TCompoundQ: Translation, Rotation, and Scaling in Quaternion Vector Space for Temporal Knowledge Graph Completion

Rushan Geng[1] and Cuicui Luo[2](✉)

[1] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
gengrushan23@mails.ucas.ac.cn
[2] International College, University of Chinese Academy of Sciences, Beijing, China
luocuicui@ucas.ac.cn

Abstract. In temporal knowledge graph completion (TKGC), translation, rotation, and scaling have proven effective in capturing temporal dynamics. To comprehensively model the diverse relationships in temporal knowledge graph embeddings (TKGE), it is crucial to incorporate multiple transformations. By employing translation, scaling, and rotation along the temporal dimension, evolving semantic information can be captured with greater precision. Biquaternions, which can seamlessly combine multiple geometric transformations—including scaling, translation, Euclidean rotation, and hyperbolic rotation—are particularly suitable for this purpose. In this work, we present TCompoundQ, a model built within the quaternion vector space that unifies translation, rotation, and scaling operations through biquaternions. Experimental results on five publicly available temporal datasets validate the effectiveness of TCompoundQ, showcasing its superior performance.

Keywords: Temporal knowledge graph completion · Temporal knowledge graph · Quaternion · Compound.

1 Introduction

The rapid progress in knowledge graph (KG) technology has made it an important tool in many semantic computing tasks. However, real-world knowledge graphs are often incomplete, which limits their use in advanced tasks like recommendation systems. Knowledge Graph Embedding (KGE) tries to solve this problem by representing entities and relationships as low-dimensional vectors. This method helps predict missing links and fill in the gaps in the graph. Although knowledge graphs store static facts about the world, real-world data is very dynamic, with facts changing over time. This means that traditional knowledge graphs need to be extended to include a time dimension, leading to the development of Temporal Knowledge Graphs (TKGs).

In conventional knowledge graphs, facts are represented as triples (s, r, o), where s denotes the head entity, r represents the relation, and o signifies the tail entity. However, in real-world scenarios, facts evolve over time. To account for this, temporal knowledge graphs introduce a timestamp τ, extending the representation to quadruples (s, r, o, τ). For example, the fact "Barack Obama, Make a visit, Malaysia, 2014-04-25" indicates that Barack Obama visited Malaysia on April 25, 2014, with "Make a visit" as a dynamic relation.

Temporal knowledge graphs require temporal reasoning to model time-dependent relationships effectively. Several works have introduced mechanisms to incorporate temporal information into KG models. TeRo [1] employs rotational transformations in embedding space, combining temporal and relational embeddings to capture time-sensitive relationships. ChronoR [2] extends this idea by applying multi-dimensional rotational transformations using quaternion embeddings to achieve finer temporal granularity. RotateQVS [3] also utilizes quaternion embeddings for both entities and relations, interpreting time as a rotation operation on entity embeddings to model evolving relationships. The DualE model [4] integrates rotation and translation within a biquaternion space, supporting a wide range of relational patterns. Research by [5] demonstrated that complex geometric transformations are effective in capturing diverse relational structures, while Guo et al. [6] proved that biquaternion transformations correspond to a combination of circular and hyperbolic rotations.

These studies highlight the importance of geometric operations, such as translation and rotation, in developing temporal knowledge graph embedding models. These operations not only enhance the expressive power of embeddings but also provide robust mechanisms for capturing complex dynamic relationships in temporal data. However, effectively integrating multiple geometric operations, including translation, scaling, and rotation, is still an open research challenge. The interplay among these transformations significantly impacts model performance, making it essential to explore methods that incorporate them in a unified manner.

To address these issues, we investigate the use of a biquaternion space to integrate translation, scaling, and rotation operations. In this study, we propose a novel model that leverages these transformations within a biquaternion space to capture both time-varying and static properties in low-dimensional relational representations. Our model applies relation-specific operations, combining them with temporal information to refine entity representations. We evaluate the confidence of quadruples by measuring the semantic similarity between head and tail entity embeddings.

In summary, the main contributions of this paper are as follows:

– We propose a new TKGC model, TCompoundQ, which integrates translation, scaling, and rotation operations within a biquaternion space.
– We provide theoretical analyses and proofs demonstrating the model's capability in capturing various relation patterns, including symmetric, asymmetric, and inverse relations.
– We conduct experiments on five benchmark datasets, showing that TCompoundQ achieves state-of-the-art performance, surpassing many recent baseline models.

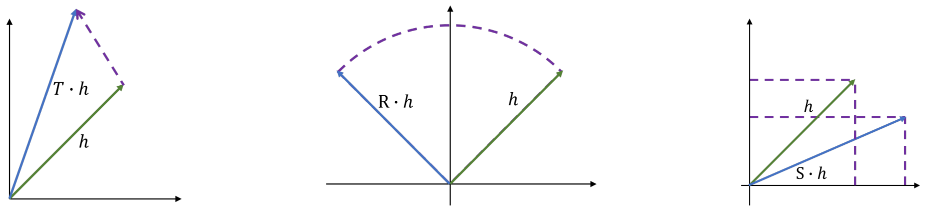
2 Related work

Generally, temporal knowledge graph embedding can be categorized into two primary types: Translational Distance Models and Semantic Match Models. Embedding-based methods leverage computed embeddings for entities and relations to assess the validity of quadruples.

Temporal Translational Distance Models: The TTransE model [7] introduces temporal order constraints within the geometric framework of the TransE embedding space, effectively capturing temporal relationships among adjacent facts. HyTE [8] extends this approach by embedding temporal dimensions into a distinct hyperplane, thereby explicitly linking time with entities and relations to enhance temporal contextualization. Similarly, TeRo [1] conceptualizes the temporal evolution of entity representations through rotations in a complex vector space, providing a dynamic representation of temporal changes. RotateQVS [3] leverages quaternion vector spaces to model temporal entities, demonstrating improved predictive performance. However, most of these models primarily employ a single geometric representation to capture temporal patterns. This limitation often results in reduced flexibility, as different temporal patterns may require distinct geometric structures. Consequently, selecting the most suitable manifold for a given Temporal Knowledge Graph often requires manual intervention based on the inherent structural characteristics of the dataset.

