# Combination of Translation and Rotation in Dual Quaternion Space for Temporal Knowledge Graph Completion

Ruiguo Yu[1,2,3], Tao Liu[1,2,3], Jian Yu[1,2,3], Wenbin Zhang[1,2,4], Yue Zhao[4], Ming Yang[5], Mankun Zhao *[1,2,3], Jiujiang Guo[1,2,3]

[1]*College of Intelligence and Computing, Tianjin University, Tianjin, China*
[2]*Tianjin Key Laboratory of Advanced Networking(TANK Lab), Tianjin, China*
[3]*Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China*
[4]*Information and Network Center, Tianjin University, Tianjin, China*
[5]*College of Computing and Software Engineering, Kennesaw State University, Marietta, GA, USA*
{Rgyu, liu_ tao83, yujian, zhangwenbin, zhao_yue, zmk}@tju.edu.cn, ming.yang@kennesaw.edu, www.353745517@qq.com

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191552

***Abstract*—Compared with static knowledge graphs (KGs) temporal KGs record the dynamic relations between entities over time, therefore, research on temporal Knowledge Graph Completion (KGC) attracts much attention. Temporal KGs exhibit complex temporal relation patterns, such as multiple relations. However, existing methods can hardly model all the relation patterns and apply to the temporal KGs. In this paper, we propose a novel temporal KGC method that Combining Translation and Rotation (ComTR) in Dual Quaternion Space for temporal KGC. Specifically, we use dual-quaternion-based multiplication to model timestamps and relations as the combination of translation and rotation operations. We analyze the relation patterns of temporal KGs in detail and demonstrate that our method can model all the relation patterns in temporal KGs. Empirically, we show that ComTR can achieve the state-of-the-art performances over four temporal KGC benchmarks datasets.**

***Index Terms*—temporal knowledge graph, Knowledge Graph Completion, dual quaternion, translation, rot**

## I. Introduction

Knowledge Graphs (KGs), as a structured semantic knowledge base, can describe entities, events, concepts and their interrelationships in the real-world. KGs use triples, denoted as $(s, r, o)$, to store real-world factual information, where $s$ is the head entity, $r$ is the relation, and $o$ is the tail entity. KGs such as DBPedia [1], Freebase [2], and WordNet [3] play an increasingly prominent role in representing, storing, and processing information, and have been widely used in many applications, such as question answering [4], recommendation system [5], and reasoning [6].

As known facts are usually sparse, KGs are incomplete. To address this problem, knowledge graph completion (KGC) methods learn the embedding of entities and relations in a low-dimensional embedding space and measure the plausibility of triples by computing the embeddings of entities and their relations with a score function. However, standard KGC methods are static, i.e., the ground truth of a fact is independent of time. But some KGs involve temporal facts, e.g., the fact *(Barack Obama, presidentOf, USA)* is only valid for a specific time period [2009, 2017]. Temporal KGs such as Wikidata [7], YAGO3 [8] and ICEWS [9] incorporate temporal information into triples. In temporal KGs, temporal facts are represented as a quadruple, denoted as $(s, r, o, \tau)$, by extending the static triple, where $\tau$ represents the timestamp. The traditional KGC methods ignore temporal information, which leads to the invalidity of KGC on temporal KGs involving temporal relations, such as *( ?, presidentOf, USA, 2010 )*. Compared with the static KGs, temporal KGs are closer to the real-world, but temporal KGC is a more difficult task due to data heterogeneity and its complex temporal dependencies.

At present, some works add timestamp hidden representation based on the static KGC methods and extend score functions to construct temporal KGC methods. These methods are mostly transformed from translation and rotation methods commonly used in static KGs. However, a single translation or rotation method cannot model all relations patterns, especially multiple relations. The multiple relations pattern is the pattern in which there are multiple relations between two entities (e.g., *(Barack Obama, Make a visit, France)*, *( Barack Obama, Praise or endorse, France)*). Translation methods cannot model many relation patterns including multiple relations. Rotation methods use circular rotation in Euclidean, and hence they can model (anti)symmetry, inversion, and composition patterns. But rotation methods are also unable to model multiple relations. The shortcomings of these methods are magnified in temporal KGs. Temporal KGs add the time dimension to the KGs, which increases the complexity of KGs and adds many multiple relations to KGs. For example, Barack

*Corresponding author: Mankun Zhao (zmk@tju.edu.cn).

Obama make a visit to France on 2014-06-05, 2012-02-12 and 2013-03-28 respectively, which formed three quadruples of multiple relations. To overcome this shortcoming, recent methods model KGs into hyperbolic space [10], [11], but hyperbolic methods inevitably lose the fundamental property of Euclidean space transformation.

In this paper, we propose a novel temporal KGC method ComTR, which **Com**bine **T**ranslation and **R**otation in dual quaternion vector space for temporal KGC. We transcend the translation and rotation representation, and introduce a more expressive dual quaternion representation. Specifically, we represent the entities, relations, and timestamps in the temporal KGs as dual quaternions vector. The multiplication between dual quaternions can combine rotation and translation together, which has a strong geometric interpretation. This method can model all the relation patterns and can choose the best representation for each relation. Our contributions are as follows:

- We are the first to use dual quaternions algebra for temporal KGC. We designed a flexible score function that combines translation and rotation and leverages multiple geometric transformations to represent temporal relations.
- Theoretically, we analyze the relation patterns of temporal KGs and the superiority and completeness of ComTR.
- Empirically, experimental results on four most common temporal knowledge graph datasets show that ComTR is superior to the state-of-the-art temporal KGC methods.

