s,i}\|_4^4 \\ &+ \|\mathbf{t}^r_{s,i+1} - \mathbf{t}^r_{s,i}\|_4^4),\end{aligned} \tag{14}$$

where the subscript i represents the i-th timestamp in $\mathcal{T}$.

4.4 Loss function

The loss function contains two parts: one is the multi-class log-loss [18, 19] used to evaluate the positive or negative facts, and the other is the regularization terms Ω and Λ. The loss function is

$$\begin{aligned}\mathcal{L}(\mathcal{G};\Theta) = \sum_{(s,p,o,\tau)\in\mathcal{G}} \Big[&- \log\Big(\frac{\exp(\phi^{\{\cdot\}}(s,p,o,\tau))}{\sum_{o'}\exp(\phi^{\{\cdot\}}(s,p,o',\tau))}\Big) \\ &- \log\Big(\frac{\exp(\phi^{\{\cdot\}}(s,p,o,\tau))}{\sum_{s'}\exp(\phi^{\{\cdot\}}(s',p,o,\tau))}\Big)\Big] + \lambda_1\Omega + \lambda_2\Lambda,\end{aligned} \tag{15}$$

where $\{\cdot\}$ can be c or q and Θ represents all the model parameters. o' and s' is the object of the negative quadruple (s,p,o',τ) and the subject of the negative quadruple (s',p,o,τ), respectively. λ_1 and λ_2 are the hyper-parameters.

5 EXPERIMENT

In this section, we first introduce the experimental setup, including datasets, baselines, evaluation metrics and implementation details. Then, in the main experiment and parameter-bounded experiment, we analyze the performance on different benchmarks. Besides, the ablation study of different modules and the parameter study of the size of shared time window are given. Finally, we will analyze the distribution of the embeddings of TLT-KGE and other models.

5.1 Experimental Setup

5.1.1 Datasets. In order to prove the adaptive ability of TLT-KGE in modeling sparse and dense TKGs, we evaluate our model on three popular benchmarks for TKG completion, including ICEWS14, ICEWS05-15 and GDELT. The first two datasets are sparse TKGs, introduced by [13]. ICEWS14 and ICEWS05-15 are two sub-sampling datasets from the Integrated Conflict Early Warning System (ICEWS). The latter GDELT is a dense KG, which is a subset of a larger knowledge graph named Global Database of Events, Language, and Tone (GDELT). And the dataset contains the facts about human behaviors with timestamps from April 1, 2015 to March 31, 2016. What's more, GDELT contains more than three million quadruples but only has 500 entities and 20 relations, making it a very dense dataset.

Table 1: Statistics for ICEWS14, ICEWS05-15 and GDELT. $|\mathcal{G}_{train}|$, $|\mathcal{G}_{valid}|$, $|\mathcal{G}_{test}|$ and $|\mathcal{G}|$ represent the number of facts in training sets, validation sets, test sets and KG, respectively.

	ICEWS14	ICEWS05-15	GDELT
$\lvert\mathcal{E}\rvert$	7,128	10,488	500
$\lvert\mathcal{R}\rvert$	230	251	20
$\lvert\mathcal{T}\rvert$	365	4,017	366
$\lvert\mathcal{G}_{train}\rvert$	72,826	386,962	2,735,685
$\lvert\mathcal{G}_{valid}\rvert$	8,963	46,092	341,961
$\lvert\mathcal{G}_{test}\rvert$	8,941	46,275	341,961
$\lvert\mathcal{G}\rvert$	90,730	479,329	3,419,607
Time span	2014	2005-2015	2015-2016
Granularity	Daily	Daily	Daily

5.1.2 Baselines. Our experiments compare TLT-KGE with several static KGE models and state-of-the-art temporal KGE models. For the static KGE models, we compare our model with TransE [5], DistMult [34]. For the temporal KGE models, we use TTransE [20], HyTE [10], TA-DistMult [13], DE-SimplE [14], TeRo [33], T(NT) ComplEx [18] and BoxTE [25] as the baselines.