Temporal Semantic Match Models: The TA-DistMult model [9] introduces temporal information into its scoring functions, significantly enhancing Temporal Knowledge Graph (TKG) modeling by integrating time as a fundamental factor. Similarly, Holographic Embeddings (HOLE) [10] utilize circular correlation to generate compositional representations, adding interpretative depth to the embeddings. ComplEx [11] further addresses the antisymmetric nature of certain relations by employing complex vector spaces to better capture relational dynamics. Building upon these foundations, TComplEx [12] extends complex embeddings by incorporating temporal components, thereby improving TKG representation through time-specific modeling. The ATiSE framework [13] refines temporal evolution decomposition by dividing entity and relation embeddings into trend, seasonal, and random components, offering a more granular perspective on temporal dynamics. DE-SimplE [14] integrates diachronic entity embedding functions into the static SimplE framework, effectively representing point-in-time event facts. TimePlex [15] leverages the cyclic nature of temporal data, embedding temporal interactions within relational structures to enrich the model's temporal representation. ChronoR [2] introduces chronological rotation embeddings, capturing the interplay between temporal and relational dimensions within TKGs.

More recently, LCGE [16] has introduced an innovative approach by applying temporal logic rules to regularize entity embeddings while incorporating commonsense reasoning, thereby improving learning efficiency. This comprehensive approach enhances both predictive accuracy and logical coherence. Lastly, TeAST [17] employs timestamps as a mapping operation along the Archimedean spiral axis, associating relations with this axis to model the dynamic evolution of relationships over time. HGE [18] embeds temporal facts into a product space consisting of three distinct geometric subspaces: complex, split-complex, and dual spaces. This design enables the model to leverage the unique geometric properties of each subspace to effectively capture a wide range of temporal patterns.



(a) Translation (b) Rotation (c) Scaling

Fig. 1: Illustrations of fundamental geometric transformations used in embedding temporal knowledge graphs.

3 Background

3.1 Problem Formulation

A temporal knowledge graph $\mathcal{G}$ consists of quadruples represented as (s, r, o, τ), where $s, o \in \mathcal{E}$ denote subject (head) and object (tail) entities, respectively, and $r \in \mathcal{R}$ represents a relation connecting the subject to the object. The timestamp $\tau \in \mathcal{T}$ specifies the temporal context of the interaction between s and o under relation r. Here, $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ represent the finite sets of entities, relation types, and timestamps, correspondingly. Temporal knowledge graph embedding techniques aim to map each entity, relation, and timestamp into a continuous vector space. These methods define a time-aware scoring function $f(s, r, o, \tau)$ to evaluate the likelihood of observing the quadruple (s, r, o, τ) in the graph. In response to queries like $(s, r, ?, \tau)$ or $(?, r, o, \tau)$, the goal of Temporal Knowledge Graph Completion is to infer the missing subject or object entity by leveraging known temporal facts. Ideally, the scoring function assigns higher scores to genuine facts over incorrect ones, reflecting their relative plausibility within the TKG framework.

3.2 Translation Rotation and Scaling

Translation, rotation, and scaling are fundamental operations in computer graphics, widely applied across various engineering applications. Typically, they are represented in matrix form to facilitate efficient computation and manipulation of graphic objects, like in Figure 1. Ge et al.[5] demonstrated that the combination of translation, rotation, and scaling can effectively handle complex relations in knowledge graphs. Below, we briefly introduce translation, rotation, and scaling in the 2D plane.

The translation operation in 2D space can be represented as:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where t_x and t_y represent the translation amounts along the x-axis and y-axis, respectively.

The 2D rotation matrix can be written as:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where θ is the rotation angle.

The 2D scaling matrix can be expressed as:

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where s_x and s_y are the scaling factors along the x-axis and y-axis, respectively.

3.3 Quaternions

Quaternions are a type of hypercomplex number that extend the concept of complex numbers. Unlike complex numbers, which have two components (a real part and an imaginary part), quaternions consist of four components: one real part and three imaginary parts. Guo et al. [6] mathematically proved that a biquaternions transformation is equivalent to a combination of a circular rotation and a hyperbolic rotation. A quaternion is typically represented as

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \tag{4}$$

where a is the real part, and b, c, and d are the imaginary parts. The imaginary units $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ satisfy the following multiplication relations:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1. \tag{5}$$

The product of two quaternions, q_1 and q_2, denoted by q_1q_2, can be obtained using the standard algebraic distributive property (following the normal multiplication rule for complex numbers when multiplying coefficients). The specific calculation is as follows:

$$\begin{aligned} q_1q_2 =& w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2 \\ &+ (w_1x_2 + x_1w_2 + y_1z_2 - z_1y_2)\mathbf{i} \\ &+ (w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2)\mathbf{j} \\ &+ (w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2)\mathbf{k} \end{aligned} \tag{6}$$

This process is known as the Hamilton product between q_1 and q_2. The multidimensional structure of quaternion algebra offers unique advantages in various applications, such as rotation and geometric transformations. Specifically, in fields like computer graphics, physics simulations, and knowledge graph embeddings, quaternions are highly regarded for their ability to achieve smooth rotations and high degrees of freedom.

