

# RotateQVS: Representing Temporal Information as Rotations in Quaternion Vector Space for Temporal Knowledge Graph Completion

Kai Chen<sup>1</sup>, Ye Wang<sup>1,2\*</sup>, Yitong Li<sup>3</sup>, Aiping Li<sup>1</sup>

<sup>1</sup>National University of Defense Technology, Changsha, China

<sup>2</sup>Pengcheng Laboratory, Shenzhen, China

<sup>3</sup>Huawei Technologies Co., Ltd.

{chenkai\_, ye.wang, liaiping}@nudt.edu.cn liyitong3@huawei.com

## Abstract

Temporal factors are tied to the growth of facts in realistic applications, such as the progress of diseases and the development of political situation, therefore, research on Temporal Knowledge Graph (TKG) attracts much attention. In TKG, relation patterns inherent with temporality are required to be studied for representation learning and reasoning across temporal facts. However, existing methods can hardly model temporal relation patterns, nor can capture the intrinsic connections between relations when evolving over time, lacking of interpretability. In this paper, we propose a novel temporal modeling method which represents temporal entities as **Rotations** in Quaternion Vector Space (RotateQVS) and relations as complex vectors in Hamilton’s quaternion space. We demonstrate our method can model key patterns of relations in TKG, such as symmetry, asymmetry, inverse, and can further capture time-evolved relations by theory. Empirically, we show that our method can boost the performance of link prediction tasks over four temporal knowledge graph benchmarks.

## 1 Introduction

Knowledge Graphs (KGs) have been widely adopted to represent informative knowledge or facts in real-world applications (Bollacker et al., 2008; Miller, 1995; Suchanek et al., 2007). However, as known facts are usually sparse, KGs are far from completeness. Thus, Knowledge Graph Completion (KGC) methods are proposed to predict missing facts, i.e. links between entities (Bordes et al., 2013; Yang et al., 2015; Dettmers et al., 2018; Chen et al., 2021b). Furthermore, in real world, many facts are bonded with a particular time by nature. For example, *Barack Obama is the president of USA* is only valid for the time period 2009 - 2017. To model such time-sensitive facts, Temporal Knowledge Graphs (TKGs) have

recently drawn growing attention from both academic and industrial communities (Lautenschlager et al., 2015; Leetaru and Schrod, 2013).

TKG Embedding (TKGE) methods (Jiang et al., 2016; Dasgupta et al., 2018; Jin et al., 2020; Sadeghian et al., 2021) were proposed to represent entities and relations with temporal features in TKGs (Lautenschlager et al., 2015; Leetaru and Schrod, 2013). But how to present them with temporal interpretability remains a challenge for state-of-the-art TKGE models. Further, it is crucial for TKG Completion (TKGC) to leverage the learned temporal information. Previous static KGC works (Sun et al., 2020; Schlichtkrull et al., 2018; Gao et al., 2020) learn explainable embeddings of various relation patterns, so that symmetric pattern (e.g. “co-author”), asymmetric pattern (e.g. “affiliation”), inverse pattern (e.g. “buyer” vs. “seller”) and complex composition pattern (e.g. “father’s wife (mother)” vs. “wife’s father (father in law)”) can be captured in static KGs. However, in TKGs, there are inherent connections between entities and their relations along with time-evolving. For example, the relation between *Kit Harington* and *Rose Leslie* is *in love* in 2012, becomes *engaged* in 2017, and then turns into *married* in 2018. To the best of our knowledge, very few of the existing TKGE methods can capture them.

To address this problem, we take inspirations from Hamilton’s quaternion number system (Hamilton, 1844; Zhang et al., 2019a; Gao et al., 2020) and propose a novel method based on quaternion. To be specific, we encode both entities and relations as quaternion embeddings, and then temporal entity embeddings can be represented as **Rotations** in Quaternion Vector Space (RotateQVS). Theoretically, we show the limitations of previous methods and demonstrate that performing quaternion embeddings can model symmetric, asymmetric, and inverse relation patterns. Meanwhile, we prove our method is capable of cap-

\* Corresponding author.

turing time-evolving information in TKG explicable. We empirically evaluate our method over four TKGC benchmarks and report state-of-the-art performance consistently. Further, we perform analysis of the learned quaternion embeddings and show the abilities of our RotateQVS for modeling various relation patterns, including temporal evolution.

We summarize our main contributions as follows:

1. We propose an original quaternion based TKGC method, namely RotateQVS, which represents temporal information as rotations in quaternion vector space.
2. We study temporal evolving relations, and we demonstrate the proposed RotateQVS can model various relation patterns including temporal evolution both theoretically and empirically.
3. Our RotateQVS outperforms the SOTA methods over all of ICEWS14, ICEWS05-15, YAGO11k and GDELT datasets on link prediction task.

## 2 Preliminaries on Hamilton’s Quaternions

Quaternion number system ([Hamilton, 1844](#)) is an extension of traditional complex numbers. Recently, quaternion has been applied in static knowledge graph embedding ([Zhang et al., 2019a; Gao et al., 2020](#)). For readers better understanding our method in Section 3, we introduce the definition and basic operations of quaternion in this section.

### 2.1 Quaternion Operations

A quaternion is expressed as  $q = a + bi + cj + dk$ , and some key quaternion operations are defined as:

**Conjugate** Similar to a traditional complex number, the conjugate of a quaternion is defined with the same real part and the opposite imaginary parts, that is

$$\bar{q} = a - bi - cj - dk.$$

**Inner Product** The inner 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 the sum of product of each corresponding factor

$$q_1 \cdot q_2 = \langle a_1, a_2 \rangle + \langle b_1, b_2 \rangle + \langle c_1, c_2 \rangle + \langle d_1, d_2 \rangle.$$

**Norm** With the definition of conjugate and inner product, the norm of a quaternion is defined as:

$$\|q\| = \sqrt{\bar{q} \cdot q} = \sqrt{\bar{q} \cdot q} = \sqrt{a^2 + b^2 + c^2 + d^2} \quad (1)$$

**Inverse** The inverse of a quaternion is defined from  $q^{-1} \cdot q = q \cdot q^{-1} = 1$ . Multiplying by  $\bar{q}$ , we have  $\bar{q} \cdot q \cdot q^{-1} = \bar{q}$ , derived from which we get:

$$q^{-1} = \frac{\bar{q}}{\|q\|^2} \quad (2)$$

**Hamilton Product** For two quaternions  $q_1$  and  $q_2$ , their product is determined by the products of the basis elements and the distributive law. The quaternion multiplication formula is:

$$\begin{aligned} q_1 q_2 &= (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &\quad + (a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2)\mathbf{i} \\ &\quad + (a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2)\mathbf{j} \\ &\quad + (a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2)\mathbf{k} \end{aligned} \quad (3)$$

Considering the conjugate of Hamilton product, we can further deduce:

$$\begin{aligned} \bar{q}_1 q_2 &= \bar{q}_2 \bar{q}_1, \\ \bar{q}_1 q_2 q_3 &= \bar{q}_3 \bar{q}_2 \bar{q}_1. \end{aligned} \quad (4)$$

### 2.2 3D Vector Space

In fact, the imaginary part  $b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$  of a quaternion behaves like a vector  $\mathbf{v} = (b, c, d)$  in a 3D vector space. Thus, conveniently, we rewrite a quaternion using **imaginary vectors**:

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = a + \mathbf{v} = (a, \mathbf{0}) + (0, \mathbf{v}). \quad (5)$$

**Multiplication rule** The multiplication of two imaginary vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  is

$$\mathbf{v}_1 \mathbf{v}_2 = \mathbf{v}_1 \times \mathbf{v}_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, \quad (6)$$

where  $\mathbf{v}_1 \times \mathbf{v}_2$  is vector cross product, resulting in a vector, and  $\mathbf{v}_1 \cdot \mathbf{v}_2$  is the dot product, which gives a scalar. Obviously, the multiplication of two imaginary vectors is non-commutative, as the cross product is non-commutative.