## II. RELATED WORK

In this section, we introduce the existing methods of KGC and temporal KGC.

### A. Static KGC methods

There has been a lot of works on static KGC . The static KGC methods represent entities and relations into low-dimensional vectors and learn these relations from the data to calculate the score of triples. Static KGC methods can be roughly divided into three families. **Translation** based methods, e.g. TransE [12] represents the relation as a translation operation and considers that the head entity $s$, relation $r$ and tail entity $o$ satisfied $s + r = o$. TransE has several variants, including TransH [13], which represents relations as hyperplanes; TransR [14] which represents entities and relations into separate spaces, and TransD [15] represents each element in the triples with two vectors and constructs the mapping matrix. Translation family cannot capture most relation patterns. **Rotation** based methods, e.g. DistMult [16] and ComplEx [17] extend the embedding space to the complex space for the first time. RotatE [18] describes the relation as rotation from the head to the tail by performing rotation operations in a complex space. QuatE [19] extends the complex space into a quaternion space with two surfaces of revolution. However, rotation methods cannot model hierarchies, nor can it model multiple relations between two entities simultaneously. **Neural network** methods such as R-GCN [20], ConvE [21], and CompGCN [22] use neural network architectures to score knowledge graph embeddings. In recent years, KGC has been widely studied, but most of them focus on static KGs without considering the time information.

### B. Temporal KGC methods

The above methods have achieved excellent results in static KGC. Similar to the static KGC, temporal KGC methods also use embedding vectors to represent entities, relations, and timestamps in temporal KGs and then construct a score function to score the facts [23]. Therefore, most temporal KGC methods are based on existing static KGC methods. For example, TTransE [24] is a temporal extension of TransE, representing timestamps as translations. HYTE [25] is a temporal extension of TransH that represents timestamps as a learned hyperplane and represents entities and relations to a time-specific hyperplane. Based on ComplEx, TComplEx [26] and TNTComplEx represent temporal KGs as fourth-order tensors using matrix factorization methods. TComplEx applies this decomposition directly, while TNTComplEx splits this decomposition into a temporal and a non-temporal component. ATISE [27] represents entities and relations as time-sensitive multi-dimensional Gaussian distributions, representing temporal uncertainty as to the covariance of the Gaussian distribution. TERO [28] combines TransE and RotatE to define the relation as translation and the timestamp as rotation. Based on RotatE, ChronoR [29] learns a k-dimensional rotation transformation parametrized by relation and time, such that after each fact's head entity is transformed using the rotation, it falls near its corresponding tail entity. TeLM [30] uses multi-vector embeddings to perform fourth-order tensor decomposition on temporal KGs. RotateQVS [31] represents temporal information as rotations in quaternion vector space. DYERNIE [10] introduced hyperbolic methods into temporal KGs, embedding entities and relations into a time-specific hyperbolic space. HERCULES [11] is a temporal extension of ATTH, which defines the curvature of the Riemannian manifold as the product of relation and time. The introduction of manifold space has played a great role in the representation of hierarchy, but optimization in manifold space is a challenge.

Some works introduce neural network sequence methods into temporal KGC. TeMP [32] utilizes a message-passing graph neural network (MPNN) to learn a structure-based entity representation at each timestamp and then uses an encoder to aggregate the representations of all timestamps to generate TeMP-GRU and TeMP-SA. T-GAP [33] encodes the query-specific substructure of TKG by paying attention to the temporal displacement between each event and the query timestamp and performing path-based reasoning by propagating attention in the graph. xERTE [34] uses temporal relation attention and subgraph iteration to build a KGC framework that can reason about query-related subgraphs of temporal KGs.

## III. PRELIMINARIES

In this section, we introduce temporal KGC and the basic concepts of quaternions and dual quaternions.

## *A. Temporal KGC*

In the temporal KGC problem, we define head entities, relations, tail entities and timestamps as $s, r, o, \tau$. We consider a quadruple $(s, r, o, \tau) \in \mathcal{S} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ are the set of entities, relations and timestamps respectively, and $\mathcal{S}$ is the set of correct quadruples. Timestamp $\tau$ has multiple cases, such as time period $[t_b, t_e]$, missing beginning time period $[-, t_e]$, missing ending time period $[t_b, -]$ and time point $t$.

**Temporal knowledge graph** $\mathcal{G}$ consists of a set of quadruples. Each quadruple represents a fact in the real-world.

**Temporal KGC** is based on the existing facts in the temporal knowledge graph $\mathcal{G}$ to predict new facts on the $(\mathcal{E}, \mathcal{R}, \mathcal{E}, \mathcal{T})$. In this work, we focus on the study of missing facts in the temporal set $\mathcal{T}$ rather than predictions of future facts.