4 Method

4.1 TCompoundQ

In this section, we present our model, TCompoundQ, which applies compound geometric operations on both relations and timestamps, mapping them into a dual quaternion space. For a quadruple (s, r, o, τ) in a temporal knowledge graph, we use $\mathbf{Q}_s \in \mathbb{R}^{2\times k}$, $\mathbf{Q}_r \in \mathbb{R}^{5\times k}$, $\mathbf{Q}_o \in \mathbb{R}^{2\times k}$, and $\mathbf{Q}_\tau \in \mathbb{R}^{4\times k}$ to denote the embeddings of the head entity s, relation r, tail entity o, and timestamp τ, respectively. Here, $\mathbf{S}_\tau$, $\mathbf{R}_\tau \in \mathbf{Q}_\tau$, and $\mathbf{T}_r$, $\mathbf{S}_r$, $\mathbf{R}_r \in \mathbf{Q}_r$. We employ $\mathbf{S}_\tau$ and $\mathbf{T}_\tau$ to perform scaling and translation operations on the relation, calculated as:

$$\mathbf{T}_{r\tau} = \mathbf{S}_\tau \circledast (\mathbf{T}_r \cdot \mathbf{T}_\tau) \tag{7}$$

where $\mathbf{S}_\tau$, $\mathbf{T}_r$, $\mathbf{T}_\tau \in \mathbb{R}^{2\times k}$, and $\circledast$ denotes the Hamilton product between two elements, specified as:

$$\begin{aligned} \mathbf{Q} &= \mathbf{Q}' \circledast \mathbf{Q}^\times \\ &= \left(w' \otimes w^\times - x' \otimes x^\times - y' \otimes y^\times - z' \otimes z^\times\right) + \\ &\quad \left(w' \otimes x^\times + x' \otimes w^\times + y' \otimes z^\times - z' \otimes y^\times\right) \mathbf{i} + \\ &\quad \left(w' \otimes y^\times - x' \otimes z^\times + y' \otimes w^\times + z' \otimes x^\times\right) \mathbf{j} + \\ &\quad \left(w' \otimes z^\times + x' \otimes y^\times - y' \otimes x^\times + z' \otimes w^\times\right) \mathbf{k} \end{aligned} \tag{8}$$

where $\otimes$ denotes element-wise multiplication of complex vector elements. Each biquaternions in $\mathbf{S}_\tau$ and $\mathbf{T}_\tau$ represents a combination of circular rotation, hyperbolic rotation, and scaling. This computation enhances potential interactions between timestamps and relations. We incorporate temporal information into relations via time-specific translation $\mathbf{S}_\tau$ and scaling $\mathbf{T}_\tau$. These operations are exclusively applied to the relation-specific scaling operation $\mathbf{T}_r$. Importantly, we perform translation first, followed by scaling, as the sequence can impact results. For the transformation operation $\mathbf{T}_{r\tau}$, we omit time information to capture time-invariant characteristics of the relation, thus yielding a relation-specific operation with integrated temporal information.

Next, we apply rotation operations to the entity using $\mathbf{R}_r$ for adjustment. This rotation is applied to the head entity, calculated as:

$$\mathbf{Q}_s^{\text{rot}} = \mathbf{Q}_s \cdot \mathbf{R}_r \tag{9}$$

where $\mathbf{Q}_s^{\text{rot}} \in \mathbb{R}^{2\times k}$.Subsequently, we adjust the head entity using $\mathbf{S}_\tau$ and $\mathbf{T}_\tau$ by translating and scaling it, given by:

$$\mathbf{Q}_s^{\mathrm{r}\tau} = (\mathbf{T}_{r\tau} \cdot \mathbf{Q}_s^{\mathrm{rot}}) \circledast \mathbf{S}_r \tag{10}$$

Thus, we obtain $\mathbf{Q}_s^{\mathrm{r}\tau} \in \mathbb{R}^{2\times k}$, an entity embedding that fuses time and relation information. Through these operations, TCompoundQ captures the compound geometric characteristics of relations and temporal information in dual quaternion space with enhanced accuracy. The scoring function of TCompoundQ measures similarity between head and tail entities, formulated as:

$$\phi(s, r, o, \tau) = \langle \mathbf{Q}_s^{\mathrm{r}\tau}, \mathbf{Q}_o \rangle \tag{11}$$

4.2 Loss Function

Following TNTComplEx [12], we adopt reciprocal learning to simplify the training process. The loss function is defined as:

$$\begin{aligned} \mathcal{L}_\omega = &- \log \left(\frac{\exp(f(s, r, o, t))}{\sum_{s' \in \mathcal{E}} \exp(f(s', r, o, t))} \right) \\ &- \log \left(\frac{\exp(f(o, r^{-1}, s, t))}{\sum_{o' \in \mathcal{E}} \exp(f(o', r^{-1}, s, t))} \right) \\ &+ L_\mu \end{aligned} \tag{12}$$

To counter overfitting, we follow previous work [6] and add an N3 regularization term to our loss function as follows:

$$L_\mu = \sum_{\{h,r,o\} \in \mathcal{TKG}} \lambda_\mu \left(\lambda_1 \left(\|\mathbf{Q}_h\|_3^3 + \|\mathbf{Q}_o\|_3^3 \right) + \lambda_2 \|\mathbf{Q}_r\|_3^3 \right) \tag{13}$$

where $\lambda_\mu, \lambda_1, \lambda_2$ are global, entity, and relation regularization hyperparameters, respectively. $|| \cdot ||_3$ represents the L_3 norm of the vector.

4.3 Temporal Regularization

Assuming temporal proximity in embeddings, TComplEx [12] introduces a smoothing regularizer that enforces temporal continuity between embeddings at successive time intervals:

$$\mathcal{L}_\tau = \sum_{i=1}^{n_\tau - 1} \|\mathbf{Q}_{\tau_{i+1}} - \mathbf{Q}_{\tau_i}\|_p^p, \tag{14}$$

where $p = 3$ is specified for the N3 regularization.

The complete loss function is therefore formulated as:

$$\mathcal{L} = \mathcal{L}_\omega + \lambda_\tau \mathcal{L}_\tau, \tag{15}$$

where λ_τ represents the temporal regularization coefficient.