5.1.3 Evaluation Metrics. We evaluate our model on link prediction task. In the text phase, we first replace the s and o with all the entities from $\mathcal{E}$ in turn for each quadruple in the test set. Then we compute the scores of all the corrupted quadruples (i.e. $(s,p,?,\tau)$ or $(?,p,o,\tau)$) and rank all the candidate entities according to the scores under the time-wise filtered settings [14]. Positive candidates are supposed to precede negative ones. we use the following metrics for comparison: (i) Mean Rank (MR, the mean of all the predicted ranks); (ii) Mean Reciprocal Rank (MRR, the mean of all the reciprocals of predicted ranks); (iii) Hits@n (the proportion of ranks not larger than n). Lower MR and larger MRR and Hits@n indicate better performance.

5.1.4 Implementation Details. We conduct two experiments to evaluate the performance of TLT-KGE:

- **Main Experiment.** In this experiment, we do the temporal knowledge graph completion on three datasets. The dimension size of embeddings d is set as 1200, 1200 and 1500 for ICEWS14,

Table 2: Experimental results on ICEWS14, ICEWS05-15 and GDELT. Results with [▲] and [◇] are taken from [9] and [25], respectively. For T(NT)ComplEx, we use the official implementation to re-implement the results to obtain MR metric.

Model	ICEWS14					ICEWS05-15					GDELT				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE [▲]	-	28.0	9.4	-	63.7	-	29.4	9.0	-	66.3	-	11.3	0.0	15.8	31.2
DistMult [▲]	-	43.9	32.3	-	67.2	-	45.6	33.7	-	69.1	-	19.6	11.7	20.8	34.8
TTransE [▲]	-	25.5	7.4	-	60.1	-	27.1	8.4	-	61.6	-	11.5	0.0	16.0	31.8
HyTE [▲]	-	29.7	10.8	41.6	65.5	-	31.6	11.6	44.5	68.1	-	11.8	0.0	16.5	32.6
TA-DistMult [▲]	-	47.7	36.3	-	68.6	-	47.4	34.6	-	72.8	-	20.6	12.4	21.9	36.5
DE-SimplE [▲]	-	52.6	41.8	59.2	72.5	-	51.3	39.2	57.8	74.8	-	23.0	14.1	24.8	40.3
TeRo [▲]	-	56.2	46.8	62.1	73.2	-	58.6	46.9	66.8	79.5	-	24.5	15.4	26.4	42.0
RotateQVS [▲]	-	59.1	50.7	64.2	75.4	-	63.3	52.9	70.9	81.3	-	27.0	17.5	29.3	45.8
TComplEx	217	61.9	54.2	66.1	76.7	113	66.5	58.3	71.6	81.1	48	34.6	25.9	37.2	51.5
TNTComplEx	239	60.7	51.9	65.9	77.2	111	66.6	58.3	71.8	81.7	47	34.1	25.2	36.8	51.5
BoxTE (k=2) [◇]	161	61.5	53.2	66.7	76.7	98	66.4	57.6	72.0	82.2	48	33.9	25.1	36.6	50.7
BoxTE (k=3) [◇]	162	61.4	53.0	66.8	76.5	101	66.6	58.2	71.9	82.0	49	34.4	25.9	36.9	50.7
BoxTE (k=5) [◇]	160	61.3	52.8	66.4	76.3	96	66.7	58.2	71.9	82.0	50	35.2	**26.9**	37.7	51.1
TLT-KGE(Complex)	161	63.0	54.9	67.8	77.7	94	68.6	60.7	73.5	83.1	45	35.6	26.7	38.5	53.2
TLT-KGE(Quaternion)	**144**	**63.4**	**55.1**	**68.4**	**78.6**	**89**	**69.0**	**60.9**	**74.1**	**83.5**	**43**	**35.8**	26.5	**38.8**	**54.3**

Table 3: Experimental Results on ICEWS14, ICEWS05-15 and GDELT in the bounded-parameter setting. For T(NT)ComplEx, we use the official implementation to re-implement the results on three datasets. Other results are taken from [25].