## *B. Quaternions and Dual Quaternions*

**Quaternion** [35] is a representative of the hypercomplex number system, extending the traditional complex number system to four-dimensional space. Quaternion $Q$ is defined as the sum of one real component and three imaginary components, that is, $Q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, where $a, b, c, d \in \mathbb{R}$ , $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. The following introduces some common operations of quaternion algebra:

**Conjugate**: The conjugate quaternion of the quaternion $Q$ is $\bar{Q} = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$.

**Norm**: The norm of quaternion $Q$ is $\|Q\| = \sqrt{a^2 + b^2 + c^2 + d^2}$.

**Inner Product**: The inner product of quaternion $Q_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$ and quaternion $Q_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$ is obtained by adding the inner product of the corresponding vector component. The calculation is as follows:

$$Q_1 \cdot Q_2 = \langle a_1, a_2 \rangle + \langle b_1, b_2 \rangle + \langle c_1, c_2 \rangle + \langle d_1, d_2 \rangle , \qquad (1)$$

where $\langle \cdot, \cdot \rangle$ is represents vector inner product.

**Dual quaternions** is an extension of Clifford's [36] theory of dual numbers, intended to be combined with Hamilton's quaternion algebra. Each dual quaternion consists of an octonion or two quaternions. These two quaternion elements are called the real part and the dual part. These two quaternion elements are called the real part and the dual part. Dual quaternion $Q = q_r + \varepsilon q_d$, where $q_r$ and $q_d$ are quaternions: $q_r = a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}$, $q_d = a_d + b_d\mathbf{i} + c_d\mathbf{j} + d_d\mathbf{k}$. Dual quaternions can also be expressed as: $Q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = (a_r + \varepsilon a_d) + (b_r + \varepsilon b_d)\,\mathbf{i} + (c_r + \varepsilon c_d)\,\mathbf{j} + (d_r + \varepsilon d_d)\,\mathbf{k}$. Where $q_r$ and $q_d$ are quaternions, which are the real and dual parts of the dual quaternion $Q$, respectively; $a, b, c, d$ are complex numbers; $a_r,\ b_r,\ c_r,\ d_r,\ a_d,\ b_d,\ c_d,\ d_d$ are real numbers; $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are imaginary units, which is the square root of -1. $\mathbf{i}, \mathbf{j}, \mathbf{k}$ satisfied the Hamilton rule: $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. $\mathbf{i}\varepsilon = \varepsilon\mathbf{i}, \mathbf{j}\varepsilon = \varepsilon\mathbf{j}, \mathbf{k}\varepsilon = \varepsilon\mathbf{k}$, Quaternion is a special dual quaternion.

**Conjugate**: The conjugate dual quaternion of the dual quaternion $Q$ is $\bar{Q} = [\bar{q_r},\ \ \bar{q_d}]$.

**Norm**: The norm of dual quaternion $Q$ is $\|Q\| = [\|q_r\|\,, \frac{q_d\bar{q_r} + q_r\bar{q_d}}{2\|q_r\|}]$.

**Dual Quaternion Multiplication**: The dual quaternion multiplication does not follow the commutative law, which means that $Q_1 \underline{\otimes} Q_2 \neq Q_2 \underline{\otimes} Q_1$. In addition to the commutative law, the associative and distributive laws hold within dual quaternions. The Multiplication product of two dual quaternions $Q_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$ and $Q_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$ is as follows:

$$\begin{aligned} Q_1 \underline{\otimes} Q_2 = &\,(a_1 \circ a_2 - b_1 \circ b_2 - c_1 \circ c_2 - d_1 \circ d_2) \\ &+ (a_1 \circ b_2 + b_1 \circ a_2 + c_1 \circ d_2 - d_1 \circ c_2)\,\mathbf{i} \\ &+ (a_1 \circ c_2 - b_1 \circ d_2 + c_1 \circ a_2 + b_1 \circ d_2)\,\mathbf{j} \\ &+ (a_1 \circ d_2 + b_1 \circ c_2 - c_1 \circ b_2 + d_1 \circ a_2)\,\mathbf{k} \end{aligned}, \qquad (2)$$

where $\circ$ denotes the element-wise multiplication between two vectors. Dual quaternion multiplication can simultaneously represent translation and rotation operations.

# IV. Proposed Method

In this section, we propose an effective temporal KGC method, which combines translation and rotation operations in dual quaternion space for temporal KGC. We can also use this method to represent temporal information as rotation in quaternion vector space. We call this method as quaternion rotation(QuatR) method. We mainly introduce the method that combines translation and rotation in dual quaternion vector space.

## *A. Combining Translation and Rotation in Dual Quaternion Vector Space for Temporal KGC*

We combine translation and rotation operations in dual quaternion vector space represented by the dual quaternion multiplication to model the effect of time on relations.