Thus, the multiplication of two quaternions can be rewritten in 3D vector perspective:

$$\begin{aligned} q_1 q_2 &= (a_1, \mathbf{v}_1) (a_2, \mathbf{v}_2) \\ &= (a_1 a_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, a_1 \mathbf{v}_2 + a_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2) \end{aligned} \quad (7)$$

### 3 Proposed Method

In this section, we introduce a novel temporal modeling approach for TKG by representing temporal information as **Rotations** in Quaternion Vector Space (RotateQVS).

#### 3.1 Notations

Suppose that we have a temporal knowledge graph, noted as  $\mathcal{G}$ . We use  $\mathcal{E}$  to denote the set of entities,  $\mathcal{R}$  to denote the set of relations, and  $\mathcal{T}$  to denote the set of time stamps. Then, the temporal knowledge graph  $\mathcal{G}$  can be defined as a collection of quadruples, noted as  $(s, r, o, t)$ , where a relation  $r \in \mathcal{R}$  holds between a head entity  $s \in \mathcal{E}$  and an tail entity  $o \in \mathcal{E}$  at time  $t$ . The actual time  $t$  is represented by a time stamp  $\tau \in \mathcal{T}$ .

#### 3.2 Representing Temporal Information using Rotations in 3D Vector Space

Similar to Tero (Xu et al., 2020a) which utilizes a rotation in complex space, we also represent temporal information using rotations while in the quaternion vector space.

In 3D vector space, according to *Euler's rotation theorem* (Euler, 1776; Verhoeff, 2014), any rotation or sequence of rotations of a rigid body or a coordinate system about a fixed point is equivalent to a single rotation by a given angle  $\theta$  about a fixed axis (called the Euler axis) that runs through the fixed point. And an extension of Euler's formula for quaternion can be expressed as follows:

$$\begin{aligned} q &= e^{\frac{\theta}{2}(v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k})} \\ &= \cos \frac{\theta}{2} + (v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}) \sin \frac{\theta}{2}, \end{aligned} \quad (8)$$

where  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  are unit vectors representing the three Cartesian axes.

##### 3.2.1 Representing Time, Entities, and Relations:

Quaternions provide us with a simple way to encode this axis-angle representation in four numbers, and can be used to perform the rotation procedure in 3D vector space. By doing so, we constrain the **time stamp** embedding  $\tau$  as a unit quaternion as

$$\tau = \cos \frac{\theta_\tau}{2} + \mathbf{u}_\tau \sin \frac{\theta_\tau}{2}, \quad (9)$$

where  $\mathbf{u}_\tau$  is a unit vector in the quaternion space. And for other elements of a quadruple  $(s, r, o, t)$ , based on the Hamilton's quaternions in Section 2,

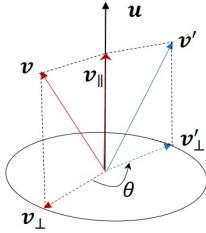


Figure 1: An illustration of the proposed rotation in 3D vector space, where  $\mathbf{v}'$  is the result of vector  $\mathbf{v}$  rotating  $\theta$  around the rotation axis  $\mathbf{u}$ .

we map each of them to its base, which is a time-independent quaternion embedding:

$$\begin{aligned} \mathbf{s} &= \{\mathbf{a}_s + \mathbf{b}_s\mathbf{i} + \mathbf{c}_s\mathbf{j} + \mathbf{d}_s\mathbf{k}\} \\ \mathbf{r} &= \{\mathbf{a}_r + \mathbf{b}_r\mathbf{i} + \mathbf{c}_r\mathbf{j} + \mathbf{d}_r\mathbf{k}\} \\ \mathbf{o} &= \{\mathbf{a}_o + \mathbf{b}_o\mathbf{i} + \mathbf{c}_o\mathbf{j} + \mathbf{d}_o\mathbf{k}\}, \end{aligned} \quad (10)$$

where  $\mathbf{a}_{\{\cdot\}}, \mathbf{b}_{\{\cdot\}}, \mathbf{c}_{\{\cdot\}}, \mathbf{d}_{\{\cdot\}} \in \mathbb{R}^k$ .

##### 3.2.2 Temporal Entities:

We make use of the quaternion rules to represent temporal information as rotations in 3D vector space. An abstract rotation procedure is illustrated in Figure 1.

**Theorem 1.** Given a unit quaternion  $q = \cos \frac{\theta}{2} + \mathbf{u} \sin \frac{\theta}{2}$ , where  $\mathbf{u} \in \mathbb{R}\mathbf{i} + \mathbb{R}\mathbf{j} + \mathbb{R}\mathbf{k}$  is a unit vector (rotation axis) in a three-dimensional space, the result of vector  $\mathbf{v}$  rotating  $\theta$  around the rotation axis  $\mathbf{u}$  is

$$\mathbf{v}' = q\mathbf{v}q^{-1} = q\mathbf{v}\bar{q}. \quad (11)$$

Theorem 1 is supported by Rodrigues' rotation formula (Rodrigues, 1840).<sup>1</sup> We then define the functional mapping that reflects the temporal evolution of an entity embedding. For each time stamp  $\tau$ , the functional mapping is an element-wise rotation from the basic entity embedding  $\mathbf{e}$  (quaternion representation) to the **time-specific entity** embedding  $\mathbf{e}_t$ , which is as follows:

$$\begin{aligned} \mathbf{e}_t &= \tau\mathbf{e}\tau^{-1} = \tau(a_e + \mathbf{v}_e)\tau^{-1} \\ &= a_e\tau\tau^{-1} + \tau\mathbf{v}_e\tau^{-1} \\ &= a_e + \tau\mathbf{v}_e\tau^{-1}, \end{aligned} \quad (12)$$

where  $a_e$  and  $\mathbf{v}_e$  are the scalar/real and vector/imaginary part of the entity quaternion representation  $\mathbf{e}$  respectively. And according to Theorem 1,  $\tau\mathbf{v}_e\tau^{-1}$  is the result of vector  $\mathbf{v}_e$  rotating  $\theta_\tau$  around the rotation axis  $\mathbf{u}_\tau$  ( $\tau = \cos \frac{\theta_\tau}{2} + \mathbf{u}_\tau \sin \frac{\theta_\tau}{2}$ , see

<sup>1</sup>See proof in Appendix A

Equation 9) which constitutes the vector/imaginary part of  $\mathbf{e}_t$ . Thus, we can get a lemma:

**Lemma 1.** *The vector (imaginary) part is rotated while the scalar (real) part remains unchanged in the functional mapping (Equation 12) which reflects the temporal evolution of an entity embedding.*

For a quadruple  $(s, r, o, t)$ , we make use of the functional mapping to get the time-specific entity embeddings  $\mathbf{s}_t$  and  $\mathbf{o}_t$  from the basic entity embeddings  $\mathbf{s}$  and  $\mathbf{o}$ :

$$\mathbf{s}_t = \boldsymbol{\tau} \mathbf{s} \boldsymbol{\tau}^{-1}, \quad \mathbf{o}_t = \boldsymbol{\tau} \mathbf{o} \boldsymbol{\tau}^{-1}. \quad (13)$$

Considering the temporal evolution of entity embedding, the **relation** embedding  $\mathbf{r}$  is regarded as a translation from the time-specific subject embedding  $\mathbf{s}_t$  to the conjugate of the time-specific object embedding  $\overline{\mathbf{o}_t}$ . In other words, we aim to make  $\mathbf{s}_t + \mathbf{r} = \overline{\mathbf{o}_t}$  for all positive quadruples. Then, the score function can be defined as:

$$f(s, r, o, t) = \|\mathbf{s}_t + \mathbf{r} - \overline{\mathbf{o}_t}\|. \quad (14)$$

Note that each embedding above is a quaternion representation, and “ $\| \cdot \|$ ” denotes the norm computation (see Equation 1).