4.4 Relation pattern

In previous TKGC models [17],[19], five types of relation patterns have been proposed, and their definitions are as follows:
Definition 1. A relation r is symmetric, if $\forall s, o, \tau, r(s, o, \tau) \wedge r(o, s, \tau)$ hold true.
Definition 2. A relation r is asymmetric, if $\forall s, o, \tau, r(s, o, \tau) \wedge \neg r(o, s, \tau)$ hold true.
Definition 3. A relation r_1 is inverse of relation r_2, if $\forall s, o, \tau, r_1(s, o, \tau) \wedge r_2(o, s, \tau)$ hold true.
Definition 4. A relation r_1 and relation r_2 are evolving over time from timestamp τ_1 to timestamp τ_2, if $\forall s, o, \tau, r_1(s, o, \tau_1) \wedge r_2(o, s, \tau_2)$ hold true.

TCompoundQ is capable of modeling the aforementioned relation patterns. All propositions are enumerated as follows, and the detailed proofs are presented in the Appendix.
Proposition 1. TCompoundQ can model the symmetric relation pattern. (See proof in Appendix 1)
Proposition 2. TCompoundQ can model the asymmetric relation pattern. (See proof in Appendix 2)
Proposition 3. TCompoundQ can model the inverse relation pattern. (See proof in Appendix 3)
Proposition 4. TCompoundQ can model the compositional relation pattern. (See proof in Appendix 4)

5 Experiment

5.1 Datasets

To evaluate our proposed model, we use five widely adopted temporal knowledge graph datasets: ICEWS14, ICEWS05-15 [9], GDELT, YAGO11k [9], and Wikidata12k [8]. ICEWS14 and ICEWS05-15 are subsets of the Integrated Crisis Early Warning System (ICEWS) dataset [25], which records temporal sociopolitical events starting from 1995. ICEWS14 contains events from 2014, while ICEWS05-15 includes events from 2005 to 2015. The GDELT dataset is part of the Global Database of Events, Language, and Tone (GDELT) [27]. It records daily event data with timestamps from April 1, 2015, to March 31, 2016. It consists of 500 entities and 20 relations. For YAGO11k and Wikidata12k, we simplified the datasets by removing specific month and day details to reduce complexity. Additionally, we addressed imbalances in the number of facts across time intervals. We merged adjacent years with fewer facts into single periods, ensuring that each interval contains at least 300 facts. This adjustment balances the datasets and improves model training quality. The detailed statistics for each dataset are provided in Table 1.

5.2 Evaluation Metrics

We utilize established evaluation metrics, specifically the Mean Reciprocal Rank (MRR) and Hits@n. Hits@n quantifies the proportion of accurate entities within

Table 1: Statistics of the experimental datasets.

Datasets	Entities	Relations	Timestamps	Time Span	Granularity	Training	Validation	Test
ICEWS14	6,869	230	365	A.D.2014	1 day	72,826	8,941	8,963
ICEWS05-15	10,094	251	4,017	A.D.2005-A.D.2015	1 day	368,962	46,275	46,092
GDELT	500	20	366	A.D.2015-A.D.2016	1 day	2,735,685	341,961	341,961
YAGO11k	10,623	10	237	453 B.C.-A.D.2844	1 year	16,406	2,050	2,051
Wikidata12k	12,554	24	232	A.D.1709-A.D.2018	1 year	539,286	67,538	63,110

the top n forecasts, serving as a robust indicator of our model's predictive accuracy. We calculate these metrics at various thresholds, n = 1, 3, and 10. In alignment with notable precedents in the field [21] , our evaluation method involves scoring potential quadruples (s', r, o, τ) and (s, r, o', τ). This process entails replacing either the subject or object entities with alternatives from the knowledge graph and assessing how well the model predicts these replacements. We then rank these alternatives based on their computed scores, applying a temporal filter to refine our analysis.

5.3 Baselines

We compare our proposed method with a range of Temporal Knowledge Graph Completion methods, including TTransE [7], TA-DistMult [9], HyTe [8], DE-SimplE [14], TComplEx and TNTComplEx [21], ATiSE [22], TeRo [1], ChronoR [2], TeLM [23], BoxTE [20], RotateQVS [3], TuckERTNT [26], TLT-KGE($\mathbb{C}$), TLT-KGE($\mathbb{Q}$) [24], TeAST [17], BDME [28], SANe [29], HGE [18], CEC-BD [30].

5.4 Experiment Setup

We conducted experiments using the PyTorch deep learning framework. All experiments were performed on an NVIDIA GeForce RTX 3090 with 24 GB of memory. We utilized the Adagrad optimizer and conducted a grid search to find the optimal hyperparameters based on performance on the validation set. The following parameter settings were used: a batch size of 3000 and an embedding rank of 6000. Our experiments were designed to run for up to 150 epochs, with a learning rate of 0.05. We disregarded variance in our results, as adjusting the random seed consistently produced stable outcomes, indicating reliable performance across different experimental runs. The optimal hyperparameters for TCompoundQ are as follows: ICEWS14: $\lambda_\mu = 0.003, \lambda_\tau = 0.003$, ICEWS05-15: $\lambda_\mu = 0.0000006, \lambda_\tau = 0.006$, GDELT: $\lambda_\mu = 0.0001, \lambda_\tau = 0.05$, YAGO11k: $\lambda_\mu = 0.05, \lambda_\tau = 0.005$, Wikidata12k: $\lambda_\mu = 0.06, \lambda_\tau = 0.005$. Our code is available at `https://github.com/AAristotle/TCompoundQ`.

5.5 Main Results

The experimental results on the ICEWS14, ICEWS05-15, GDELT, YAGO11k, and Wikidata12k datasets are shown in Table 2. Our model achieves the best per-

Table 2: Link prediction results on ICEWS05-15, ICEWS14, GDELT, YAGO11k, and Wikidata12k datasets. All results are reported from the original papers. **Bold** indicates the best result, underline indicates the second-best result. Dashes indicate unobtainable results. Results marked with ♣ are reported in HGE.