Model	ICEWS14					ICEWS05-15					GDELT				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
DE-SimplE	-	52.6	41.8	59.2	72.5	-	51.3	39.2	57.8	74.8	-	23.0	14.1	24.8	40.3
TComplEx	246	57.0	48.5	61.9	72.4	134	58.7	49.4	64.0	76.0	**61**	24.5	16.3	26.2	40.3
TNTComplEx	257	56.9	47.7	62.3	74.1	122	60.0	50.5	65.0	76.9	**61**	24.4	16.2	26.1	40.2
BoxTE(k=1)	183	57.6	47.8	63.9	75.3	122	56.4	45.2	63.5	77.0	62	**25.0**	**16.7**	**27.0**	**41.1**
BoxTE(k=2)	177	58.0	48.3	64.2	75.5	110	56.7	45.8	63.1	77.5	63	24.6	16.4	26.5	40.4
BoxTE(k=3)	182	58.2	49.1	64.0	74.8	125	57.0	46.5	63.6	76.3	64	24.2	16.1	26.0	39.8
BoxTE(k=5)	183	58.1	49.3	63.2	74.2	134	56.7	46.9	62.3	74.6	66	23.6	15.6	25.3	39.0
TLT-KGE(Complex)	192	59.7	**51.8**	64.5	74.4	116	60.8	51.6	66.2	78.3	**61**	24.7	16.4	26.5	40.6
TLT-KGE(Quaternion)	**175**	**59.9**	51.3	**64.9**	**75.6**	**108**	**61.0**	**51.9**	**66.4**	**78.6**	**61**	24.8	16.5	26.8	40.7

ICEWS05-15 and GDELT, respectively. All the dimension sizes are smaller than that in T(NT)ComplEx and BoxTE. Then we optimize our model with Adagrad [12] and the learning rate is 0.1. The max training epoch is 200. For the parameters of regularization terms in TLT-KGE (Complex), λ_1 is set to 0.001 and λ_2 is set to 0.1 on all datasets. In TLT-KGE(Quaternion), λ_1 is set to 0.003, 0.001 and 0.0005 on ICEWS14, ICEWS05-15 and GDELT, respectively. λ_2 is set to 0.03, 0.1 and 0.03 on ICEWS14, ICEWS05-15 and GDELT, respectively. The size of shared time window W is 120 days (4 months) on both ICEWS14 and GDELT, and 1440 days (4 years) on ICEWS05-15.

- **Parameter-bounded Experiment.** To investigate the robustness of TLT-KGE with a restricted computational budget, we keep the same number of parameters as the 100-dimensional DE-SimplE to conduct the parameter-bounded experiment. Table 4 gives the number of parameters of different competitors. The size of shared time window W is set to 122 for ICEWS14 and GDELT, and 1440 for ICEWS05-15. According to these settings, we can calculate that the dimension of TLT-KGE (Complex / Quaternion) is 152, 104 and 65 on ICEWS14, ICEWS05-15 and GDELT, respectively. Finally, other hyper-parameters keep the same as the main experiment above.

5.2 Main Experimental Results

Experimental results are shown in Table 2. **Bold** font represents the best results, and underlined font denotes the second-best results. From Table 2, we have the following findings.

(i) Generally, TLT-KGE (Quaternion) and TLT-KGE (Complex) outperform all the state-of-the-art competitors, demonstrating the effectiveness of our model. It is also worth noting that the proposed method achieves better results than baselines on all the three dataset, indicating that our model is suitable for datasets with different sparsity.

(ii) We find TLT-KGE (Quaternion) performs better than TLT-KGE (ComplEx). Paricularly, TLT-KGE (Quaternion) achieves better MR score than TLT-KGE (ComplEx) on ICEWS14 by 17. The results

indicate that the interactions in the quaternion vector space is more powerful for representing entities and relations with timestamps.

(iii) We find that our model achieves greater performance gains on the two sparser datasets ICEWS14 and ICEWS05-15. Specifically, TLT-KGE (Quaternion) outperforms the best baseline by 1.5%, 2.3% and 0.6% on the three datasets, respectively. The results show that our way of modeling semantic information and temporal information can overcome the problem of data sparseness to a certain extent.

In a nutshell, TLT-KGE makes a remarkable performance on three widely used datasets. The method that regards the time information as a timeline is adequate for temporal knowledge graph completion task.