### 3.2.3 Loss Function

We use the same margin loss function with multiple negative sampling as proposed in (Sun et al., 2019), which has been proved to be effective on distance-based KGE models (Bordes et al., 2013; Sun et al., 2019) and as well as the TKGE models (Xu et al., 2019, 2020a). In details, our loss function is

$$\mathcal{L} = -\log \sigma(\gamma - f(\xi)) - \sum_{i=1}^{\eta} \log \sigma(f(\xi'_i) - \gamma), \quad (15)$$

where  $\eta$  is the number of negative training samples over the positive one,  $\xi$  is the positive training quadruple,  $\sigma(\cdot)$  denotes the sigmoid function,  $\gamma$  is a fixed margin, and  $\xi'_i$  denotes the  $i$ -th negative sample generated by randomly corrupting the subject or the object of  $\xi$  such as  $(s', r, o, t)$  and  $(s, r, o', t)$ .

## 3.3 Modeling Various Relation Patterns

In this section, we demonstrate that our RotateQVS can model various relation patterns. In TKGE, four kinds of relation patterns are mostly considered and studied in previous static KGE and TKGE works (Sun et al., 2019; Gao et al., 2020). Their definitions are given as follows:

**Definition 1.** *A relation  $r$  is symmetric, if  $\forall s, o, t, r(s, o, t) \wedge r(o, s, t)$  holds True.*

**Definition 2.** *A relation  $r$  is asymmetric, if  $\forall s, o, t, r(s, o, t) \wedge \neg r(o, s, t)$  holds True.*

**Definition 3.** *Relation  $r_1$  is the inverse of  $r_2$ , if  $\forall s, o, t, r_1(s, o, t) \wedge r_2(o, s, t)$  holds True.*

**Definition 4.** *Relation  $r_1$  and  $r_2$  are evolving over time from  $t_1$  (time stamp  $\tau_1$ ) to  $t_2$  (time stamp  $\tau_2$ ), if  $\forall s, o, r_1(s, o, t_1) \wedge r_2(s, o, t_2)$  holds True.*

Comparing with other TKGE methods, we show RotateQVS can model all these four patterns, while previous methods (see Section 4.3) fail to do so.<sup>2</sup> One advantage of applying quaternion embeddings is that our method supports all these relation patterns, while other representation forms cannot, such as TeRo (Xu et al., 2020a) using complex number system  $a + bi$ .<sup>3</sup>

As seen in our score function (Equation 14), our aim is to make

$$\begin{aligned} \boldsymbol{\tau} \mathbf{s} \boldsymbol{\tau}^{-1} + \mathbf{r} &= \overline{\boldsymbol{\tau} \mathbf{o} \boldsymbol{\tau}^{-1}} = \boldsymbol{\tau} \overline{\mathbf{o}} \boldsymbol{\tau}^{-1} \\ \Leftrightarrow \overline{\mathbf{o}} - \mathbf{s} &= \boldsymbol{\tau}^{-1} \mathbf{r} \boldsymbol{\tau}. \end{aligned} \quad (16)$$

Then we can get following results:

**Lemma 2.** *RotateQVS can model the symmetric pattern for TKG. (See proof in Appendix B)*

**Lemma 3.** *RotateQVS can model the asymmetric pattern for TKG. (See proof in Appendix C)*

**Lemma 4.** *RotateQVS can model the inversion pattern for TKG. (See proof in Appendix D)*

**Lemma 5.** *RotateQVS can model the temporal-evolution pattern for TKG.*

*Proof.* For temporal-evolution pattern,  $r_1(s, o, t_1) \wedge r_2(s, o, t_2)$  in Definition 4 can be expressed as:

$$\begin{cases} \overline{\mathbf{o}} - \mathbf{s} = \boldsymbol{\tau}_1^{-1} \mathbf{r}_1 \boldsymbol{\tau}_1 \\ \overline{\mathbf{o}} - \mathbf{s} = \boldsymbol{\tau}_2^{-1} \mathbf{r}_2 \boldsymbol{\tau}_2 \\ \Leftrightarrow \boldsymbol{\tau}_2 \boldsymbol{\tau}_1^{-1} \mathbf{r}_1 (\boldsymbol{\tau}_2 \boldsymbol{\tau}_1^{-1})^{-1} = \mathbf{r}_2. \end{cases} \quad (17)$$

For the same head entity and tail entity, if a relation  $r_1$  holds at time  $t_1$  (time stamp  $\tau_1$ ) and a relation  $r_2$  holds at time  $t_2$  (time stamp  $\tau_2$ ), we are supposed to get  $\boldsymbol{\tau}_2 \boldsymbol{\tau}_1^{-1} \mathbf{r}_1 (\boldsymbol{\tau}_2 \boldsymbol{\tau}_1^{-1})^{-1} = \mathbf{r}_2$ .

In addition, based on Equation 17, we have

$$\boldsymbol{\tau}_1^{-1} \mathbf{r}_1 \boldsymbol{\tau}_1 = \boldsymbol{\tau}_2^{-1} \mathbf{r}_2 \boldsymbol{\tau}_2. \quad (18)$$

<sup>2</sup>Statistics of several baselines modeling on various relation patterns are summarised in Appendix E.

<sup>3</sup>Theoretical analysis of TeRo’s defect is shown in Section 3.4.

Model	Space Complexity
TransE	$\mathcal{O}(n_e d + n_r d)$
TTransE	$\mathcal{O}(n_e d + n_r d + n_\tau d)$
HyTE	$\mathcal{O}(n_e d + n_r d + n_\tau d)$
TA-DistMult	$\mathcal{O}(n_e d + n_r d + n_{token} d)$
ATiSE	$\mathcal{O}(n_e d + n_r d)$
DE-SimplE	$\mathcal{O}(n_e d + n_r d)$
TeRo	$\mathcal{O}(n_e d + n_r d + n_\tau d)$
RotateQVS	$\mathcal{O}(n_e d + n_r d + n_\tau d)$

Table 1: Space complexity comparison of our RotateQVS with several baselines.

Since we have Theorem 1,  $\tau_1^{-1}\mathbf{r}_1\tau_1$  and  $\tau_2^{-1}\mathbf{r}_2\tau_2$  can be regarded as rotations in quaternion vector space for  $\mathbf{r}_1$  and  $\mathbf{r}_2$ , respectively, which indicates the norm of  $\mathbf{r}_1$  is the same as that of  $\mathbf{r}_2$ . Furthermore, Lemma 1 indicates the rotation mapping keeps the scalar/real part unchanged for a vector. Thus, we can have the following deductions:

$$\begin{cases} \|\mathbf{r}_1\| = \|\mathbf{r}_2\| \\ Re(\mathbf{r}_1) = Re(\mathbf{r}_2). \end{cases} \quad (19)$$

Notice that Equation 19 is a sufficient and unnecessary conclusion of Equation 18.  $\square$

### 3.4 Theoretical Comparison Against TeRo

TeRo (Xu et al., 2020a) is the main baseline for our model. The rotated head entity embedding and tail entity embedding of TeRo in complex number system are  $\mathbf{s} \circ \boldsymbol{\tau}$ , and  $\mathbf{o} \circ \boldsymbol{\tau}$  respectively, where  $\circ$  denotes Hermitian dot product. The translational score function of TeRo  $f(s, r, o, t) = \|\mathbf{s}_t + \mathbf{r} - \overline{\mathbf{o}}_t\|$  is to make

$$\mathbf{s} \circ \boldsymbol{\tau} + \mathbf{r} = \overline{\mathbf{o}} \circ \overline{\boldsymbol{\tau}} = \overline{\boldsymbol{\tau}} \circ \overline{\mathbf{o}} = \overline{\mathbf{o}} \circ \overline{\boldsymbol{\tau}}. \quad (20)$$

And we further prove that TeRo can not model relations with temporal evolution by means of reduction to absurdity.<sup>4</sup>

To this end, taking advantages of quaternion representation, our RotateQVS can deduce further derivation:

$$\begin{aligned} \boldsymbol{\tau} s \boldsymbol{\tau}^{-1} + \mathbf{r} &= \overline{\boldsymbol{\tau} o \boldsymbol{\tau}^{-1}} = \boldsymbol{\tau} \overline{\mathbf{o}} \boldsymbol{\tau}^{-1} \\ \Leftrightarrow \overline{\mathbf{o}} - \mathbf{s} &= \boldsymbol{\tau}^{-1} \mathbf{r} \boldsymbol{\tau}, \end{aligned} \quad (21)$$

where time stamp embeddings and relation embeddings can be particularly extracted to analyse the influence of temporal evolution on relations,

<sup>4</sup>See proof in Appendix F.

