Simple but Effective Compound Geometric Operations for Temporal Knowledge Graph Completion

Rui Ying[1]   **Mengting Hu**[2] *   **Jianfeng Wu**[1]   **Yalan Xie**[1]   **Xiaoyi Liu**[1]
Zhunheng Wang[1]   **Ming Jiang**[1]   **Hang Gao**[3]   **Linlin Zhang**[4]   **Renhong Cheng**[1]

[1] College of Computer Science, Nankai University, [2] College of Software, Nankai University
[3] College of Artificial Intelligence, Tianjin University of Science and Technology
[4] China Automotive Technology and Research Center Co., Ltd.
ruiying@mail.nankai.edu.cn, mthu@nankai.edu.cn

Abstract

Temporal knowledge graph completion aims to infer the missing facts in temporal knowledge graphs. Current approaches usually embed factual knowledge into continuous vector space and apply geometric operations to learn potential patterns in temporal knowledge graphs. However, these methods only adopt a single operation, which may have limitations in capturing the complex temporal dynamics present in temporal knowledge graphs. Therefore, we propose a simple but effective method, i.e. TCompoundE, which is specially designed with two geometric operations, including time-specific and relation-specific operations. We provide mathematical proofs to demonstrate the ability of TCompoundE to encode various relation patterns. Experimental results show that our proposed model significantly outperforms existing temporal knowledge graph embedding models. Our code is available at https://github.com/nk-ruiying/TCompoundE.

1 Introduction

A knowledge graph (KG) comprises a collection of structured knowledge presented in triples, offering a simple and effective means of describing factual information (Tiwari et al., 2021; Liu et al., 2023). In a KG, a triplet is conventionally represented as $(s, \hat{r}, o)$, where s, o and $\hat{r}$ correspond to the head entity, tail entity and the relation linking between the head and tail entities, respectively. However, knowledge is not static in the real world (Li et al., 2023). Temporal knowledge graph (TKG) represents knowledge $(s, \hat{r}, o)$ occurring at timestamp τ, denoted as a quadruple $(s, \hat{r}, o, \tau)$, thereby adding a temporal dimension to knowledge graphs. The introduction of the timestamp τ enables TKG to delineate the temporal scope of knowledge more accurately and helps us better uncover the potential information within it (Zhang et al., 2022). Therefore, TKG is widely applied in downstream tasks

---
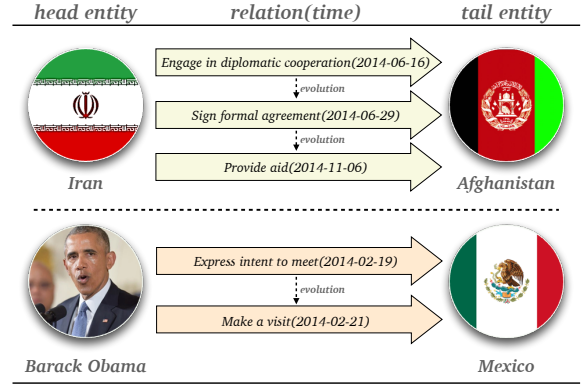* Mengting Hu is the corresponding author.

Figure 1: An illustration of temporal evolution patterns. It can be observed that the relationships between head entity and tail entity are dynamically determined by both relation and time.

such as question answering (Jia et al., 2018), information retrieval (Campos et al., 2014), and recommendation systems (Lathia et al., 2010) due to its temporal characteristics.

However, TKG usually does not cover all the facts. The incompleteness of TKG hinders the performance of its downstream tasks. To enhance the overall completeness of TKG, the temporal knowledge graph embedding (TKGE) model utilizes existing knowledge to predict and estimate missing facts. Specifically, the TKGE model employs distinct score functions to acquire effective vector space representations for entities, relations and timestamps, thus utilizing these representations to predict missing facts in TKG. Furthermore, how to enhance the expressive capabilities of the TKGE model is also an important issue that has received widespread attention.

Geometric operations such as translation and scaling are widely used operations in the field of graphics. These operations help to distinguish between these different classes of entities and to model different relational patterns. Models like TTransE (Leblay and Chekol, 2018), TComplEx

(Lacroix et al., 2020) and TNTComplEx (Lacroix et al., 2020) all use these operations and achieve good performance. Previous TKGEs often use a single type of operation for embedding. This approach is not conducive to modeling different relation patterns and temporal evolution patterns (shown in Fig. 1) since each operator may have modeling limitations. Ge et al. (2022) show that using complex geometry operations in knowledge graphs can effectively model different relational patterns. Inspired by them, we work on compound geometric operations that fit relationships and timestamps in TKGs.