Method	**ICEWS05-15**				**ICEWS14**				**GDELT**				**YAGO11k**				**Wikidata12k**			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TTansE	27.1	8.4	-	61.6	25.5	7.4	-	60.1	11.5	0.0	16.0	31.8	10.8	2.0	15.0	25.1	17.2	9.6	18.4	32.9
TA-DistMult	47.4	34.6	-	72.8	47.7	36.3	-	68.6	20.6	12.4	21.9	36.5	15.5	9.8	-	26.7	23.0	13.0	-	46.1
HyTe	31.6	11.6	44.5	68.1	29.7	10.8	41.6	65.5	11.8	0.0	16.5	32.6	13.6	3.3	-	29.8	18.0	9.8	19.7	33.3
DE-SimplE	51.3	39.2	57.8	74.8	52.6	41.8	59.2	72.5	23.0	14.1	24.8	40.3	-	-	-	-	-	-	-	-
TComplEx	66.0	59.0	71.0	80.0	61.0	53.0	66.0	76.0	34.0	24.9	36.1	49.8	18.5	12.7	18.3	30.7	33.1	23.3	35.7	53.9
TNTComplEx	67.0	59.0	71.0	81.0	62.0	52.0	66.0	76.0	34.9	25.8	37.3	50.2	18.0	11.0	-	31.2	30.1	19.7	-	50.6
ATiSE	51.9	37.8	60.6	79.4	55.0	43.6	62.9	75.0	-	-	-	-	18.5	12.6	18.9	30.1	25.2	14.8	28.8	46.2
TeRo	58.6	46.9	66.8	79.5	56.2	46.8	62.1	73.2	24.5	15.3	26.4	42.0	18.7	12.1	19.7	31.9	29.9	19.8	32.9	50.7
ChronoR	67.5	59.6	72.3	82.0	62.5	54.7	66.9	77.3	-	-	-	-	-	-	-	-	-	-	-	-
TeLM	67.8	59.9	72.8	82.3	62.5	54.5	67.3	77.4	35.0	26.1	37.5	50.4	19.1	12.9	19.4	32.1	33.2	23.1	36.0	54.2
BoxTE	66.7	58.2	71.9	82.0	61.3	52.8	66.4	76.3	35.3	26.9	37.7	51.1	-	-	-	-	-	-	-	-
RotateQVS	63.3	52.9	70.9	81.3	59.1	50.7	64.2	75.4	27.0	17.5	29.3	45.8	-	-	-	-	-	-	-	-
TuckERTNT	63.8	55.9	68.6	78.3	60.4	52.1	65.5	75.3	38.1	28.3	40.1	54.4	-	-	-	-	-	-	-	-
TLT-KGE(ℂ)	68.6	60.7	73.5	83.1	63.0	54.9	67.8	77.7	35.6	26.7	38.5	53.2	-	-	-	-	-	-	-	-
TLT-KGE(ℚ)	69.0	60.9	74.1	**83.5**	63.4	55.1	68.4	78.6	35.8	26.5	38.8	54.3	-	-	-	-	-	-	-	-
LCGE ♣	61.8	51.4	68.1	81.2	61.6	53.2	66.7	77.5	-	-	-	-	-	-	-	-	-	-	-	-
TeAST	68.3	60.4	73.2	82.9	63.7	56.0	68.2	78.2	37.1	28.3	40.1	54.4	-	-	-	-	-	-	-	-
BDME	-	-	-	-	63.5	55.5	68.3	77.8	27.8	19.1	29.9	44.8	-	-	-	-	-	-	-	-
SANe	68.3	60.5	73.4	82.3	63.8	55.8	68.8	78.2	-	-	-	-	-	-	-	-	-	-	-	-
HGE(TLT-KGE)	68.8	60.8	74.0	**83.5**	63.4	55.0	68.5	78.8	37.1	27.7	40.2	55.6	-	-	-	-	-	-	-	-
CEC-BD	68.1	60.2	73.0	82.5	63.3	55.4	68.0	77.7	29.6	20.1	33.4	46.5	**21.2**	**15.4**	21.5	33.9	33.9	24.1	36.9	54.3
TCompoundQ	**69.2**	**61.4**	**74.0**	83.4	**64.5**	**56.6**	**69.2**	**79.2**	**51.4**	**44.8**	**54.3**	**63.7**	20.7	13.9	**21.0**	**35.3**	**34.2**	**24.0**	**37.2**	**55.9**

formance on the ICEWS05-15 dataset. TLT-KGE(ℚ), another model that uses quaternion embeddings, ranks second. This result highlights the effectiveness of quaternion embeddings in temporal knowledge graph representation.

On the ICEWS14 dataset, TCompoundQ achieves a 0.7% improvement in MRR over the SANE model, demonstrating its competitive advantage. For the GDELT dataset, which contains dense factual data, TCompoundQ shows significant performance gains, highlighting its ability to handle high-density information effectively.

The YAGO11k dataset contains many 1-N and N-1 relationships. In these cases, CEC-BD outperforms TCompoundQ, indicating its strength in handling such relationship types. However, on the Wikidata12k dataset, TCompoundQ achieves the highest performance. Since Wikidata12k has diverse entity types but lacks fine-grained temporal details (e.g., no monthly or daily timestamps), the performance of different methods remains relatively consistent, with only minor variations.

5.6 Ablation Studies

To assess the impact of different components on model performance, we conducted detailed ablation experiments on four datasets, with the results presented in Table 3.

Removing $\mathbf{S}_\tau$ and $\mathbf{T}_\tau$ eliminates the translation and scaling operations for relation embeddings, resulting in a noticeable performance drop compared to the original model. This effect is particularly evident on the ICEWS14 dataset, where translation plays a crucial role. Additionally, removing $\mathbf{Q}_\tau$, which encodes temporal information, leads to a significant decline in TCompoundQ's accuracy for

Table 3: Performance comparison of different variants on ICEWS14 and YAGO11k datasets. Bold values indicate the best performance.