Dataset	ICEWS14	ICEWS05-15	YAGO11k	GDELT
Entities	7,128	10,488	10,623	500
Relations	230	251	10	20
Time Stamps	365	4,017	70	366
Train	72,826	386,962	16,408	2,735,685
Validation	8,941	46,275	2,050	341,961
Test	8,963	46,092	2,051	341,961

Table 2: Statistics of four experimented datasets.

since our derivation result is independent with entity embeddings. Above all, we demonstrate that our RotateQVS can model relations with temporal evolution while TeRo cannot.<sup>5</sup>

### 3.5 Complexity Comparison

Table 1 summarizes the space complexities of several baselines and our model.  $n_e$ ,  $n_r$ ,  $n_\tau$  and  $n_{token}$  denote numbers of entities, relations, time stamps, and temporal tokens used in (García-Durán et al., 2018); and  $d$  is the dimension of embeddings. The space complexity of our RotateQVS is  $\mathcal{O}(n_e d + n_r d + n_\tau d)$ , the same as TTransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018) and TeRo (Xu et al., 2020a).

## 4 Experiments

### 4.1 Benchmark Datasets

To evaluate our proposed Quaternion embeddings, we perform link prediction task on four commonly used TKG benchmark datasets, namely ICEWS14, ICEWS05-15 (García-Durán et al., 2018), YAGO11k (Dasgupta et al., 2018) and GDELT (Trivedi et al., 2017).<sup>6</sup> Table 2 summarises the details of the four datasets, where it is easy to find ICEWS14 and ICEWS05-15 have more quantitative relations than the other two datasets.

ICEWS (Lautenschlager et al., 2015) is a repository containing political events with a specific timestamp. ICEWS14 and ICEWS05-15 (García-Durán et al., 2018) are two subsets of ICWES corresponding to facts in 2014 and facts between 2005 and 2015.

YAGO11k (Dasgupta et al., 2018) is a subset of YAGO3 (Mahdisoltani et al., 2015), where time annotations are represented as time intervals. We derive the dataset from HyTE (Dasgupta et al., 2018)

<sup>5</sup>Proof process is shown in Lemma 5, and case based analysis is shown in Section 4.5.2.

<sup>6</sup>GDELT is derived from <https://github.com/BorealisAI/de-simple/tree/master/datasets/gdelt>, and other datasets can be downloaded from <https://github.com/soledad921/ATISE>.

to obtain the same year-level granularity by dropping the month and date information, which results in 70 different time stamps.

For GDELT, we use the subset extracted by Trivedi et al., consisting of the facts from April 1, 2015 to March 31, 2016. We take the same pretreatment of the train, validation and test sets as (Goel et al., 2020), to make the problem into a TKGC rather than an extrapolation problem.

## 4.2 Evaluation Protocol

Link prediction task that aims to infer incomplete time-wise fact with a missing entity  $((s, r, ?, t)$  or  $(?, r, o, t)$ ) is adopted to evaluate the proposed model. During the inference, we follow the same procedure of Xu et al. to generate candidates. For a test sample  $(s, r, o, t)$ , we first generate candidate quadruples set  $C = \{(s, r, \bar{o}, t) : \bar{o} \in \mathcal{E}\} \cup \{(\bar{s}, r, o, t) : \bar{s} \in \mathcal{E}\}$  by replacing  $s$  or  $o$  with all possible entities, and then rank all the quadruples by their scores (Equation 14) under the time-wise filtered settings (Xu et al., 2019; Goel et al., 2020).

The performance is reported on the standard evaluation metrics: the proportion of correct triples ranked in top 1, 3 and 10 (Hits@1, Hits@3, and Hits@10), and Mean Reciprocal Rank (MRR). All the metrics (Hits@1, Hits@3, Hits@10 and MRR) are the higher the better. For all experiments, we report averaged results across 5 runs, and we omit the variance as it is generally low.

## 4.3 Baselines

We compare with both sota static and temporal KGE baselines. For static baselines, we use TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2019), and QuatE (Zhang et al., 2019a). For TKGE methods, we consider TTransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018), TA-DistMult (García-Durán et al., 2018), DE-SimplE (Goel et al., 2020), ATiSE (Xu et al., 2019), and TeRo (Xu et al., 2020a).<sup>7</sup>

Note that TeRo (Xu et al., 2020a) is also based on the idea of rotations, and thus we consider TeRo as a directly baseline. Because our quaternion representation ( $\mathbf{a} + \mathbf{b}\mathbf{i} + \mathbf{c}\mathbf{j} + \mathbf{d}\mathbf{k}$ ) doubles the embedding parameters of TeRo which uses complex representation ( $\mathbf{a} + \mathbf{b}\mathbf{i}$ ), we further adopt two models for fair comparisons: (i) TeRo-Large: TeRo using dou-

ble embedding dimension;<sup>8</sup> (ii) RotateQVS-Small: The proposed RotateQVS with half embedding dimension. By doing so, their parameter complexities can be comparable with TeRo’s.

## 4.4 Results

The experimental results over four TKG datasets are shown in Table 3.<sup>9</sup> Overall, TKGE methods are better than static KGE methods, which shows the effectiveness of modeling temporal information. For the proposed RotateQVS, we observe that our model outperforms all the baseline models over the four datasets across all metrics consistently.<sup>10</sup> To demonstrate the superiority of the proposed quaternion method, we compare our RotateQVS with the direct baseline TeRo (Xu et al., 2020a). For fair comparisons of model sizes, we observe that our RotateQVS outperforms TeRo-Large and RotateQVS-Small outperforms TeRo. This shows our methods with quaternion embeddings makes great improvements, demonstrating our advantages. Specially, we see that our RotateQVS achieves more improvements on ICEWS14 and ICEWS05-15 datasets. We believe this is because these two datasets have much more quantitative relations (see Table 2) and it is also evident our method behaves better on datasets with complex relation patterns.

## 4.5 Analysis and Case Study

To further demonstrate the learned quaternion embeddings and the ability of our model, we perform case studies on multiple relation patterns, through visualization and quantitative analysis on intuitive examples from ICEWS14.

### 4.5.1 Symmetric/Asymmetric/Inversion Patterns

Since symmetric, asymmetric and inversion patterns have been discussed in previous work (Sun et al., 2019; Xu et al., 2020a), we present the case studies of them to Appendix J.

### 4.5.2 Temporal-evolution Pattern

As shown in Lemma 5, if a relation  $r_1$  and a relation  $r_2$  are evolving over time from  $t_1$  (time stamp  $\tau_1$ )

<sup>8</sup>We reuse the original implementation of (Xu et al., 2020a) from <https://github.com/soledad921/ATiSE> and follow the their best setups.

<sup>9</sup>See hyperparameter setup in Appendix G.

<sup>10</sup>We also take time granularity analysis and embedding dimension analysis in Appendix H and I.

<sup>7</sup>See complexity comparison in Appendix 3.5.