In this paper, we present a model called **TCompoundE** based on CompoundE (Ge et al., 2022). Different from CompoundE's work, we discuss the effects of representing relations and timestamps as different geometric operations, which we call relation-specific operations and time-specific operations. More specifically, we use composite operations involving translation and scaling as relation-specific operations and time-specific operations. We integrate time-specific operations within the framework of relation-specific operations, aiming to capture both time-varying and time-invariant features within the low-dimensional representations of relations. Relation-specific operations, incorporating temporal information, are applied to the head entity embedding. Subsequently, the confidence of the quadruple is expressed by computing the semantic similarity between the head entity embedding and the tail entity embedding.

In summary, the main contributions of our work are as follows:

- We present a novel TKG embedding model called TCompoundE, which introduces relation-specific and time-specific compound geometric operations.

- We substantiate the suitability of our model for important relation patterns through mathematical formulations.

- Our experimental results on three benchmark datasets demonstrate that our method both meets and surpasses the performance of existing TKGE methods.

2 Related Work

2.1 Knowledge Graph Embedding

Alternatively referred to as static knowledge graph embedding, knowledge graph embedding com-

monly involves embedding entities and relations into low-dimensional vector spaces. This low-dimensional embedding is employed to enhance the representation of entities and relations through score functions. The model can be categorized into a translation model and a bilinear model based on the distinct score functions employed. **In translation model**, TransE (Bordes et al., 2013) originally introduces a straightforward and efficient score function $\phi(s, \hat{r}, o) = \|e_s + e_{\hat{r}} - e_o\|_2$, where $e_s, e_{\hat{r}}, e_o$ represent the lower dimensional embedding of head entity, relation and tail entity respectively, utilizing relations to denote the translation distance from head entities to tail entities. Subsequent models, including TransH (Wang et al., 2014), TransR (Lin et al., 2015) and TransD (Ji et al., 2015) adopt distinct projection strategies to implement a relation-specific representation of entities. RotatE (Sun et al., 2019) introduces the concept of relations into knowledge graph embedding through a rotation operation in complex space. The PairRE (Chao et al., 2020) model suggests implementing separate relation-specific scaling operations on the head and tail entities. HAKE (Zhang et al., 2020) operates by mapping the embeddings into polar coordinate space. CompoudE (Ge et al., 2022) employs a combination of translation, scaling and rotation to form a low-dimensional representation of the relation, addressing the limitations associated with individual operations. **In bilinear model**, DisMult (Yang et al., 2014) represents relationships through a symmetric matrix, deriving a scoring function $\phi(s, \hat{r}, o) = <e_s, W_{\hat{r}}, e_o>$ by evaluating the semantic similarity between head and tail entities, where $W_{\hat{r}}$ represents the symmetric matrix of relation. ComplEx (Trouillon et al., 2016) operates within complex spaces. TuckER (Balazevic et al., 2019) employs Tucker decomposition to assess the plausibility of fact triples.

2.2 Temporal Knowledge Graph Embedding

Temporal knowledge graph embedding closely resembles static knowledge graph embedding. Therefore, the majority of TKGE are adapted from knowledge graph embeddings to accommodate dynamic changes in facts. For instance, TTransE (Leblay and Chekol, 2018) represents timestamp through a translation operation, following the concept introduced by TransE (Bordes et al., 2013). Its score function is denoted by $\phi(s, \hat{r}, o, \tau) = \|e_s + e_{\hat{r}} + e_{\tau} - e_o\|_2$, where e_{τ} represents the lower-dimensional embedding of timestamp. TA-

DistMult (García-Durán et al., 2018) integrates the timestamp through the application of the DistMult (Yang et al., 2014) method. TComplEx (Lacroix et al., 2020) and TNTComplEx (Lacroix et al., 2020) expand the tensor decomposition methodology introduced by the ComplEx (Trouillon et al., 2016) model to encompass four dimensions, facilitating the embedding of TKGs. ChronoR (Sadeghian et al., 2021) employs the rotation operation from RotatE (Sun et al., 2019) to comprehend the influence of the combination between timestamp and relation on the entity's rotation operation. RotateQVS (Chen et al., 2022) employs QuatE (Zhang et al., 2019) quaternion rotation to depict the temporal impact on entities. BoxTE (Messner et al., 2022) embeds TKGs using the box embedding model, BoxE (Abboud et al., 2020). **In addition to this**, DE-Simple (Goel et al., 2020) incorporates diachronic embedding, expanding the static knowledge graph embedding methods into TKGE models capable of handling temporal information. TeAST (Li et al., 2023) employs timestamp as a mapping operation for the Archimedean spiral axis, associating relations with the spiral axis to capture the dynamic evolution of relations over time.