**Dual Quaternion Embedding of Entities, Relations and Timestamps**. Given a temporal knowledge graph $\mathcal{G}$ with $N$ entities, $M$ relations and $T$ timestamps, we use the dual quaternion matrix $\mathbf{Q} \in \mathbb{H}^{N \times k}$ to represent all entity embeddings, use the dual quaternion matrix $\mathbf{R} \in \mathbb{H}^{M \times k}$ to represents all relation embeddings, and use the dual quaternion matrix $\mathbf{W} \in \mathbb{H}^{T \times k}$ to represent all timestamp embeddings, where k is the embedding dimensions. Given a quadruple $(s, r, o, \tau)$, the embedding of the head entity $s$ is expressed as $Q_s = a_{rs} + b_{rs}\mathbf{i} + c_{rs}\mathbf{j} + d_{rs}\mathbf{k} + \varepsilon\,(a_{ds} + b_{ds}\mathbf{i} + c_{ds}\mathbf{j} + d_{ds}\mathbf{k})$; the embedding of the tail entity $o$ is expressed as $Q_o = a_{ro} + b_{ro}\mathbf{i} + c_{ro}\mathbf{j} + d_{ro}\mathbf{k} + \varepsilon\,(a_{do} + b_{do}\mathbf{i} + c_{do}\mathbf{j} + d_{do}\mathbf{k})$; the embedding of relation $r$ is expressed as $R_r = a_{rr} + b_{rr}\mathbf{i} + c_{rr}\mathbf{j} + d_{rr}\mathbf{k} + \varepsilon\,(a_{dr} + b_{dr}\mathbf{i} + c_{dr}\mathbf{j} + d_{dr}\mathbf{k})$; the embedded representation of the timestamp $\tau$ is expressed as $W_\tau = a_{r\tau} + b_{r\tau}\mathbf{i} + c_{r\tau}\mathbf{j} + d_{r\tau}\mathbf{k} + \varepsilon\,(a_{d\tau} + b_{d\tau}\mathbf{i} + c_{d\tau}\mathbf{j} + d_{d\tau}\mathbf{k})$.

**Relation Rotation Based on Dual quaternion Multiplication**. ComTR represents the timestamp $\tau$ as the combination translation and rotation, and obtains the feature of the relation $r$ at timestamp $\tau$. First, we normalize the timestamp dual quaternion $W_\tau$ to the unit dual quaternion $W_\tau^\Delta$. We define $W_\tau = (r, d)$, $r = (a_{rr}, b_{rr}, c_{rr}, d_{rr})$, $d = (a_{dr}, b_{dr}, c_{dr}, d_{dr})$. We first get a standardized variable called $r'$:

$$r' = \frac{r}{\|r\|} = \frac{a_{rr} + b_{rr}\mathbf{i} + c_{rr}\mathbf{j} + d_{rr}\mathbf{k}}{\sqrt{{a_{rr}}^2 + {b_{rr}}^2 + {c_{rr}}^2 + {d_{rr}}^2}}. \tag{3}$$

Then we get

$$d' = d - \frac{(d, r)}{r, r} r, \tag{4}$$

$W_\tau^\Delta = (r', d')$ is the unit dual quaternion. We can deduce that $\|r'\| = 1$, $r'\bar{d'} + d'\bar{r'} = 0$.

Next, we translate and rotate the relation by doing a dual quaternion multiplication between the relation $R_r$ and the timestamp $W_\tau^\Delta$ to get a representation of the relation at at timestamp $\tau$:

$$R_{r\tau} = a_{r\tau} + b_{r\tau}\mathbf{i} + c_{r\tau}\mathbf{j} + d_{r\tau}\mathbf{k} = R_r \underline{\otimes} W_\tau^\Delta, \tag{5}$$

where $\underline{\otimes}$ is the dual quaternion multiplication, which can represent the translation and rotation of $R_r$ using $W_\tau^\Delta$ in space.

**Head entity Rotation Based on Dual quaternion Multiplication**. ComTR represents the relation $R_{r\tau}$ under timestamp $\tau$ as the combination translation and rotation. First, we normalize the timestamp dual quaternion $R_{r\tau}$ to the unit dual quaternion $R_{r\tau}^\Delta$. We then translate and rotate the head entity by doing a dual quaternion multiplicationt between the head entity $Q_s$ and the temporal relation $R_{r\tau}$ to get the head entity representation:

$$Q_{sr\tau} = a_{sr\tau} + b_{sr\tau}\mathbf{i} + c_{sr\tau}\mathbf{j} + d_{sr\tau}\mathbf{k} = Q_s \underline{\otimes} R_{r\tau}^\Delta, \tag{6}$$

where $\underline{\otimes}$ is the dual quaternion multiplication.

**Scoring Function**: We apply the quaternion inner product as the scoring function:

$$\begin{aligned}\phi(s, r, o, \tau) &= Q_{sr\tau} \cdot Q_o \\ &= \langle a_{sr\tau}, a_o \rangle + \langle b_{sr\tau}, b_o \rangle. \\ &+ \langle c_{sr\tau}, c_o \rangle + \langle d_{sr\tau}, d_o \rangle\end{aligned} \tag{7}$$

For the time period quadruple $(s, r, o, \tau) \in S \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$, where $\tau = [t_b, t_e]$, $t_b$ represents the start time of the fact, and $t_e$ represents the end time of the fact. We divide the quadruple into two quadruples, and its score function takes the average of the two scores to reduce the computational cost:

$$\phi(s, r, o, [t_b, t_e]) = \frac{1}{2}\phi(s, r, o, t_b) + \frac{1}{2}\phi(s, r, o, t_e). \tag{8}$$

### *B. Optimization*

**Regularization**: KGC methods generally use various regularization methods to reduce the degree of overfitting and improve the generalization ability of methods. Temporal KGs can be regarded as order 4 tensors, so we design a nuclear 4-norm regularization to alleviate overfitting as follows:

$$\mathcal{L}_4 = \|Q\|^4 + \|R\|^4 + \|W\|^4, \tag{9}$$

where $\|\cdot\|^4$ represents the four-norm of the tensor kernel.