Dataset	ICEWS14				ICEWS05-15			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
TransE (Bordes et al., 2013)	0.094	-	0.637	0.280	0.090	-	0.663	0.294
DistMult (Yang et al., 2015)	0.323	-	0.672	0.439	0.337	-	0.691	0.456
RotatE (Sun et al., 2019)	0.291	0.478	0.690	0.418	0.164	0.355	0.595	0.304
QuatE (Zhang et al., 2019a)	0.353	0.530	0.712	0.471	0.370	0.529	0.727	0.482
TTransE (Leblay and Chekol, 2018)	0.074	-	0.601	0.255	0.084	-	0.616	0.271
HyTE (Dasgupta et al., 2018)	0.108	0.416	0.655	0.297	0.116	0.445	0.681	0.316
TA-DistMult (García-Durán et al., 2018)	0.363	-	0.686	0.477	0.346	-	0.728	0.474
ATiSE (Xu et al., 2019)	0.436	<u>0.629</u>	<u>0.750</u>	0.550	0.378	0.606	0.794	0.519
DE-SimplE (Goel et al., 2020)	0.418	0.592	0.725	0.526	0.392	0.578	0.748	0.513
TeRo (Xu et al., 2020a)	0.468	0.621	0.732	0.562	0.469	0.668	0.795	0.586
TeRo-Large	0.432	0.596	0.722	0.534	0.395	0.627	0.800	0.534
RotateQVS-Small (ours)	<u>0.489</u>	0.625	0.737	<u>0.575</u>	<u>0.473</u>	<u>0.685</u>	<u>0.802</u>	<u>0.591</u>
RotateQVS (ours)	<b>0.507</b>	<b>0.642</b>	<b>0.754</b>	<b>0.591</b>	<b>0.529</b>	<b>0.709</b>	<b>0.813</b>	<b>0.633</b>
YAGO11k								
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
TransE	0.015	0.138	0.244	0.100	0.0	0.158	0.312	0.113
DistMult	0.107	0.161	0.268	0.158	0.117	0.208	0.348	0.196
RotatE	0.103	0.167	0.305	0.167	-	-	-	-
QuatE	0.107	0.148	0.270	0.164	-	-	-	-
TTransE	0.020	0.150	0.251	0.108	0.0	0.160	0.318	0.115
HyTE	0.015	0.143	0.272	0.105	0.0	0.165	0.326	0.118
TA-DistMult	0.103	0.171	0.292	0.161	0.124	0.219	0.365	0.206
ATiSE	0.110	0.171	0.288	0.170	-	-	-	-
DE-SimplE	-	-	-	-	0.141	0.248	0.403	0.230
TeRo	<u>0.121</u>	<u>0.197</u>	0.319	<u>0.187</u>	0.154	0.264	0.420	0.245
TeRo-Large	0.094	<b>0.199</b>	<b>0.323</b>	0.181	0.163	<u>0.278</u>	<u>0.437</u>	0.256
RotateQVS-Small	<b>0.124</b>	0.193	0.320	<u>0.187</u>	<u>0.165</u>	0.270	0.428	0.259
RotateQVS	<b>0.124</b>	<b>0.199</b>	<b>0.323</b>	<b>0.189</b>	<b>0.175</b>	<b>0.293</b>	<b>0.458</b>	<b>0.270</b>

Table 3: Results on link prediction task over four experimented datasets. The best score is in **bold** and second best score is underlined.

to  $t_2$  (time stamp  $\tau_2$ ), we have

$$\tau_2 \tau_1^{-1} \mathbf{r}_1 (\tau_2 \tau_1^{-1})^{-1} = \mathbf{r}_2. \quad (22)$$

To analyse the temporal-evolution pattern, we focus on the relations between the same head and tail entities with different time stamps. For example, from ICEWS14, we observe a base fact (*South Korea, Engage in negotiation, North Korea, 2014-02-12*) and a temporal-evolution fact (*South Korea, Sign formal agreement, North Korea, 2014-02-15*), where *Sign formal agreement* is considered as the consequence of *Engage in negotiation*. Thus, in our model, the embeddings of *Sign formal agreement* at time stamp 2014-02-15 and of *Engage in negotiation* at 2014-02-12 should satisfy Equation 22.

To illustrate this pattern, we measure the matrix cosine similarity between  $\mathbf{r}_2$  (base) and  $\tau_2 \tau_1^{-1} \mathbf{r}_1 (\tau_2 \tau_1^{-1})^{-1}$  (temporal-evolved). For each true fact, we sample a random negative relation and show their similarity difference. Figure 2 illus-

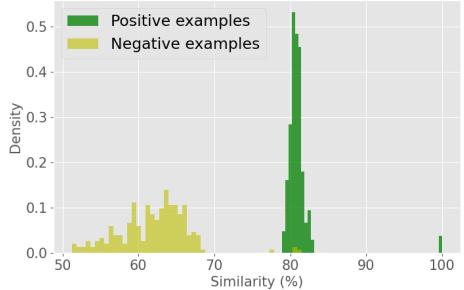


Figure 2: Density histogram with bin size 1% of similarity scores for temporal-evolution relations. All positive and negative examples are randomly sampled and compared with the base relation *Engage in negotiation*.

trates the density histogram of similarities with random 250 fact quadruples at different time stamps between *South Korea* and *North Korea*. We observe that the distributions of positive examples and negative examples are distinct, which explains

	Head entity	Relation	Tail entity	Time	Similarity
Example 1	John Kerry	Express intent to meet or negotiate	Pietro Parolin	2014-01-13	0.810
		Consult		2014-01-16	0.508
		Detonate nuclear weapons			
Example 2	Member of Legislative (Govt) (Iran)	Make statement	Iran	2014-03-16	0.819
		Make statement		2014-05-04	0.492
		Detonate nuclear weapons			
Example 3	Federal Bank	Make a visit	European Central Bank	2014-02-04	0.815
		Make statement		2014-02-25	0.510
		Receive inspectors			

Table 4: Examples of temporal-evolution patterns in ICEWS14 dataset. The similarity score is based on base fact.

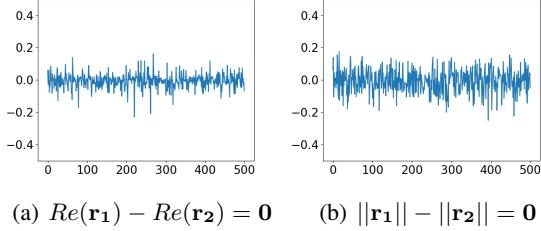


Figure 3: Quaternion representations of Equation 19 for temporal-evolution pattern.

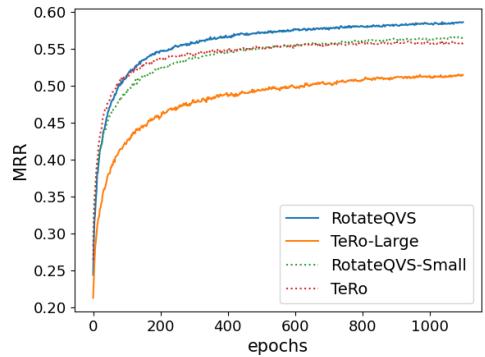


Figure 4: The convergence study of RotateQVS, TeRo-Large, RotateQVS-Small and TeRo by epochs on ICEWS14 test set, and we use the metric MRR here.

our RotateQVS can model temporal-evolution patterns more effectively. Comparing with TeRo (Xu et al., 2020a), which is the main baseline for our model, we show TeRo cannot model this pattern theoretically (see Section 3.4).

In addition, Figure 3 shows our quaternion representation do well in reflecting Equation 19, the sufficient and unnecessary deductions of theoretical analysis for temporal-evolution pattern.