3 Background and Notation

In subsequent sections of this article, the CompoundE (Ge et al., 2022) methodology is employed to elucidate translation, scaling, and rotation operations. The ensuing discussion provides a comprehensive account of the representation of these operations, followed by an introduction to the CompoundE model.

3.1 Translation, Rotation, and Scaling

Translation, rotation and scaling transformations are fundamental operations in graphics, frequently employed in various engineering applications. In the processing of robot motion (LaValle, 2006), the cascade application of translation, rotation and scaling operations constitutes a method for precise position determination. The translation, rotation and scaling operations are illustrated in Fig. 2. Expressing these operations in matrix form proves to be both more efficient and straightforward. The translation operation in Fig. 2 is represented in 2D as follows:

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$
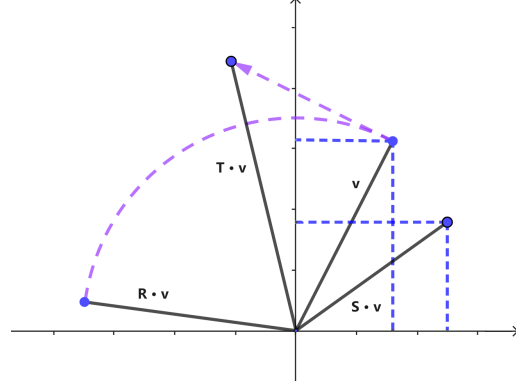

Figure 2: An illustration of translation, rotation and scaling. Where v is the original vector, $T \cdot v$ is the vector after the translation operation, $R \cdot v$ is the vector after the rotation operation, and $S \cdot v$ is the vector after scaling.

while 2D rotation matrix can be written as:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

And 2D scaling matrix can be expressed as:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

3.2 CompoundE

CompoundE (Ge et al., 2022) encompasses a range of models that integrate various translation, rotation and scaling operations. In this discussion, we focus on a specific combination approximating our model. This involves applying a blend of translation, rotation and scaling to the head entity. The ultimate score is determined by computing the distance between the transformed head and the initial tail entity embeddings. The corresponding score function is formally defined as follows:

$$\phi(s, \hat{r}, o) = \|T_{\hat{r}} \cdot R_{\hat{r}} \cdot S_{\hat{r}} \cdot e_s - e_o\| \quad (4)$$

where $S_{\hat{r}}, R_{\hat{r}}, T_{\hat{r}}$ denote the translation, rotation and scaling operations for the head entity embedding. These constituent operators are specific to relations. It is essential to highlight that the scaling, rotation and translation operations employed here are sequential, and altering their order yields distinct outcomes (Ge et al., 2022).

3.3 Problem formulation

For a temporal knowledge graph $\mathcal{G}$, we use $\mathcal{E}$ to denote the set of entities. $\mathcal{R}$ and $\mathcal{T}$ represent the set of relations and the set of timestamps in the temporal knowledge graph respectively. A fact in the temporal knowledge graph is represented by the quadruple $(s, \hat{r}, o, \tau)$, where $s, o \in \mathcal{E}$, $\hat{r} \in \hat{\mathcal{R}}$ and $\tau \in \mathcal{T}$. The task of knowledge graph completion is to predict the missing facts through the existing facts in the knowledge graph. We train the data in the training set through the score function $\phi(s, \hat{r}, o, \tau)$ in the model. When predicting a fact, we give either the head entity, relation and timestamp or the tail entity, relation and timestamp in the quadruple. We input these missing quadruples $(s, \hat{r}, ?, \tau)$ or $(?, \hat{r}, o, \tau)$ and candidate entities into the score function, taking the highest score to form a new fact quadruple for the entity.

4 Methodology

4.1 TCompoundE Model

In this section, we present our model, TCompoundE, which employs compound geometric operations on both relations and timestamps. For a quadruple $(s, \hat{r}, o, \tau)$ in TKG. We utilize the notations e_s, e_o to represent the embeddings of the head entity s and tail entity o. We utilize $S_{\hat{r}}$ and $T_{\hat{r}}$ to represent relation-specific scaling, translation operations. Integrate temporal information into relation-specific operations before applying them to entity embeddings. This merging involves time-specific translation T_τ and scaling S_τ operations in a relationship-specific process. In our model, we employ translation and scaling operations to represent relation-specific operations and time-specific operations. To facilitate a comprehensive introduction to our model, we categorize it into two distinct sections: **Time-Specific Operation** and **Relation-Specific Operation**; The initial section elucidates the utilization of time-specific operations in conjunction with relation-specific operations within a quadruple. In the subsequent section, we elaborate on the impact of relation-specific operations on the embedding of the head entity.