Table 4: Model parameter count for TLT-KGE and other competing models. d is the dimension size. For TLT-KGE, m is the number of shared time windows, which is calculated by $\lceil \frac{|\mathcal{T}|}{W} \rceil$). For DE-SimplE, γ denotes the shared rate of temporal embedding features. For BoxTE, k donates the number of embeddings for every timestamp.

Model	Number of Parameters
DE-SimplE	$2d((3\gamma + (1-\gamma))\|\mathcal{E}\| + \|\mathcal{R}\|)$
TNTComplEx	$2d(\|\mathcal{E}\| + \|\mathcal{T}\| + 4\|\mathcal{R}\|)$
BoxTE	$d(2\|\mathcal{E}\| + k\|\mathcal{T}\| + 2\|\mathcal{R}\|) + k\|\mathcal{R}\|$
TLT-KGE(Complex)	$2d(\|\mathcal{E}\| + 2\|\mathcal{R}\| + 3\|\mathcal{T}\| + m)$
TLT-KGE(Quaternion)	$2d(\|\mathcal{E}\| + 2\|\mathcal{R}\| + 3\|\mathcal{T}\| + m)$

5.3 Parameter-bounded Experimental Results

Table 3 shows the results of parameter-bounded experiment. We can see that TLT-KGE has an outstanding improvement on ICEWS14 and ICEWS05-15, indicating that our model has strong robustness compared with other baselines. Again, the results demonstrate that TLT-KGE is capable of modeling the time evolution on sparse datasets. For GDELT, TLT-KGE (Quaternion) achieves the second-best result. We infer the reason is that the small dimension size (65 in ours vs. 128 in BoxTE(k=1)) might limit the expression ability for such a large dataset.

To further analyze the reason, we remove the STW and RTC modules in both TLT-KGE(Complex) and TLT-KGE(Quaternion) and conduct the experiment on GDELT. Under this setting, the dimension size of embedding changes to 92. The results are shown as Table 5. From the results, both TLT-KGE(Complex) and TLT-KGE(Quaternion) perform better than BoxTE(k=1), which supports our inference. Therefore, when the number of parameters is small, increasing the embedding size is substantially beneficial in the parameter-bounded setting for GDELT.

5.4 Ablation Study of Additional Modules

In this section, we compare different variants of TLT-KGE (Complex / Quaternion), removing the additional modules in **Section 4.2.4** to investigate their influence on the model. Table 6 shows the results of ablation studies. **Bold** fonts represent the best result in each group. From Table 6, we have following findings.

Table 5: Experimental Results of TLT-KGE without RTC and STW on GDELT in the bounded-parameter setting. C and Q denote Complex and Quaternion, respectively.

Model	MR	MRR	Hits@1	Hits@3	Hits@10
BoxTE(k=1)	62	25.0	16.7	27.0	41.1
TLT-KGE(C)-RTC-STW	60	25.1	16.8	26.8	41.0
TLT-KGE(Q)-RTC-STW	**59**	**25.3**	**16.9**	**27.1**	**41.2**

(i) Generally, the proposed model outperforms the variants without RTC and STW modules, which shows that both two modules have a significant impact on TLT-KGE(Complex) and TLT-KGE(Quaternion), and demonstrates that the additional designed modules have a positive influence on temporal knowledge graph completion task.

(ii) We find that the RTC module plays a more important role on the dense GDELT dataset. We conjecture the reason lies as follow. The RTC model aims to strengthen the representations of temporal relations. In the dense dataset GDELT, each relation is associated with more facts, thus the representations of the relations becomes more important. Therefore, the RTC module can play a better role in improving the performance.

(iii) We observe that STW module is more effective on the two sparse datasets ICEWS14 and ICEWS05-15. We conjecture the reason lies as follows. The module enables information sharing between adjacent timestamps. In this way, low-frequency entities and relations in a certain timestamp can be enriched with more information from adjacent timestamps, which is beneficial for better performance.

Moreover, TLT-KGE without both two additional modules also achieves state-of-the-art results on ICEWS14 and ICEWS05-15, and comparable results on GDELT. The phenomenon again verifies the effectiveness of our way in modeling temporal entities and relations.