More examples of temporal-evolution pattern are shown in Table 4, where we use the relation in base fact and time information to get a generated embedding  $\tau_2 \tau_1^{-1} \mathbf{r}_1 (\tau_2 \tau_1^{-1})^{-1}$ , and also sample a random negative relation for each example. We compute the matrix cosine similarity between  $\tau_2 \tau_1^{-1} \mathbf{r}_1 (\tau_2 \tau_1^{-1})^{-1}$  and  $\mathbf{r}_2$ , and also compute the similarity between  $\tau_2 \tau_1^{-1} \mathbf{r}_1 (\tau_2 \tau_1^{-1})^{-1}$  and the embedding of another relation in the negative sample. Time stamps in negatives are taken as same as the true facts. The comparison between the two sets of results can once again prove the ability of our model in modeling this pattern.

#### 4.6 Convergence Analysis

For convergence analysis, we consider two fair comparisons, where the compared two methods have the same number of parameters:<sup>11</sup> RotateQVS (blue solid) vs. TeRo-Large (yellow solid) and RotateQVS-Small (green dotted) vs.

TeRo (red dotted) in Figure 4. We observe that RotateQVS and TeRo-Large converge at approximately the same rate, and so do RotateQVS-Small and TeRo. We can conclude that our proposed RotateQVS can achieve better results in comparisons of both large and small levels without sacrificing additional training efforts.

## 5 Related work

Models working on Static Knowledge graph have been well studied (Zhang et al., 2019b; Xu et al., 2020b; Mao et al., 2020; Chen et al., 2021a) with semantic and structure information. Translation based methods, e.g. TransE (Bordes et al., 2013) and TransR (Lin et al., 2015), formalise the factual distance between a head entity  $s$  and a tail entity  $o$  with the translation carried out by the relation. Adopting tensor factorization with a bilinear transformation, semantic matching models, e.g. RESCAL (Nickel and Tresp, 2013) and DistMult (Yang et al., 2015), capture the semantic relevance of entities. Recently, more attention were paid to study various relation patterns. RotatE (Sun et al., 2019) treat each relation as a rotation so that symmetric/asymmetric, inversion and composition patterns can be inferred to predict missing links. Further, quaternion number system (Hamil-

<sup>11</sup>Refer to Section 4.3 for more details

ton, 1844) is applied to model more complex composition patterns in 3D space, such as Rotate3D (Gao et al., 2020) and QuatE (Zhang et al., 2019a).

Many aforementioned methods (Dasgupta et al., 2018; Leblay and Chekol, 2018; Trivedi et al., 2017; García-Durán et al., 2018; Goel et al., 2020; Sadeghian et al., 2021) are extended from static Static KGs to TKGs. They integrate time information into previous static methods as independent features. Others study the dynamic evolution of TKG. ATiSE (Xu et al., 2019) regards the temporal evolution of entity and relation embeddings as combinations of trend component, seasonal component and random component. CyGNet (Zhu et al., 2021) proposes a time-aware copy-generation mechanism leveraging known facts in the past to predict unknown facts in the future. TeRo (Xu et al., 2020a) defines the temporal evolution of entity embedding as a rotation in the complex vector space. Inspired by TeRo, our RotateQVS further represents temporal entities as rotations in quaternion vector space and obtains more advantages.<sup>12</sup>

Modeling various temporal relation patterns (Goel et al., 2020; Xu et al., 2020a), especially the temporal-evolution patterns, is crucial for TKGE and the following TKGC. Zhang et al. mentions the time-evolution property, but does not make a systematic research on it. It remains an open research question with few researches. Our work (RotateQVS) takes inspirations from the idea of rotation and generalises it into the quaternion number system to model the complex temporal-evolution pattern that TeRo can hardly do.

## 6 Conclusion

In this paper, we introduce a novel TKGC method RotateQVS which represents temporal information of TKGs as rotations in quaternion vector space. Targeting temporal interpretability, we theoretically analyse that RotateQVS can model various relation patterns and demonstrate it with extensive experiments. Compared to previous methods, RotateQVS has made significant improvements on link prediction tasks over four benchmark datasets. Furthermore, we show our RotateQVS has great advantages in modeling various relation patterns with temporal evolution.

<sup>12</sup>Refer to Section 3.4 for more details.

## Acknowledgements

This work is supported by the Key R&D Program of Guangdong Province (No.2019B010136003), the National Natural Science Foundation of China (No. 61732004, 61732022).

## References

- Kurt D. 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 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008*, pages 1247–1250.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.
- Kai Chen, Ye Wang, Aiping Li, Chenchen Li, Xiaojuan Zhao, and Ruidong Ding. 2021a. Learning knowledge graph embedding in semantic space: A novel bi-linear semantic matching method. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE.
- Kai Chen, Ye Wang, Yitong Li, Aiping Li, and Xiaojuan Zhao. 2021b. Contextualise entities and relations: An interaction method for knowledge graph completion. In *Artificial Neural Networks and Machine Learning - ICANN 2021 - 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14-17, 2021, Proceedings, Part III*, volume 12893 of *Lecture Notes in Computer Science*, pages 179–191. Springer.
- 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*, pages 2001–2011.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1811–1818.
- Leonhard Euler. 1776. Novi commentarii academiae scientiarum petropolitanae. *Nr*, 20:189–207.
- Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 385–394.

- Alberto García-Durán, Sebastijan Dumancic, 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821. Association for Computational Linguistics.
- Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3988–3995.
- Sir William Rowan Hamilton. 1844. II. On quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13.
- Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. 2016. Towards time-aware knowledge graph completion. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1715–1724. ACL.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683.
- Jennifer Lautenschlager, Steve Shellman, and Michael Ward. 2015. Icews events and aggregations. *Harvard Dataverse*, 3.
- Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776.
- Kalev Leetaru and Philip A Schrod. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2181–2187. AAAI Press.
- Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research*.
- Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM ’20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 420–428. ACM.
- George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Maximilian Nickel and Volker Tresp. 2013. Tensor factorization for multi-relational learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 617–621. Springer.
- Olinde Rodrigues. 1840. *Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace: et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire*.
- Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6471–6479. AAAI Press.
- Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018*, volume 10843, pages 593–607.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW ’07*, page 697–706, New York, NY, USA. Association for Computing Machinery.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha P. Talukdar, and Yiming Yang. 2020. A re-evaluation of knowledge graph completion methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5516–5522. Association for Computational Linguistics.
- Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3462–3471. PMLR.

T Verhoeff. 2014. Euler’s rotation theorem.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Jens Lehmann, and Hamed Shariat Yazdi. 2019. Temporal knowledge graph embedding model based on additive time series decomposition. *CoRR*, abs/1911.07893.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020a. TeRo: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1583–1593, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020b. Product knowledge graph embedding for e-commerce. In *WSDM ’20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 672–680. ACM.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015*.

Jiasheng Zhang, Shuang Liang, Zhiyi Deng, and Jie Shao. 2021. Spatial-temporal attention network for temporal knowledge graph completion. In *Database Systems for Advanced Applications - 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11-14, 2021, Proceedings, Part I*, volume 12681 of *Lecture Notes in Computer Science*, pages 207–223. Springer.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pages 2735–2745.

Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019b. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 96–104. ACM.

Cunchao Zhu, Muhaoo Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4732–4740. AAAI Press.