Time-Specific Operation. In our model, we employ time-specific translation T_τ and scaling S_τ operations to imbue temporal information into the relation. We exclusively apply these operations to the relation-specific scaling operation $S_{\hat{r}}$. It is crucial to highlight that we scale the relation-specific operation by first applying translation and then scal-

ing. This sequencing is intentional, as the order of operations can influence the outcome. However, for the relation-specific translation operation, we refrain from integrating time information. This approach aims to capture features of relations that remain constant over time. Subsequently, we obtain relation-specific operations that integrate temporal information. Herein, $S_{\hat{r}\tau}$ and $T_{\hat{r}\tau}$ denote the relation-specific scaling and translation operations, respectively, after incorporating time information. These operations can be precisely described by the following formula:

$$S_{\hat{r}\tau} = S_\tau \cdot T_\tau \cdot S_{\hat{r}} \tag{5}$$

$$T_{\hat{r}\tau} = T_{\hat{r}} \tag{6}$$

Relation-Specific Operation. We denote the relation-specific translation and relation-specific scaling operations for the head entity as $T_{\hat{r}}$ and $S_{\hat{r}}$ respectively. To capture temporal information within the TKG, we refrain from applying these operations directly to the head entity embeddings. Instead, we execute relation-specific operations subsequent to the time-specific operations. Specifically, we utilize $S_{\hat{r}\tau}$ and $T_{\hat{r}\tau}$ to conduct relation-specific operations incorporating time information on the head entity embedding. This operation is formally represented as:

$$e_s^{\hat{r}\tau} = S_{\hat{r}\tau} \cdot T_{\hat{r}\tau} \cdot e_s \tag{7}$$

We obtain the head entity representation $e_s^{\hat{r}\tau}$ incorporating fused time and relation information using Formula 7. Unlike CompoundE (Ge et al., 2022), our chosen score function is not a distance metric; instead, it is determined by the semantic similarity between $e_s^{\hat{r}\tau}$ and the tail entity e_o. This decision is grounded in our belief that semantic similarity offers more advantages than distance metrics in the context of TKG *(proof in Appendix G)*. The score function for TCompoundE is expressed as:

$$\phi(s, \hat{r}, o, \tau) = <e_s^{\hat{r}\tau}, e_o> \tag{8}$$

4.2 Loss Function

Building upon TNTComplEx (Lacroix et al., 2020) and TeAST (Li et al., 2023), we adopt reciprocal learning for training our model, with the loss func-

tion defined as follows:

$$\mathcal{L}_u = -\log\left(\frac{\exp(\phi(s, \hat{r}, o, \tau))}{\sum_{o' \in \varepsilon} \exp(\phi(s, \hat{r}, o', \tau))}\right)$$
$$- \log\left(\frac{\exp(\phi(o, \hat{r}^{-1}, s, \tau))}{\sum_{s' \in \varepsilon} \exp(\phi(o, \hat{r}^{-1}, s', \tau))}\right)$$
$$+ \lambda_u \sum_{i=1}^{k}(\|\boldsymbol{e_s}\|_3^3 + \|\boldsymbol{T_{\hat{r}\tau}} + \boldsymbol{S_{\hat{r}\tau}}\|_3^3 + \|\boldsymbol{e_o}\|_3^3)$$
(9)

where λ_u is the weight of the N3 regularization, and r^{-1} denotes the inverse relation. We use the smoothing temporal regularizer in TNTComplEx (Lacroix et al., 2020) to make neighboring timestamps having close representations. It is defined as:

$$\mathcal{L}_\tau = \frac{1}{N_\tau - 1} \sum_{i=1}^{N_\tau - 1} \|e_{\tau(i+1)} - e_{\tau(i)}\|_3^3 \quad (10)$$

The total loss function of TCompoundE is defined as:

$$\mathcal{L} = \mathcal{L}_u + \lambda_\tau \mathcal{L}_\tau \quad (11)$$

where λ_τ represents temporal regularization.