**Temporal Regularization**: For temporal knowledge graph time constraints, time regularization is commonly used to smooth the representation of adjacent timestamps. In this work, we use linear time regularization as follows:

$$\mathcal{L}_\tau = \sum_{i=1}^{T-1} \left\| W_{\tau_{i+1}} - W_{\tau_i} - W_b \right\|, \tag{10}$$

where $W_b$ is a linear time-constrained bias term, initialized randomly and can be learned during training. The linear time constraint can ensure that the timestamps of adjacent times are close to each other, and the timestamps of distant times are significantly different, ensuring the smoothness of time. And the deviation term can represent the sudden change of adjacent times.

### *C. Loss Function*

We use full multiclass log-softmax loss function as quadruple loss. The total loss is the sum of the quadruple loss and regularization terms. The final training objective is to minimize:

$$\begin{aligned}\mathcal{L} = \sum_{(s,r,o,\tau)\in S} &\left( -log \frac{exp(\phi(s, r, o, \tau))}{\sum_{s' \in \mathcal{E}} exp(\phi(s', r, o, \tau))} \right. \\ &\left. -log \frac{exp(\phi(s, r, o, \tau))}{\sum_{o' \in \mathcal{E}} exp(\phi(s, r, o', \tau))} \right) \\ &+ \lambda_\alpha \mathcal{L}_4 + \lambda_\tau \mathcal{L}_\tau\end{aligned}, \tag{11}$$

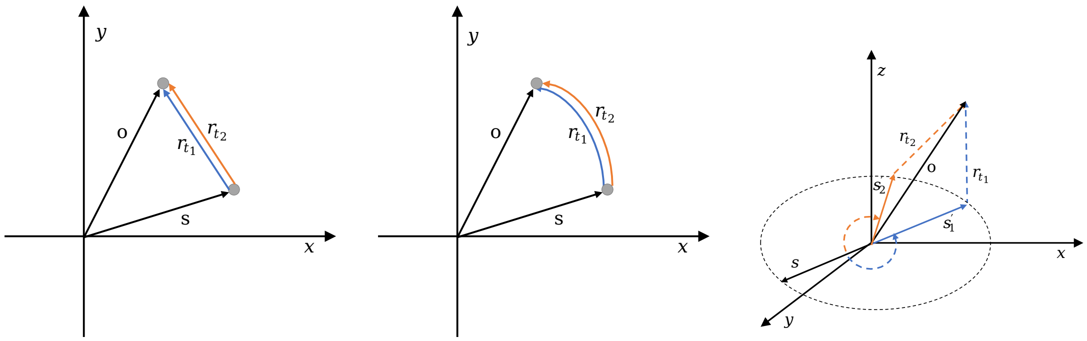
where $\lambda_\alpha$ is the regularization coefficient and $\lambda_\tau$ is the time constraint regularization coefficient.

### *D. Analysis of relation patterns in temporal KGs*

In this section, we analyze the main features of temporal KGs compared to static KGs and the advantages of ComTR compared to other existing methods.

In static KGs, the performance of methods is related to its relation patterns (such as symmetry/antisymmetry, inversion, synthesis, and multiple relations) that can be expressed. Different geometry structures (e.g. translation) have different expressive abilities for relation patterns. The commonly used embedding structures in the field of KGC include translation and rotation. However, a single translation or rotation structure cannot express all the relation patterns. Translation structure can model inversion and composition relations, but cannot model symmetry/antisymmetry and multiple relations (As shown in Figure 1(a)). Rotation structure can model symmetry/antisymmetry, but still cannot model multiple relations (As shown in Figure 1(b)). Simple translation and rotation cannot model all relation patterns. We can see that the combination of rotation and translation can overcome their respective shortcomings to model multiple relations (As shown in Figure 1(c)).

Next, we consider the importance of multiple relations in temporal KGs. As we stated in introduction, due to the addition of temporal information, the complexity of temporal KGs is




(a) translation.

(b) rotation.

(c) The combination of translation and rotation.

Fig. 1. Illustrations of different transformations modeling multiple relations. $r_t$ represents the relation $r$ at timestamp $t$. We can see that combining translation and rotation can model multiple relations.

greatly increased. The compounding of time and relation forms many multiple relations. For these multiple relations, neither translation nor rotation methods can model them. Therefore, for temporal KGs, the simple translation or the rotation method is not perfectly applicable.