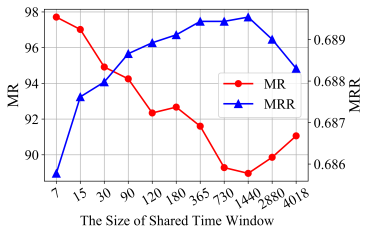


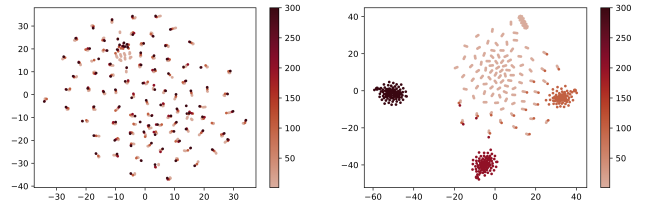
Figure 4: The change of MR and MRR with the size of shared time window (STW) W increasing on ICEWS05-15.

5.5 Parameter Study

We also carry out the parameter study to test the effects of the window size W of STW module on ICEWS05-15. Figure 4 presents the change of MRR and MR with W increasing on ICEWS05-15. TLT-KGE(Quaternion) keeps achieving better results as we raise W from 7 to the optimal value. Then, after W exceeds the optimal point, the performance starts falling down. The reason lies as: When W is small, an event cannot take full advantage of all the information from related events. In this way, the model cannot achieve the optimal results. When W is large, some weakly related or even unrelated events may join the same window, leading to unsatisfactory results.

Table 6: Experimental Results of two additional modules in Section 4.2.4. STW represents shared time window. RTC denotes relation-timestamp composition module.

Model	ICEWS14					ICEWS05-15					GDELT				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TLT-KGE(Complex)	161	**63.0**	**54.9**	67.8	77.7	94	**68.6**	**60.7**	**73.6**	83.1	**45**	**35.6**	**26.7**	**38.5**	**53.2**
TLT-KGE(Complex)-RTC	**153**	62.9	54.3	**68.1**	**78.1**	**90**	68.4	60.1	**73.6**	**83.4**	47	34.9	25.3	38.0	52.9
TLT-KGE(Complex)-STW	178	62.5	54.4	67.6	77.1	110	68.1	60.2	73.1	82.5	46	35.5	26.5	38.3	53.0
TLT-KGE(Complex)-RTC-STW	173	62.5	54.3	67.8	77.9	105	68.1	60.0	73.3	82.7	47	34.8	25.3	37.9	52.7
TLT-KGE(Quaternion)	**144**	**63.3**	**55.0**	**68.3**	**78.6**	**89**	**68.9**	**60.9**	**74.0**	**83.5**	**43**	**35.8**	26.5	**38.8**	**54.3**
TLT-KGE(Quaternion)-RTC	147	63.0	54.4	68.2	78.4	91	68.5	60.3	73.8	83.2	**43**	35.5	26.2	38.5	53.9
TLT-KGE(Quaternion)-STW	158	62.9	54.8	67.7	77.7	102	68.7	60.8	73.6	82.9	44	35.5	26.2	38.5	53.8
TLT-KGE(Quaternion)-RTC-STW	163	62.6	54.2	67.7	77.8	109	68.1	60.0	73.2	82.7	44	35.4	26.1	38.5	53.8



(a) Embeddings (Entities) in DE-SimplE

(b) Embeddings (Entities) in TLT-KGE(Quaternion)

(c) Embeddings (Relations) in TNTComplEx

(d) Embeddings (Relations) in TLT-KGE(Quaternion)

Figure 5: t-SNE visualization. The color from pale red to dark red represents the timestamps from 2014.01.01 (the first day) to 2014.11.28 (the last day). Each point corresponds to a temporal entity or relation.

5.6 Capacity of Distinguishing the Entities / Relations with Temporal Information

We discuss the ability of TLT-KGE in distinguishing entities or relations with time information in this section.

For all models, We randomly choose the same 100 entities, 50 relations and 8 timestamps (2014.1.1-2014.1.5, 2014.4.11, 2014.7.20 and 2014.11.28) for visualization on ICEWS14. We compare the embeddings of entities with DE-SimplE [14] and the embeddings of relations with TNTComplEx [18] because DE-SimplE utilizes diachronic embeddings only for entities while TNTComplEx applies a dynamic module on relations.