## A Proof of Theorem 1

*Proof.* Based on Equation 2, for a unit quaternion  $q$ , it follows that  $q^{-1} = \bar{q}$ . Unfolding the Equation 11, we can get

$$\begin{aligned}\mathbf{v}' &= (\cos \frac{\theta}{2} + \mathbf{u} \sin \frac{\theta}{2})\mathbf{v}(\cos \frac{\theta}{2} - \mathbf{u} \sin \frac{\theta}{2}) \\ &= \mathbf{v} \cos^2 \frac{\theta}{2} + (\mathbf{u} \mathbf{v} - \mathbf{v} \mathbf{u}) \sin \frac{\theta}{2} \cos \frac{\theta}{2} \\ &\quad - \mathbf{u} \mathbf{v} \sin^2 \frac{\theta}{2}.\end{aligned}\quad (23)$$

Bearing in mind that  $\mathbf{u} \mathbf{v} = \mathbf{u} \times \mathbf{v} - \mathbf{u} \cdot \mathbf{v}$  (based on the Equation 6), further we can get

$$\begin{aligned}\mathbf{v}' &= \mathbf{v} \cos^2 \frac{\theta}{2} + 2(\mathbf{u} \times \mathbf{v}) \sin \frac{\theta}{2} \cos \frac{\theta}{2} \\ &\quad - ((\mathbf{u} \times \mathbf{v}) - (\mathbf{u} \cdot \mathbf{v})) \mathbf{u} \sin^2 \frac{\theta}{2} \\ &= \mathbf{v} \left( \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} \right) + (\mathbf{u} \times \mathbf{v}) (2 \sin \frac{\theta}{2} \cos \frac{\theta}{2}) \\ &\quad + \mathbf{u} (\mathbf{u} \cdot \mathbf{v}) (2 \sin^2 \frac{\theta}{2}).\end{aligned}\quad (24)$$

Using trigonometric identities, we can get

$$\begin{aligned}\mathbf{v}' &= \mathbf{v} \cos \theta + (\mathbf{u} \times \mathbf{v}) \sin \theta \\ &\quad + \mathbf{u} (\mathbf{u} \cdot \mathbf{v}) (1 - \cos \theta) \\ &= (\mathbf{v} - \mathbf{u} (\mathbf{u} \cdot \mathbf{v})) \cos \theta + (\mathbf{u} \times \mathbf{v}) \sin \theta \\ &\quad + \mathbf{u} (\mathbf{u} \cdot \mathbf{v}) \\ &= \mathbf{v}_\perp \cos \theta + (\mathbf{u} \times \mathbf{v}) \sin \theta + \mathbf{v}_\parallel\end{aligned}\quad (25)$$

where  $\mathbf{v}_\perp = \mathbf{v} - \mathbf{u} (\mathbf{u} \cdot \mathbf{v})$  and  $\mathbf{v}_\parallel = \mathbf{u} (\mathbf{u} \cdot \mathbf{v})$  are the components of  $\mathbf{v}$  (perpendicular and parallel to the axis  $\mathbf{u}$  respectively). Our Equation 25 satisfies the Rodrigues' rotation formula (Rodrigues, 1840) in 3D vector space (illustrated in Figure 5). Therefore, the Equation 11 is proven to be a rotation in 3D vector space.  $\square$

## B Proof of Lemma 2

*Proof.* For symmetric pattern, considering our rotation based temporal representation, we express the  $r(s, o, t) \wedge r(o, s, t)$  in Definition 1 as:

$$\begin{cases} \bar{\mathbf{o}} - \mathbf{s} = \tau^{-1} \mathbf{r} \tau \\ \bar{\mathbf{s}} - \mathbf{o} = \tau^{-1} \mathbf{r} \tau \end{cases} \Leftrightarrow \mathbf{r} + \bar{\mathbf{r}} = 0 \Leftrightarrow Re(\mathbf{r}) = 0,\quad (26)$$

where  $Re$  denotes the real part of a quaternion representation.  $\square$

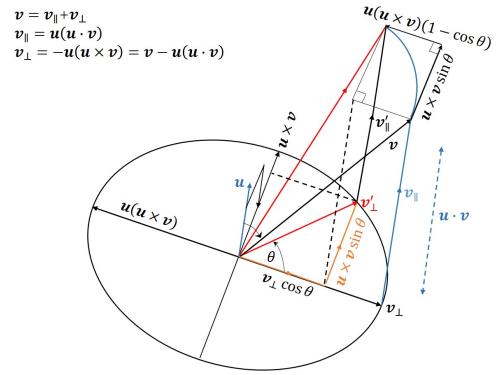


Figure 5: An illustration of our rotation equation, which satisfies the Rodrigues' rotation formula (Rodrigues, 1840), where  $\mathbf{v}'$  is the result of vector  $\mathbf{v}$  rotating  $\theta$  around rotation axis  $\mathbf{u}$ .

## C Proof of Lemma 3

*Proof.* For asymmetric pattern,  $r(s, o, t) \wedge \neg r(o, s, t)$  in Definition 2 can be expressed as:

$$\begin{cases} \bar{\mathbf{o}} - \mathbf{s} = \tau^{-1} \mathbf{r} \tau \\ \bar{\mathbf{s}} - \mathbf{o} \neq \tau^{-1} \mathbf{r} \tau \end{cases} \Leftrightarrow \mathbf{r} + \bar{\mathbf{r}} \neq 0 \Leftrightarrow Re(\mathbf{r}) \neq 0.\quad (27)$$

$\square$

## D Proof of Lemma 4

*Proof.* For inversion pattern,  $r_1(s, o, t) \wedge r_2(o, s, t)$  in Definition 3 can be expressed as:

$$\begin{cases} \bar{\mathbf{o}} - \mathbf{s} = \tau^{-1} \mathbf{r}_1 \tau \\ \bar{\mathbf{s}} - \mathbf{o} = \tau^{-1} \mathbf{r}_2 \tau \end{cases} \Leftrightarrow \mathbf{r}_1 + \bar{\mathbf{r}_2} = 0 \Leftrightarrow \begin{cases} Re(\mathbf{r}_1) + Re(\mathbf{r}_2) = 0 \\ Im(\mathbf{r}_1) - Im(\mathbf{r}_2) = 0 \end{cases},\quad (28)$$

where  $Im$  denotes the imaginary part of a quaternion representation.  $\square$

## E Statistics of several previous TKGE methods modeling on various relation patterns

Table 5 shows the statistics of several previous TKGE methods modeling on various relation patterns, containing symmetry, asymmetry, inversion and temporal-evolution.

## F Proof by Contradiction for TeRo

*Proof.* Supposing TeRo (Xu et al., 2020a) can model the temporal-evolution relation pattern (defined in Definition 4), then relations with temporal-

Methods	Symmetry	Asymmetry	Inversion	Temporal-evolution
TTransE	✗	✓	✓	✗
TA-DistMult	✓	✗	✗	✗
DE-SimplE	✓	✓	✓	✗
TeRo	✓	✓	✓	✗
RotateQVS (ours)	✓	✓	✓	✓

Table 5: Statistics of several previous TKGE methods modeling on various relation patterns.

evolution pattern will satisfy the Equation 20. Notice that our relation patterns defined are unconcerned with some specific entities, but focusing on the general rules among relations inside the universal entities.

If relation  $r_1$  and  $r_2$  are evolving over time from  $t_1$  to  $t_2$ , considering the same head entity  $s$  and two different tail entities  $o_1$  and  $o_2$  which satisfy  $r_1(s, o_1, t_1) \wedge r_2(s, o_1, t_2)$  and  $r_1(s, o_2, t_1) \wedge r_2(s, o_2, t_2)$ , we can get

$$\begin{cases} s \circ \tau_1 + r_1 = \bar{o}_1 \circ \bar{\tau}_1 \\ s \circ \tau_2 + r_2 = \bar{o}_1 \circ \bar{\tau}_2 \end{cases} \wedge \begin{cases} s \circ \tau_1 + r_1 = \bar{o}_2 \circ \bar{\tau}_1 \\ s \circ \tau_2 + r_2 = \bar{o}_2 \circ \bar{\tau}_2 \end{cases}, \quad (29)$$

where we can find the derivations depend on entity embeddings. And we can further derive that

$$\begin{aligned} & \begin{cases} s \circ (\tau_2 - \tau_1) + (r_2 - r_1) = \bar{o}_1 \circ (\bar{\tau}_2 - \bar{\tau}_1) \\ s \circ (\tau_2 - \tau_1) + (r_2 - r_1) = \bar{o}_2 \circ (\bar{\tau}_2 - \bar{\tau}_1) \end{cases} \\ & \Leftrightarrow o_1 = o_2, \end{aligned} \quad (30)$$

where two different tail entities  $o_1$  and  $o_2$  have the exactly same embeddings in TeRo. Obviously, it is not in line with our common sense and has a big problem in modelling the temporal-evolution relation pattern.  $\square$

## G Hyperparameter

To seek and find proper hyperparameters, we utilize a grid search empirically over the following ranges for all three datasets: embedding dimension in  $\{100, 200, 300, 400, 500\}$ , learning rate in  $\{1, 0.5, 0.3, 0.1, 0.05, 0.03, 0.02, 0.01\}$ , the ratio of negative over positive training sample in  $\{1, 3, 5, 10\}$ , the margin used in loss function in  $\{1, 2, 3, 5, 10, 20, \dots, 120\}$ , the time granularity parameter in  $\{1, 2\}$ , and the optimizer we use is Adagrad.