As seen in our method and score function (Equation 11), we can get following results in ComTR:

**Lemma 1.** *ComTR can model the (anti)symmetry pattern for temporal KGs.*

**Lemma 2.** *ComTR can model the inversion pattern for temporal KGs*

**Lemma 3.** *ComTR can model the composition pattern for temporal KGs.*

**Lemma 4.** *ComTR can model the multiple relations pattern for temporal KGs.*

## V. Experiments

In this section, we evaluate ComTR on four public benchmarks datasets ICEWS14, ICEWS05-15, YAGO11k, and Wikidata12k. We first introduce the dataset and baseline, and then report the experimental results.

TABLE I
Statistics of datasets.

| Dataset | ICEWS14 | ICEWS05-15 | YAGO11k | Wikidata12k |
|---|---|---|---|---|
| Entities | 6869 | 10194 | 10623 | 12544 |
| Relations | 230 | 251 | 10 | 24 |
| Facts | 90730 | 461329 | 20507 | 40621 |
| Time Span | 2014 | 2005-2015 | 1513-2017 | 1526-2020 |

### A. Benchmark Datasets

We briefly present ICEWS14, ICEWS5-15, YAGO11k and Wikidata12k, and report their main statistics in Table I. ICEWS14 and ICEWS05-15 are the two most commonly used datasets extracted from the large global events database - the Integrated Crisis Early Warning System (ICEWS). ICEWS is a database of important global political events, which are political events with time information (eg, *(Barack Obama, Make a visit, China, 2009-11-21)*). It is worth noting that the time annotations in ICEWS are all time points. ICEWS14 contains events that occurred in 2014, and ICEWS05-15 contains events that occurred between 2005-2015. Both datasets are filtered only by selecting the most frequently occurring entities in the knowledge graph.

YAGO11k and Wikidata12k are two common-sense temporal knowledge graph datasets from YAGO and Wikidata datasets, respectively. The time annotations of these two datasets have two ways: time points (such as *(Allan_ Guy, wasBornIn, Launceston, _ Tasmania, 1890 – 11 – 30)* ) and time period (such as *(Newsweb _ Corporation, possession, WCFJ _ ( defunct ), 1998 – ## – # – 2015 – #–# )*). For a time period quadruple which one of the timestamp is *'####-##-##'*, the default is the start or end of the dataset time boundary.

### B. Evaluation Protocol and Baselines

We follow the popular evaluation indicators of previous temporal KGC, and use mean reciprocal rank (MRR), Hits@1, Hits@3, and Hits@10, as evaluation indicators to evaluate our model. MRR is the mean inverse rank of the correct entity. Hit@n measures the proportion of correct entities among the top n entities, where n=1, 3, 10.

We compare with the classic and the latest, most popular and most effective temporal KGC baselines and the static KGC baselines. For static baselines, we use TransE, DistMult, SimpIE [37], RotatE, and QuatE. For temporal KGC methods, we consider TTransE, HyTE, TA-DistMult [**?**], DE-SimplE [**?**], ATiSE, TeRo, TComplEx, DyERNIE, ChronoR, TeLM, RotateQVS and HERCULES.

### C. Experimental Setup

We implemented our method using the PyTorch framework, used the Adagrad optimizer to train our method, and tested it on a single GPU. The specific hyperparameters are determined by the grid search on the validation set, where the learning

TABLE II
KGC RESULTS ON ICEWS14 AND ICEWS05-15.THE BEST SCORE IS IN **BOLD** AND SECOND BEST SCORE IS UNDERLINED. QUATR IS THE QUATERNION ROTATION METHOD.

| Dataset | ICEWS14 | | | | ICEWS05-15 | | | |
|---|---|---|---|---|---|---|---|---|
| Metrics | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| TransE | 28.0 | 9.4 | - | 63.7 | 29.4 | 9.0 | - | 66.3 |
| DistMult | 43.9 | 32.3 | - | 67.2 | 45.6 | 33.7 | - | 69.1 |
| SimpIE | 45.8 | 34.1 | 51.6 | 68.7 | 47.8 | 35.9 | 53.9 | 70.8 |
| RotatE | 41.8 | 29.1 | 47.8 | 69.0 | 30.4 | 16.4 | 35.5 | 59.5 |
| QuatE | 47.1 | 35.3 | 53.0 | 71.2 | 48.2 | 37.0 | 52.9 | 72.7 |
| TTransE | 25.5 | 7.4 | - | 60.1 | 16.4 | 10.5 | 18.9 | 27.2 |
| HyTE | 29.7 | 10.8 | 41.6 | 65.5 | 31.6 | 11.6 | 44.5 | 66.8 |
| TA-DistMult | 47.7 | 36.3 | - | 68.6 | 47.4 | 34.6 | - | 72.8 |
| DE-SimpIE | 52.6 | 41.8 | 59.2 | 72.5 | 51.3 | 39.2 | 57.8 | 74.8 |
| ATiSE | 55.0 | 43.6 | 62.9 | 75.0 | 51.9 | 37.8 | 60.6 | 79.4 |
| TeRo | 56.2 | 46.8 | 62.1 | 73.2 | 58.6 | 46.9 | 66.8 | 79.5 |
| TComplEx | 61.0 | 51.9 | 65.9 | 77.2 | 66.6 | 58.3 | 71.8 | 81.6 |
| DyERNIE | 66.9 | 59.9 | <u>71.4</u> | <u>79.7</u> | 73.9 | 67.9 | <u>77.3</u> | **85.5** |
| ChronoR | 62.5 | 54.7 | 66.9 | 77.3 | 67.5 | 59.6 | 72.3 | 82.0 |
| TeLM | 62.5 | 54.5 | 67.3 | 77.4 | 67.8 | 59.9 | 72.8 | 82.3 |
| HERCULES | <u>69.4</u> | <u>65.0</u> | <u>71.4</u> | 77.9 | <u>73.5</u> | <u>68.6</u> | 76.1 | 82.9 |
| RotateQVS | 59.1 | 50.7 | 64.2 | 75.4 | 63.3 | 52.9 | 70.9 | 81.3 |
| QuatR | 62.7 | 54.9 | 67.1 | 77.3 | 68.1 | 60.4 | 72.7 | 82.3 |
| ComTR | **71.5** | **66.4** | **73.8** | **80.2** | **75.2** | **70.4** | **78.0** | <u>84.0</u> |