		ICEWS14				YAGO11k			
#	Model	MRR	Hits@1	Hits@3	H@its10	MRR	Hits@1	Hits@3	Hits@10
0	TCompoundQ	**64.5**	**56.6**	**69.2**	**79.2**	**20.7**	**13.9**	**21.0**	**35.3**
1	w/o. $\mathbf{S}_\tau$	63.5	55.5	68.0	78.6	19.9	13.2	20.1	34.2
2	w/o. $\mathbf{T}_\tau$	64.1	55.5	69.2	79.7	18.2	11.8	18.1	32.1
3	w/o. $\mathbf{Q}_\tau$	47.2	35.3	53.2	70.9	16.5	10.3	16.5	28.9
4	w/o. $\mathbf{R}_r$	64.0	56.0	68.7	79.0	20.4	13.6	20.4	35.1
5	w/o. $\mathbf{T}_r$	63.3	55.4	67.8	78.4	19.7	13.1	19.8	33.9
6	w/o. $\mathbf{S}_r$	44.9	32.9	51.2	68.5	15.1	8.9	15.3	27.3
7	w/o. $\mathbf{Q}_r$	45.3	33.2	51.5	69.1	14.8	9.48	14.5	25.8
8	w/o. $\mathbf{T}_\tau \circledast \mathbf{T}_r$	63.9	55.6	68.8	79.1	19.7	13.1	20.1	33.7
9	w/o. $\mathbf{Q}_s^{\text{rot}} \circledast \mathbf{S}_r$	63.6	55.5	68.5	78.7	18.1	12.3	18.1	30.1
10	w/o. 8 and 9	63.4	55.4	68.0	78.0	18.2	12.3	18.0	30.2

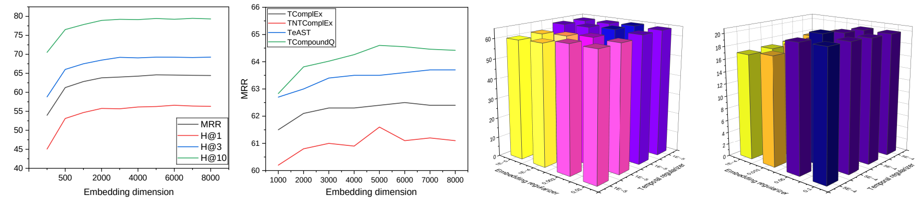
entity prediction. This highlights the importance of incorporating time-specific information.

The component $\mathbf{R}_r$ is responsible for relation rotation, but its impact on performance remains relatively minor across both datasets. This suggests that rotation provides fine-grained semantic adjustments rather than major structural modifications. In contrast, $\mathbf{T}_r$ and $\mathbf{S}_r$, which handle relation translation and scaling, play more critical roles. Notably, relation scaling ($\mathbf{S}_r$) significantly enhances TCompoundQ's performance by embedding temporal dependencies into the model. The exclusion of any of these components results in reduced performance, underscoring their unique contributions. The best results are achieved when all components are combined.

Removing the Hamilton product $\mathbf{T}_\tau \circledast \mathbf{T}_r$ and replacing it with standard multiplication eliminates the quaternion-based interaction between time and relation. Similarly, excluding $\mathbf{Q}_s^{\text{rot}} \circledast \mathbf{S}_r$ disrupts the quaternion interaction between the relation and the head entity. Both modifications lead to a substantial performance decline across the datasets.The impact is particularly pronounced on the YAGO11k dataset, which lacks fine-grained temporal data, as it only records time at the yearly level without monthly or daily timestamps. This coarse granularity further amplifies the effect of removing quaternion operations, resulting in a more significant drop in performance.

5.7 Model Efficiency

We investigate the effect of varying embedding dimensions on model performance. We selected embedding dimensions from the set {100, 500, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000} and conducted experiments on the ICEWS14 and YAGO11k datasets. The results are shown in Figure 2. From the figure, it is evident that on the ICEWS14 dataset, performance initially increases, with 500 dimensions marking a turning point, and near-peak performance achieved around 2000 dimensions. Figure 2 (b) compares the performance of the TCompoundQ



(a) (b) (c) (d)

Fig. 2: Performance variations across different embedding dimensions. (a) Performance on the ICEWS14 dataset as embedding dimensions vary. (b) Performance comparison of TCompoundQ with TComplEx, TNTComplEx, and TeAST models at equivalent embedding dimensions.(c) Results on ICEWS14 dataset with varying weight combinations. (d) Results on YAGO11k dataset under different weight combinations.

model with TComplEx, TNTComplEx, and TeAST models at the same embedding dimensions. Under equivalent dimensions, TCompoundQ achieves superior performance.

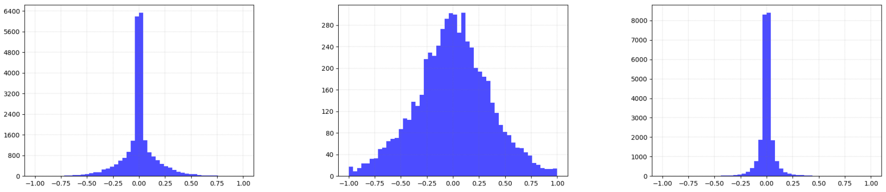
We further investigate the effects of different regularization weights for both the embedding and temporal regularizers, with results illustrated in Figure 2 (c) and 2 (d). On the ICEWS14 dataset, the overall performance remains relatively stable with minor fluctuations, indicating resilience to changes in regularization weight. On the YAGO11k dataset, performance drops significantly when the embedding regularizer weight is set below 0.001. In contrast, the temporal regularizer shows minimal impact on the performance, suggesting that temporal regularization is less critical for this dataset.

5.8 Visualization

We visualized the embeddings of the "Consult" relation in ICEWS14 by plotting histograms of translation, scaling, and rotation parameter values, as shown in Figure 3. Since "Consult" is a symmetric relation, its translation values should be close to zero. This expectation is confirmed in Figure 3(a). Figure 3(b) shows the rotation component of "Consult," which adjusts the semantics of the head entity. Additionally, "Consult" is an N-to-N relation. Therefore, we expect the compound operation to exhibit degeneracy. As shown in Figure 3(c), most of the scaling values are near zero. This observation supports our theoretical analysis of TCompoundQ's properties.