And we have found out the best hyperparameters combination as follows: for ICEWS14, we set the margin as 110, the time granularity parameter as 1; for ICEWS05-15, we set the margin as 120, the time granularity parameter as 2; for YAGO11k, we set the margin as 50, the time granularity parameter as 100; for GDELT, we set the margin as 110, the

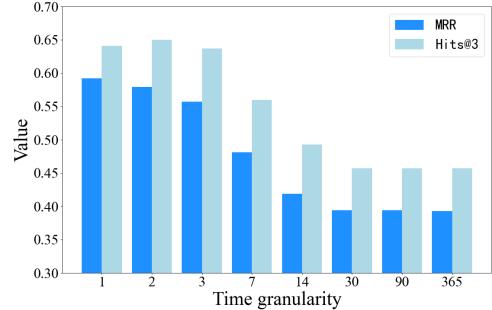


Figure 6: Results of RotateQVS with different time granularities on ICEWS14 dataset.

time granularity parameter as 1; and for all the datasets, we choose the learning rate as 0.1, the embedding dimension as 500, the ratio of negative over positive training sample as 10.

## H Time Granularity Analysis

As shown in Figure 6, we take time granularity analysis on ICEWS14 dataset. It find that the results of smaller granularities are better than that of larger granularities, as larger-granularity setups fuzz the time information. Smaller granularity means more time stamps to compute, while we believe in current dataset the number of time-stamps are relatively small compared with the numbers of relations and entities, and thus we suggest small time granularity in TKG tasks.

## I Size of Embedding Dimension

As shown in Figure 7, we take embedding dimension analysis on ICEWS14 dataset. We can find out that the values of all the four metrics increase as the dimension increases from 100 to 500, while the improvement gains decrease when approaching the size of 500. This indicates that larger embedding size are recommended, while larger embeddings can drag time efficiency and requires more computational resources, thus it is a time-efficiency trade-off.

## J Analysis and Case Study for Symmetric/Asymmetric/Inversion Pattern

### J.1 Symmetric Pattern

As shown in Lemma 2 and its proof (see Appendix B), if  $r$  is a symmetric relation, we have  $r + \bar{r} = \mathbf{0} \Leftrightarrow Re(r) = \mathbf{0}$ . That is if  $r$  is a symmetric relation, the real part of quaternion representa-

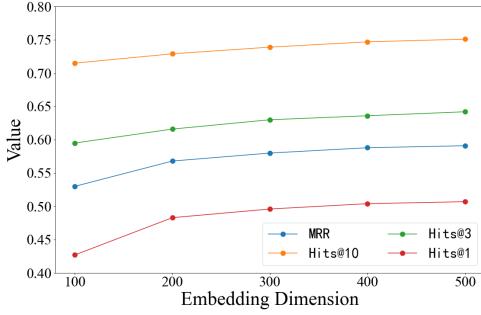


Figure 7: Results of RotateQVS with different embedding dimensions on ICEWS14 dataset.

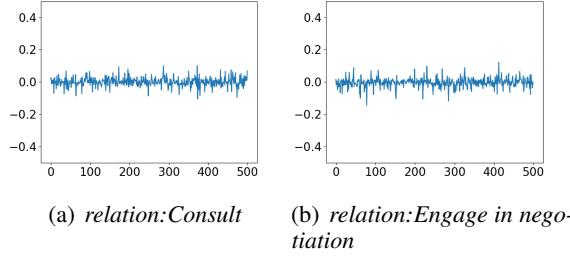


Figure 8: Illustrations of the real parts ( $Re$ ) closing to 0 for two symmetric relations, *Consult* and *Engage in negotiation*, in quaternion embeddings with size 500.

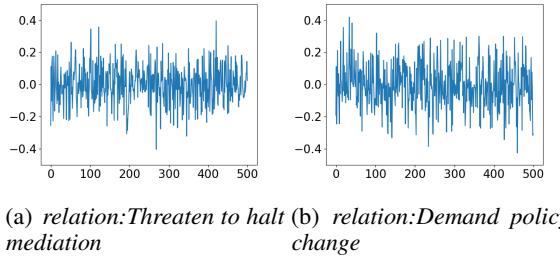


Figure 9: Illustrations of the real parts  $Re(r) \neq 0$  for two asymmetric relations in quaternion embeddings.

tion of  $r$  is close to zero. To empirically study the learned quaternion embedding of  $r$ , we illustrate the real parts of quaternion embeddings in Figure 8 for two symmetric relations, *Consult* and *Engage in negotiation*. For *Consult*, we have (*France, Consult, Canada, 2014-10-23*) and (*Canada, Consult, France, 2014-10-23*). For *Engage in negotiation*, we have (*Victor Ponta, Engage in negotiation, Klaus Johannis, 2014-11-11*) and (*Klaus Johannis, Engage in negotiation, Victor Ponta, 2014-11-11*). These suggest that the relation *Consult* and the relation *Engage in negotiation* are two symmetric relations. We observe that the learned quaternion embeddings in Figure 8 are close to 0, which con-

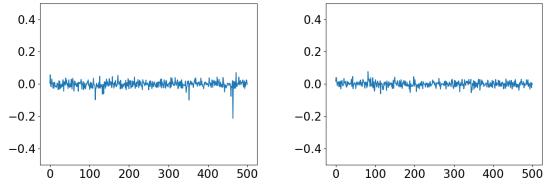


Figure 10: Quaternion representations of Equation 28, with the real part ( $Re$ ) and three imaginary parts ( $i$ ,  $j$ , and  $k$ ) for an inverse relation pair: *Make a visit* vs. *Host a visit*.

firms the ability of our model.

## J.2 Asymmetric Pattern

Opposite to symmetric pattern, if  $r$  is an asymmetric relation, we have  $Re(r) \neq 0$ . Intuitively, if  $r$  is an asymmetric relation, the real part of quaternion representation of  $r$  is supposed to be far away from zero. Since we have (*Nabih Berri, Threaten to halt mediation, Israeli Defense Forces, 2014-05-12*) and (*Islamic Preacher (Iran), Demand policy change, Iran, 2014-03-02*), the two relations *Threaten to halt mediation* and *Engage in negotiation* are taken as two asymmetric relations. Figure 9 illustrates the real parts of quaternion representation of them. These observations from Figure 8 and Figure 9 show that our RotateQVS can effectively model the symmetry and asymmetry patterns and can distinguish them.

## J.3 Inversion Pattern

Lemma 4 and its proof (see Appendix D) show that if the relation  $r_1$  is the inverse of the relation  $r_2$ , we have  $Re(r_1) + Re(r_2) = \mathbf{0}$  and  $Im(r_1) - Im(r_2) = \mathbf{0}$ . From two existing facts (*Romania, Host a visit, Evangelos Venizelos, 2014-02-20*) and (*Evangelos Venizelos, Make a visit, Romania, 2014-02-20*) in ICEWS14, we can find out the relation *Host a visit* is the inverse of the relation *Make a visit*, which satisfies the Definition 4. Figure 10 shows the quaternion representation as in

Equation 28 for the inverse relation pair, containing the real part and the imaginary part (3D), where  $i, j$  and  $k$  denote three directions of the imaginary parts. This demonstrates our method has effective modeling for inverse relations.