TABLE III
KGC RESULTS ON YAGO11K AND WIKIDATA12K.THE BEST SCORE IS IN **BOLD** AND SECOND BEST SCORE IS UNDERLINED. QUATR IS THE QUATERNION ROTATION METHOD.

| Dataset | YAGO11k | | | | Wikidata12k | | | |
|---|---|---|---|---|---|---|---|---|
| Metrics | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| TTransE | 10.8 | 2.0 | 15.0 | 25.1 | 17.2 | 9.6 | 18.4 | 32.9 |
| HyTE | 13.6 | 3.3 | - | 29.8 | 25.3 | 14.7 | - | 48.3 |
| TA-DistMult | 15.5 | 9.8 | - | 26.7 | 23.0 | 13.0 | - | 46.1 |
| TeRo | 18.7 | 12.1 | 19.7 | 31.9 | 29.9 | 19.8 | 32.9 | 50.7 |
| ATiSE | 18.5 | 12.6 | 18.9 | 30.1 | 25.2 | 14.8 | 28.8 | 46.2 |
| TComplEx | 18.5 | 12.7 | 18.3 | 30.7 | 33.1 | <u>23.3</u> | 35.7 | 53.9 |
| TeLM | <u>19.1</u> | <u>12.9</u> | 19.4 | 32.1 | 33.2 | 23.1 | 36.0 | <u>54.2</u> |
| RotateQVS | 18.9 | 12.4 | <u>19.9</u> | <u>32.3</u> | - | - | - | - |
| QuatR | 18.8 | 12.8 | 18.5 | 31.8 | <u>33.3</u> | <u>23.3</u> | <u>36.4</u> | 54.1 |
| ComTR | **33.9** | **25.3** | **37.7** | **51.4** | **56.4** | **48.1** | **59.7** | **74.2** |

rate is 0.1 and the batch size is fixed as 1000. Regularization weights $\lambda_\alpha$ and $\lambda_\tau$ are adjusted in the range of 0, 0.001, 0.0025, 0.005, 0.0075 ,0.01, ... , 0.1. Embedded dimension $d$ is 100.

### D. Results

The results of testing according to the standard temporal KGC experiment settings are shown in Table II and Table III. To better evaluate the performance of our model, we augment the quaternion rotation method, named QuatR.

As shown in Table II, our ComTR basically achieves state-of-the-art results on both ICEWS14 and ICEWS05-15, and QuatR also surpasses the results of the 2021 rotation method ChronoR. Our ComTR far exceeds the effect of rotation and translation methods including QuatR. this demonstrates the shortcomings of rotation methods in temporal KGC and the advantages of ComTR. The performance of DyERNIE and HERCULES are much better than other existing methods, because these two methods are models on manifold space, and can also represent multi-relational manifold space. However, compared with ComTR, the curvature model on the manifold space is relatively complex and difficult to optimize.

As shown in Table III, we observe that our method outperforms all the baseline methods over the two datasets across all metrics consistently. The performance of QuatR is close to the state-of-the-art model. The performance of ComTR greatly exceeds the current state-of-the-art model, and the improvement exceeds the improvement on ICEWS datasets, which demonstrates that ComTR is not only suitable for temporal data in a single domain. And in more common commonsense datasets, due to the rising complexity of the data, a more applicable ComTR is needed.

### E. Case Study

To provide a fine-grained analysis of the results of ComTR, we report its performance of multiple relations on ICEWS14 in Table IV. We selected several sets of facts of multiple relationships and tested the performance of the two methods on these facts. Compared with static KGs, Temporal KGs have a large portion of multiple relations facts. We see that ComTR have a better performance on multiple relations compared with

TABLE IV
COMPARISON OF H@1 OF MULTIPLE RELATIONS FOR COMTR AND QUATR ON ICEWS14. THESE CASES HAVE THE SAME HEAD ENTITIES AND TAIL ENTITIES.

| Quadruple | QuatR | ComTR | Lift |
|---|---|---|---|
| *(North Korea, –, South Korea, –)* | 45.6 | 61.2 | 34.2% |
| *(Iran, –, Iraq, –)* | 47.3 | 62.9 | 32.9% |
| *(China, –, Japan, –)* | 48.5 | 61.8 | 27.4% |
| *(Japan, –, North Korea, –)* | 47.4 | 63.3 | 33.5% |

the rotation method QuatR. And the promotion on multiple relations facts outweighs the overall promotion. These fine-grained results support our hypothesis that ComTR can model all the relation patterns and can pick the best geometric transformation (or combinations thereof) for optimal performance.