4.3 Modeling Various Relation Patterns

TCompoundE can model crucial relation patterns, encompassing symmetric, asymmetric, inverse and temporal evolution patterns (detail in Appendix A). We enumerate all the propositions in this section, with corresponding proofs provided in the Appendix.

Proposition 1 *TCompoundE can model the symmetric relation pattern. (proof in Appendix B)*

Proposition 2 *TCompoundE can model the asymmetric relation pattern. (proof in Appendix C)*

Proposition 3 *TCompoundE can model the inverse relation pattern. (proof in Appendix D)*

Proposition 4 *TCompoundE can model the temporal evolution relation pattern. (proof in Appendix E)*

5 Experiments

5.1 Datasets

We assess the performance of TCompoundE on three benchmark datasets for TKGE. The datasets include ICEWS14 and ICEWS05-15 (García-Durán et al., 2018), both derived from the Integrated Crisis Early Warning System (ICEWS)

	ICEWS14	ICEWS05 − 15	GDELT
#$\mathcal{E}$	7,128	10,488	500
#$\hat{\mathcal{R}}$	230	251	20
#$\mathcal{T}$	365	4017	366
#Train	72,826	386,962	2,735,685
#Dev	8,963	46,092	31,961
#Test	8,941	46,275	31,961

Table 1: Data statistics. #$\mathcal{E}$, #$\hat{\mathcal{R}}$ and #$\mathcal{T}$ denote the number of entities, the number of relations and the number of timestamps respectively.

(Lautenschlager et al., 2015). The ICEWS database system records significant political events, with ICEWS14 capturing events from the year 2014, and ICEWS05-15 encompassing major political events spanning from 2005 to 2015. Additionally, we evaluate TCompoundE on GDELT (Leetaru and Schrodt, 2013), a subset of the extensive Global Database of Events, Language, and Tone (GDELT) Temporal Knowledge Graph dataset. GDELT incorporates information from diverse sources, comprising facts with daily timestamps between April 1, 2015, and March 31, 2016. Notably, GDELT is limited to the 500 most common entities and 20 most frequent relations. More information about the datasets can be found in Table 1.

5.2 Baselines

We conduct a comprehensive comparison of our model with state-of-the-art Temporal Knowledge Graph Embedding (TKGE) models, which include TTransE (Leblay and Chekol, 2018), DE-SimplE (Goel et al., 2020), TA-DisMult (García-Durán et al., 2018), ChronoR (Sadeghian et al., 2021), TComplEx (Lacroix et al., 2020), TNTComplEx (Lacroix et al., 2020), BoxTE (Messner et al., 2022), RotateQVS (Chen et al., 2022), and TeAST (Li et al., 2023).

Among these existing TKGE methods, TeAST achieves state-of-the-art results on the ICEWS14, ICEWS05-15, and GDELT datasets. Consequently, we consider TeAST (Li et al., 2023) as the primary baseline for our comparative analysis. In addition, we also used a variant of the TCompoundE model for ablation experiments. A detailed description of the variant model can be found in Appendix F.

5.3 Evaluation Protocol

This paper evaluates our temporal knowledge graph embedding (TKGE) model using the aforementioned benchmarks. Following established base-

Methods	ICEWS14				ICEWS05 − 15				GDELT			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TTransE	0.255	0.074	-	0.601	0.271	0.084	-	0.616	0.115	0.0	0.160	0.318
DE-SimplE	0.526	0.418	0.592	0.725	0.513	0.392	0.578	0.748	0.230	0.141	0.248	0.403
TA-DisMult	0.477	0.363	-	0.686	0.474	0.346	-	0.728	0.206	0.124	0.219	0.365
ChronoR	0.625	0.547	0.669	0.773	0.675	0.596	0.723	0.820	-	-	-	-
TComplEx	0.610	0.530	0.660	0.770	0.660	0.590	0.710	0.800	0.340	0.294	0.361	0.498
TNTComplEx	0.620	0.520	0.660	0.760	0.670	0.590	0.710	0.810	0.349	0.258	0.373	0.502
BoxTE	0.613	0.528	0.664	0.763	0.667	0.582	0.719	0.820	0.352	0.269	0.377	0.511
RotateQVS	0.591	0.507	0.642	0.754	0.633	0.529	0.709	0.813	0.270	0.175	0.293	0.458
TeAST	0.637	0.560	0.682	0.782	0.683	0.604	0.732	0.829	0.371	0.283	0.401	0.544
TCompoundE	**0.644**	**0.561**	**0.694**	**0.795**	**0.692**	**0.612**	**0.743**	**0.837**	**0.433**	**0.347**	**0.469**	**0.595**