6 Conclusion

In this paper, we propose a novel approach that integrates translation, rotation, and scaling operations within the quaternion space for TKGC. Temporal information is incorporated by applying translation and scaling to relations,



(a) $\mathbf{T}_r$ (b) $\mathbf{R}_r$ (c) $\mathbf{S}_r$

Fig. 3: Distribution of entity embedding values and relation embedding values for the "Consult" relation in ICEWS14.

while translation, scaling, and rotation operations are applied to the head entity to model relational interactions. Experimental results demonstrate that TCompoundQ effectively captures both relational and temporal information in Temporal Knowledge Graphs. Different combinations of these operations yield varying levels of performance, indicating that selecting an optimal combination can significantly enhance model effectiveness. Future research could further explore the impact of translation, scaling, and rotation operations in three-dimensional space on model performance. Additionally, investigating the adaptability of these transformations in different geometric settings may offer further improvements in temporal knowledge graph reasoning.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 72210107001, the Beijing Natural Science Foundation under Grant No. IS23128, the Fundamental Research Funds for the Central Universities, and the CAS PIFI International Outstanding Team Project (Grant No. 2024PG0013).

References

1. 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.
2. 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, 2021, pp. 6471–6479.
3. 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.

4. Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, “Dual quaternion knowledge graph embeddings,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 6894–6902.
5. X. Ge, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, “Compounde: Knowledge graph embedding with translation, rotation and scaling compound operations,” *arXiv e-prints*, pp. arXiv–2207, 2022.
6. J. Guo and S. Kok, “Bique: Biquaternionic embeddings of knowledge graphs,” *arXiv preprint arXiv:2109.14401*, 2021.
7. 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.
8. S. S. Dasgupta, S. N. Ray, and P. Talukdar, “Hyte: Hyperplane-based temporally aware knowledge graph embedding,” in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 2001–2011.
9. A. Garcia-Duran, S. Dumančić, and M. Niepert, “Learning sequence encoders for temporal knowledge graph completion,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4816–4821.
10. M. Nickel, L. Rosasco, and T. Poggio, “Holographic embeddings of knowledge graphs,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.
11. 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.
12. T. Lacroix, G. Obozinski, and N. Usunier, “Tensor decompositions for temporal knowledge base completion,” in *International Conference on Learning Representations.* Brussels, Belgium: Association for Computational Linguistics, 2019, pp. 4816–4821.
13. C. Xu, M. Nayyeri, F. Alkhoury, H. Shariat Yazdi, and J. Lehmann, “Temporal knowledge graph embedding model based on additive time series decomposition,” *arXiv e-prints*, pp. arXiv–1911, 2019.
14. R. Goel, S. M. Kazemi, M. Brubaker, and P. Poupart, “Diachronic embedding for temporal knowledge graph completion,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 3988–3995.
15. P. Jain, S. Rathi, S. Chakrabarti *et al.*, “Temporal knowledge base completion: New algorithms and evaluation protocols,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3733–3747.
16. G. Niu and B. Li, “Logic and commonsense-guided temporal knowledge graph completion,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 4569–4577.
17. J. Li, X. Su, and G. Gao, “Teast: Temporal knowledge graph embedding via archimedean spiral timeline,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 15 460–15 474.
18. J. Pan, M. Nayyeri, Y. Li, and S. Staab, “Hge: Embedding temporal knowledge graphs in a product space of heterogeneous geometric subspaces,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 8913–8920.
19. H. Sun, J. Zhong, Y. Ma, Z. Han, and K. He, “Timetraveler: Reinforcement learning for temporal knowledge graph forecasting,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 8306–8319.

20. J. Messner, R. Abboud, and I. I. Ceylan, “Temporal knowledge graph completion using box embeddings,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 7779–7787.
21. T. Lacroix, N. Usunier, and G. Obozinski, “Canonical tensor decomposition for knowledge base completion,” in *International Conference on Machine Learning.* PMLR, 2018, pp. 2863–2872.
22. C. Xu, M. Nayyeri, F. Alkhoury, H. Yazdi, and J. Lehmann, “Temporal knowledge graph completion based on time series gaussian embedding,” in *The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part I 19.* Springer, 2020, pp. 654–671.
23. 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.
24. F. Zhang, Z. Zhang, X. Ao, F. Zhuang, Y. Xu, and Q. He, “Along the time: timeline-traced embedding for temporal knowledge graph completion,” in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2529–2538.
25. E. Boschee, J. Lautenschlager, S. O’Brien, S. Shellman, J. Starz, and M. Ward, “Icews coded event data,” *Harvard Dataverse*, vol. 12, 2015.
26. P. Shao, D. Zhang, G. Yang, J. Tao, F. Che, and T. Liu, “Tucker decomposition-based temporal knowledge graph completion,” *Knowledge-Based Systems*, vol. 238, p. 107841, 2022.
27. K. Leetaru and P. A. Schrodt, “Gdelt: Global data on events, location, and tone, 1979–2012,” in *ISA annual convention*, vol. 2. Citeseer, 2013, pp. 1–49.
28. L. Yue, Y. Ren, Y. Zeng, J. Zhang, K. Zeng, and J. Wan, “Block decomposition with multi-granularity embedding for temporal knowledge graph completion,” in *International Conference on Database Systems for Advanced Applications.* Springer, 2023, pp. 706–715.
29. Y. Li, X. Zhang, B. Zhang, and H. Ren, “Each snapshot to each space: Space adaptation for temporal knowledge graph completion,” in *International Semantic Web Conference.* Springer, 2022, pp. 248–266.
30. L. Yue, Y. Ren, Y. Zeng, J. Zhang, K. Zeng, J. Wan, and M. Zhou, “Complex expressional characterizations learning based on block decomposition for temporal knowledge graph completion,” *Knowledge-Based Systems*, vol. 290, p. 111591, 2024.
31. W. R. Hamilton, “Lxxviii. on quaternions; or on a new system of imaginaries in algebra: To the editors of the philosophical magazine and journal,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, pp. 489–495, 1844.