## VI. CONCLUSION

We propose a novel temporal KGC methods, which combining translation and rotation in dual quaternion space to represent temporal relations. The multiplication in dual quaternion space can represent translation and rotation at the same time, and can model multiple relations common to temporal KGs, making up for the shortcomings of translation and rotation methods. We then evaluate our method and show that our method achieves the best performance for temporal KGC on four common datasets, and can handle complex temporal relations in temporal KGs. In future work, we will study other geometric models to express various relations in temporal KGs, and explore more complex and comprehensive modeling of temporal KGs.

## REFERENCES


[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*. Springer, 2007, pp. 722–735.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[3] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[4] A. Saxena, S. Chakrabarti, and P. Talukdar, "Question answering over temporal knowledge graphs," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 6663–6676.

[5] K. Tu, P. Cui, D. Wang, Z. Zhang, J. Zhou, Y. Qi, and W. Zhu, "Conditional graph attention networks for distilling and refining knowledge graphs in recommendation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1834–1843.

[6] J. Jung, J. Jung, and U. Kang, "Learning to walk across time for interpretable temporal knowledge graph completion," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 786–795.

[7] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić, "Introducing wikidata to the linked data web," in *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13*. Springer, 2014, pp. 50–65.

[8] J. Biega, E. Kuzey, and F. M. Suchanek, "Inside yago2s: A transparent information extraction architecture," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 325–328.

[9] J. Lautenschlager, S. Shellman, and M. Ward, "Icews event aggregations," *Harvard Dataverse*, vol. 3, no. 595, p. 28, 2015.

[10] Z. Han, P. Chen, Y. Ma, and V. Tresp, "Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7301–7316.

[11] S. Montella, L. M. R. Barahona, and J. Heinecke, "Hyperbolic temporal knowledge graph embeddings with relational and time curvatures," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 3296–3308.

[12] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[13] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[14] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[15] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.

[16] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.

[17] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[18] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*.

[19] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," *Advances in neural information processing systems*, vol. 32, 2019.

[20] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 2018, pp. 593–607.

[21] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[22] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *International Conference on Learning Representations*.

[23] M. Yu, J. Guo, J. Yu, T. Xu, M. Zhao, H. Liu, X. Li, and R. Yu, "Bdri: block decomposition based on relational interaction for knowledge graph completion," *Data Mining and Knowledge Discovery*, pp. 1–21, 2023.

[24] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *Companion proceedings of the the web conference 2018*, 2018, pp. 1771–1776.

[25] S. S. Dasgupta, S. N. Ray, and P. P. Talukdar, "Hyte: Hyperplane-based temporally aware knowledge graph embedding." in *EMNLP*, 2018, pp. 2001–2011.

[26] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," in *International Conference on Learning Representations*.

[27] C. Xu, M. Nayyeri, F. Alkhoury, H. S. Yazdi, and J. Lehmann, "Temporal knowledge graph embedding model based on additive time series decomposition," *arXiv preprint arXiv:1911.07893*, 2019.

[28] C. Xu, M. Nayyeri, F. Alkhoury, H. S. Yazdi, and J. Lehmann, "Tero: A time-aware knowledge graph embedding via temporal rotation," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 1583–1593.

[29] A. Sadeghian, M. Armandpour, A. Colas, and D. Z. Wang, "Chronor: Rotation based temporal knowledge graph embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6471–6479.

[30] C. Xu, Y.-Y. Chen, M. Nayyeri, and J. Lehmann, "Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 2569–2578.

[31] K. Chen, Y. Wang, Y. Li, and A. Li, "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)*, 2022, pp. 5843–5857.

[32] J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton, "Temp: Temporal message passing for temporal knowledge graph completion," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5730–5746.

[33] J. Jung, J. Jung, and U. Kang, "T-gap: Learning to walk across time for temporal knowledge graph completion," *arXiv preprint arXiv:2012.10595*, 2020.

[34] Z. Han, P. Chen, Y. Ma, and V. Tresp, "Explainable subgraph reasoning for forecasting on temporal knowledge graphs," in *International Conference on Learning Representations*, 2021.

[35] W. R. Hamilton, "Xi. on quaternions; or on a new system of imaginaries in algebra," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 33, no. 219, pp. 58–60, 1848.

[36] C. Clifford, "Preliminary sketch of biquaternions," *Proceedings of the London Mathematical Society*, vol. 1, no. 1, pp. 381–395, 1871.

[37] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," *Advances in neural information processing systems*, vol. 31, 2018.