Figure 5 shows the 2D visualization of trained embeddings with the help of t-SNE [29]. We find the baseline models tend to embed the same entity (relation) with different timestamps into a cluster. Such a result makes it difficult for the baseline models to distinguish the representations of entities (relations) at different timestamps. While our model can easily do so, which again validates the efficiency of our methodology.

Table 7: Average distance of embeddings on three models. (·, ·) represents the distance is between two dates in 2014.

Model	Entity Distance		Relation Distance	
	(1.1, 1.1)	(1.1, 11.28)	(1.1,1.1)	(1.1,11.28)
DE-SimplE	36.5	40.4	-	-
TNTComplEx	-	-	46.1	56.4
TLT-KGE(Quaternion)	46.9	139.8	48.5	256.8

Besides, We also test the model's ability to discriminate between different entities (relations) at the same timestamp. Table 7 shows the results. We can observe that on Jan 1st, the average entity (relation) distance in TLT-KGE(Quaternion) is larger than that in baseline models, which demonstrates our model can also better distinguish entities (relations) at the same timestamp.

6 CONCLUSION

In this paper, we presented a novel model TLT-KGE for temporal knowledge graph completion, which regards the time information as a timeline. TLT-KGE embeds both entities and relations at different times with a semantic part from the static knowledge graph and a temporal part from the time information. TLT-KGE is qualified to model temporal evolution over the timeline with two independent parts. Additionally, the proposed method also introduced the shared time window module and the relation-timestamp composition module to enhance the ability of representation learning. Extensive experiments on benchmark datasets clearly validated the superiority of TLT-KGE against various state-of-the-art baselines.

ACKNOWLEDGMENTS

The research work supported by the National Natural Science Foundation of China under Grant No.U1836206, 61976204, 62176014, U1811461. Xiang Ao is also supported by the Project of Youth Innovation Promotion Association CAS, Beijing Nova Program Z201100006820062. Zhao Zhang is supported by the China Postdoctoral Science Foundation under Grant No. 2021M703273.

REFERENCES

[1] Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems* 33 (2020), 9649–9661.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.

[3] Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5185–5194.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[6] Elizabeth Boschee, Jennifer Lautenschlager, and et al. 2015. ICEWS coded event data. *Harvard Dataverse* (2015).

[7] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*. 151–161.

[8] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual quaternion knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6894–6902.

[9] Kai Chen, Ye Wang, Yitong Li, and Aiping Li. 2022. RotateQVS: Representing Temporal Information as Rotations in Quaternion Vector Space for Temporal Knowledge Graph Completion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5843–5857.

[10] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 2001–2011.

[11] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[12] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12, 7 (2011).

[13] Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4816–4821.

[14] Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3988–3995.

[15] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[16] William Rowan Hamilton. 1848. Xi. on quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 33, 219 (1848), 58–60.

[17] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6669–6683.

[18] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2019. Tensor Decompositions for Temporal Knowledge Base Completion. In *International Conference on Learning Representations*.

[19] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*. PMLR, 2863–2872.

[20] Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*. 1771–1776.

[21] Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, Vol. 2. Citeseer, 1–49.

[22] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. 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*. 408–417.

[23] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

[24] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2395–2405.

[25] Johannes Messner, Ralph Abboud, and Ismail Ilkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7779–7787.

[26] Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 327–333.

[27] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

[28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.

[29] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[30] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[31] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[32] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*. 763–772.

[33] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. TeRo: A Time-aware Knowledge Graph Embedding via Temporal Rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1583–1593.

[34] Bishan Yang, Wen-tau Yih, and et al. 2014. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

[35] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.

[36] Fuwei Zhang, Zhao Zhang, Xiang Ao, Dehong Gao, Fuzhen Zhuang, Yi Wei, and Qing He. 2022. Mind the Gap: Cross-Lingual Information Retrieval with Hierarchical Knowledge Enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence* (2022).

[37] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. *Advances in neural information processing systems* 32 (2019).

[38] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Thirty-second AAAI conference on artificial intelligence*.

[39] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, and Qing He. 2020. Relational graph neural network with hierarchical attention for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9612–9619.