Appendix

Proof of Proposition 1

First, we define the score function as follows:

$$\begin{aligned}\phi(s,r,o,\tau) &= \langle \mathbf{Q}_s^{\mathrm{r}\tau}, \mathbf{Q}_o \rangle \\ &= \langle (\mathbf{S}_\tau \circledast (\mathbf{T}_r \cdot \mathbf{T}_\tau) \cdot \mathbf{Q}_s^{\mathrm{r}\tau} \cdot \mathbf{R}_r) \circledast \mathbf{S}_r, \mathbf{Q}_o \rangle \\ &= (\mathbf{S}_\tau \circledast (\mathbf{T}_r \cdot \mathbf{T}_\tau) \cdot \mathbf{Q}_s^{\mathrm{r}\tau} \cdot \mathbf{R}_r) \circledast \mathbf{S}_r \circ \mathbf{Q}_o\end{aligned} \tag{16}$$

Let $\mathbf{M} = \mathbf{S}_\tau \circledast (\mathbf{T}_r \cdot \mathbf{T}_\tau)$ and $\mathbf{K} = \mathbf{R}_r \circledast \mathbf{S}_r$. For symmetric patterns, according to Definition 1, we have $r(s, o, t) \wedge r(o, s, t)$. Therefore, we deduce:

$$\phi(s, r, o, \tau) = \phi(o, r, s, \tau) \tag{17}$$

From the above equations, it is clear that when $\mathbf{M}$ and $\mathbf{K}$ are invertible, the condition for symmetric patterns is satisfied:

$$\begin{gathered} \phi(s, r, o, \tau) = \phi(o, r, s, \tau) \Leftrightarrow \\ \mathbf{M} \cdot \mathbf{Q}_s^{\mathrm{r}\tau} \cdot \mathbf{K} \circ \mathbf{Q}_o = \mathbf{M} \cdot \mathbf{Q}_o^{\mathrm{r}\tau} \cdot \mathbf{K} \circ \mathbf{Q}_s \Leftrightarrow \\ \mathbf{Q}_s^{\mathrm{r}\tau} \circ \mathbf{Q}_o = \mathbf{K}^{-1}\mathbf{M}^{-1}\mathbf{M}\mathbf{Q}_o^{\mathrm{r}\tau} \circ \mathbf{Q}_s \end{gathered} \tag{18}$$

Thus, TCompoundQ can model symmetric patterns when the matrices $\mathbf{M}$ and $\mathbf{K}$ are invertible.

Proof of Proposition 2

For asymmetric patterns, according to Definition 2, we have:

$$r(s, o, t) \wedge \neg r(o, s, t) \tag{19}$$

Thus, it can be deduced that the matrices $\mathbf{M}$ and $\mathbf{K}$ are not invertible.

Proof of Proposition 3

For inverse patterns, according to Definition 3, we have $r_1(s, o, \tau) \wedge r_2(o, s, \tau)$. Hence, we get:

$$\begin{gathered} \phi(s, r_1, o, \tau) = \phi(o, r_1, s, \tau) \Leftrightarrow \\ \mathbf{Q}_s^{\mathrm{r}\tau} \circ \mathbf{Q}_o = \mathbf{K}_2^{-1}\mathbf{M}_2^{-1}\mathbf{M}_1\mathbf{K}_1\mathbf{Q}_o^{\mathrm{r}\tau} \circ \mathbf{Q}_s \Leftrightarrow \\ \mathbf{K}_2^{-1}\mathbf{M}_2^{-1}\mathbf{M}_1\mathbf{K}_1 = I \quad \text{or} \quad \mathbf{K}_1^{-1}\mathbf{M}_1^{-1}\mathbf{M}_2\mathbf{K}_2 = I \end{gathered} \tag{20}$$

Therefore, TCompoundQ can model inverse patterns when $\mathbf{M}_1\mathbf{K}_1$ and $\mathbf{M}_2\mathbf{K}_2$ are inverse matrices.

Proof of Proposition 4

For temporal evolution patterns, according to Definition 4, we have $r_1(s, o, \tau_1) \wedge r_2(s, o, \tau_2)$. Hence, the formula is as follows:

$$\begin{gathered} \phi(s, r_1, o, \tau_1) = \phi(o, r_2, s, \tau_2) \Leftrightarrow \\ \mathbf{M}_{11}\mathbf{K}_{11}\mathbf{Q}_s^{\mathrm{r}\tau} \circ \mathbf{Q}_o = \mathbf{M}_{22}\mathbf{K}_{22}\mathbf{Q}_o^{\mathrm{r}\tau} \circ \mathbf{Q}_s \end{gathered} \tag{21}$$

Then, we derive:

$$\begin{gathered} \mathbf{Q}_s^{\mathrm{r}\tau} \circ \mathbf{Q}_o = \mathbf{M}_{11}^{-1}\mathbf{K}_{11}^{-1}\mathbf{M}_{22}\mathbf{K}_{22}\mathbf{Q}_o^{\mathrm{r}\tau} \circ \mathbf{Q}_s \Rightarrow \\ \mathbf{M}_{11}^{-1}\mathbf{K}_{11}^{-1}\mathbf{M}_{22}\mathbf{K}_{22} = I \quad \text{or} \quad \mathbf{M}_{22}^{-1}\mathbf{K}_{22}^{-1}\mathbf{M}_{11}\mathbf{K}_{11} = I \end{gathered} \tag{22}$$

Therefore, TCompoundQ can model temporal evolution patterns when $\mathbf{M}_{11}\mathbf{K}_{11}$ and $\mathbf{M}_{22}\mathbf{K}_{22}$ are inverse matrices.