Table 2: Knowledge graph completion results on ICEWS14, ICEWS05-15 and GDELT. ALL results are taken from the original papers.

lines, we assess the ranking quality of each test triplet by computing all possible substitutions for both the head and tail entities. Specifically, for a missing quadruple $(s, \hat{r}, ?, \tau)$ or $(?, \hat{r}, o, \tau)$, we calculate the scores for all entities and rank them accordingly. We then filter out other correct quadruples to make predictions about the current missing entity. Performance is evaluated using standard metrics, including mean reciprocal rank (MRR) and Hits@n. Hits@n measures the percentage of correct entities within the top n predictions. Higher values of MRR and Hits@n indicate superior performance. The Hits ratio is calculated with cut-off values n = 1, 3, 10, denoted as H@n for convenience in this paper.

5.4 Experimental Setup

We implement our proposed model TCompoundE via pytorch based on the TeAST training framework. All experiments are trained on a single NVIDIA RTX A6000 with 48GB memory. We use the Adagrad optimizer and conduct a lot of experiments to find the optimal parameter configuration on each dataset. The learning rate is set to 0.01 and the embedding dimension d is set to 6000 and the batch size is set to 4000 in the ICEWS14 dataset and the max epoch is set to 400. In the ICEWS05-15 dataset, we set the learning rate to 0.08, d to 8000, the batch size to 6000 and the max epoch to 100. In GDELT, the learning rate, d, the batch size and the max epoch are set to 0.35, 6000, 2000 and 50 respectively. The optimal hyperparameters for TCompound are as follows:

- **ICEWS14**: $\lambda_u = 0.0025$, $\lambda_\tau = 0.01$

- **ICEWS05-15**: $\lambda_u = 0.002$, $\lambda_\tau = 0.1$

- **GDELT**: $\lambda_u = 0.001$, $\lambda_\tau = 0.001$

We report the average results on the test set for five runs.

6 Results and Analysis

6.1 Main Results

Table 2 presents the results of knowledge graph completion for ICEWS14, ICEWS05-15 and GDELT datasets. The best-performing results are highlighted in bold font. Our observations indicate that TCompoundE outperforms all baseline models across all metrics for t datasets. As TCompoundE exclusively applies the time-specific operation to the relation-specific scaling operation, this enables relations occurring simultaneously to utilize the same time-specific operation, facilitating the evolution of all relations over time. For the relation-specific translation operation, we maintain the timestamps unchanged to preserve the features of relations that remain constant over time. This demonstrates the significance of applying the time-specific operation exclusively to the relation-specific scaling operation. Additionally, Table 2 reveals that TCompoundE has achieved significant improvements over TKGE models utilizing a single operation for both relation-specific and time-specific operations, such as translation (TTransE) and scaling (TComplEx). This confirms that employing compound geometric operations for both relation-specific and time-specific operations is an effective strategy for designing TKG embeddings. Furthermore, BoxTE (Messner et al., 2022) highlights that GDELT necessitates a considerable level of temporal inductive capacity to achieve effective encoding. This is because GDELT displays a notable degree of temporal variability, wherein certain facts endure across multiple consecutive timestamps, while others are momentary and sparse. The

Model	Mapping on	Relation			ICEWS14				ICEWS05 − 15			
		$T_{\hat{r}}$	$S_{\hat{r}}$	$R_{\hat{r}}$	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TCompoundE	$S_{\hat{r}}$	✔	✔		**0.644**	**0.561**	**0.694**	**0.795**	**0.692**	**0.612**	**0.743**	**0.837**
V1	$T_{\hat{r}}$	✔			0.304	0.128	0.412	0.632	0.374	0.194	0.490	0.702
V2		✔	✔		0.576	0.476	0.639	0.755	0.623	0.521	0.690	0.809
V3		✔	✔	✔	0.571	0.468	0.636	0.757	0.620	0.517	0.686	0.807
V4	$S_{\hat{r}}$		✔		<u>0.602</u>	<u>0.518</u>	0.650	<u>0.762</u>	<u>0.676</u>	<u>0.595</u>	<u>0.727</u>	<u>0.824</u>
V5		✔	✔	✔	0.576	0.485	0.630	0.746	0.652	0.571	0.702	0.801
V6	$R_{\hat{r}}$			✔	0.601	0.515	<u>0.651</u>	0.761	0.634	0.555	0.680	0.781
V7		✔	✔	✔	0.582	0.491	0.638	0.752	0.638	0.550	0.690	0.804

Table 3: Results of different relation-specific operations on ICEWS14 and ICEWS05-15.

Model	Mapping on	Time			ICEWS14				ICEWS05 − 15			
		T_{τ}	S_{τ}	R_{τ}	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TCompoundE	$S_{\hat{r}}$	✔	✔		**0.644**	**0.561**	**0.694**	**0.795**	<u>0.692</u>	**0.612**	<u>0.743</u>	<u>0.837</u>
V8	$T_{\hat{r}}$	✔			0.578	0.478	0.640	0.756	0.621	0.519	0.685	0.807
V9			✔		0.577	0.478	0.639	0.754	0.620	0.517	0.685	0.807
V10				✔	0.559	0.460	0.619	0.737	0.584	0.482	0.646	0.771
V11		✔	✔	✔	0.564	0.462	0.628	0.747	0.606	0.505	0.670	0.794
V12	$S_{\hat{r}}$	✔			0.634	0.548	<u>0.687</u>	<u>0.791</u>	**0.693**	<u>0.607</u>	**0.750**	**0.846**
V13			✔		<u>0.638</u>	<u>0.558</u>	0.685	0.783	0.682	0.606	0.735	0.832
V14				✔	0.587	0.499	0.642	0.743	0.639	0.557	0.688	0.786
V15		✔	✔	✔	0.582	0.505	0.627	0.734	0.673	0.595	0.723	0.815

Table 4: Results of different time-specific operations on ICEWS14 and ICEWS05-15.

performance of our model demonstrates it can capture the dynamic evolution of relations. The results across all three datasets indicate the effectiveness of our combined approach in addressing the temporal knowledge graph completion problem.

6.2 Different Combinations

We investigate the impact of various combinations on ICEWS14 and ICEWS05-15. And we also compare the performance of TCompoundE with its variants. More comprehensive experimental results can be found in Appendix H. We classify these variants into two groups: the first group involves maintaining the time-specific operation unchanged while modifying the relation-specific operation, and the second group involves maintaining the relation-specific operation unchanged while modifying the time-specific operation. The results of the ICEWS14 and ICEWS05-15 variants from the aforementioned two groups are presented in Table 3 and Table 4. The best results are highlighted in bold font, while the second best are underlined.

In Table 3, we can observe that TCompoundE outperforms its variants across all metrics on ICEWS14 and ICEWS05-15. As TCompoundE utilizes translation and scaling operations as relation-specific operations, this enables TCompoundE to

model critical relation patterns. We can observe from Table 3 that TCompoundE has significantly outperformed its variants when employing a single operation for relation-specific operations, including translation (V1), scaling (V4) and rotation (V6). It confirms that utilizing a single operation as relation-specific has drawbacks in TKGs. We also observe that TCompoundE outperforms its variants across all geometric operations, including V3, V4 and V7. These variant models employ translation, scaling, and rotation as relation-specific operations. The distinction among these models lies in the usage of time-specific operations in different relation-specific operations, such as translation (V3), scaling (V5) and rotation (V7). This underscores that incorporating too many geometric operations as relation-specific operations is not optimal in TKGs. Conversely, V2 utilizes the same relation-specific operation as TCompoundE, while applying the time-specific operation to the relation-specific translation operation. This indicates that more features that remain constant over time can be learned from relation-specific translation operations.

Table 4 displays the results of the second group of TCompoundE variants. In Table 4, V8, V9, and V10 employ single translation, scaling and rotation as time-specific operations. V12, V13 and V14 are

(a) The initial state of e_s and e_o (b) e_s apply $T_{\hat{r}\tau}$ (c) e_s apply $T_{\hat{r}\tau}$ and $S_{\hat{r}\tau}$
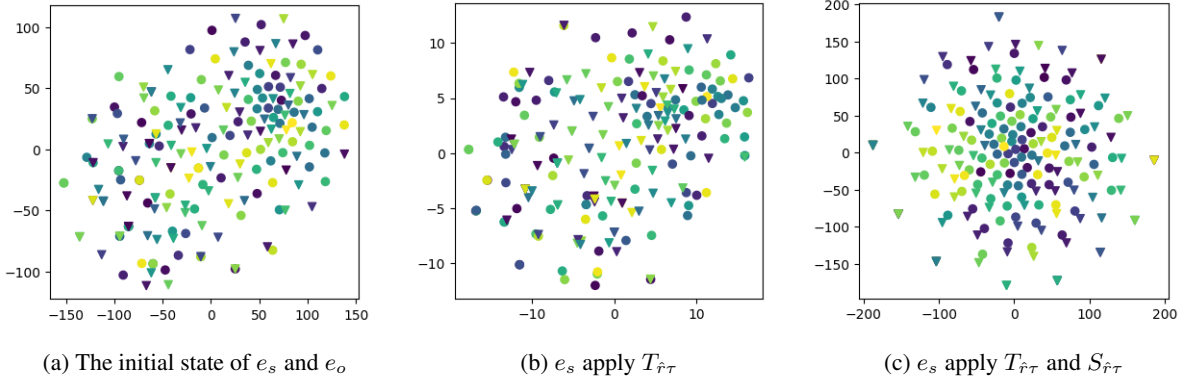
Figure 3: Visualisations of the learned entity embeddings on ICEWS14. The circle in the figure represents the head entity and the triangle represents the tail entity; Circles and triangles of the same color indicate that the head and tail entities come from the same quadruple.
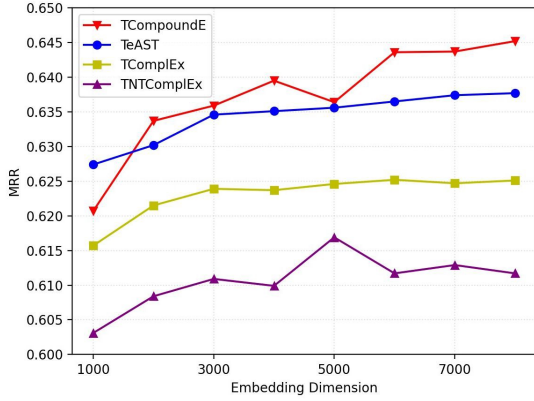

Figure 4: MRR scores on ICEWS14 dataset.

identical to V8, V9 and V10, respectively. From the results, we observe that TCompoundE remains highly competitive compared to its variant models when employing a single operation as a time-specific operation. V11 and V15 use translation, scaling and rotation as time-specific operations, applying them to relation-specific translation and relation-specific scaling operations, respectively. It is evident from Table 4 that TCompoundE demonstrates significant improvements compared to its variants when employing all geometric operations as time-specific operations. This confirms that incorporating too many operations is often suboptimal.

6.3 Effects of Operation

We randomly select 100 quadruples from ICEWS14 and employ t-SNE (Van der Maaten and Hinton, 2008) to visualize the head and tail entity embeddings of TCompoundE. The visualization

results are depicted in Fig. 3. We note that the initial state of the head entity and tail entity embeddings appears chaotic and irregular. However, when the head entity embeddings undergo $T_{\hat{r}\tau}$ and $S_{\hat{r}\tau}$ transformations, the distribution of the head and tail entity embeddings becomes more regular. The reason for this phenomenon, we think head entity is mapped to a space close to tail entity by relation-specific operation merging the temporal information. This is due to the design of the scoring function and illustrates the effectiveness of the composite operation. Specifically, the head and tail entities in the same quadruple exhibit a high degree of proximity. This confirms the effectiveness of employing both relation-specific and time-specific operations. In addition, we have a case study of the quadruple we use. Details can be found in the appendix I.

6.4 Effects of Embedding Dimension

We investigate the effect of embedding dimension on TCompoundE. In Fig. 4, we compare the MRR scores of TCompoundE and previous state-of-the-art (SOTA) embedding models on the ICEWS14 dataset across different dimension settings $d \in 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000$. According to the experimental results, TCompoundE is suboptimal in low dimensions, but it outperforms benchmarking methods when $d \geq 2000$.

7 Conclusion

This paper introduces a novel method, TCompoundE, designed to address the challenge of

knowledge graph completion in TKGs. TCompoundE applies a compound of translation and scaling as relation-specific and time-specific operations. Our experimental results demonstrate that TCompoundE effectively manages both relation and time information in TKGs. Furthermore, we provide mathematical evidence supporting TCompoundE's capability to handle various relational patterns. Additionally, we explore the effectiveness of different combinations of relation-specific and time-specific operations.

Limitations

Similar to many temporal knowledge graph embedding models, our proposed method, TCompoundE, faces limitations during the training phase as it cannot learn about invisible entities and time. Consequently, TCompoundE cannot directly perform temporal knowledge graph extrapolation tasks. Additionally, our model excels at high embedding dimensions, leading to its larger size compared to others.

Acknowledgements

References

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

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Conference on Empirical Methods in Natural Language Processing*.

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*, volume 26. Curran Associates, Inc.

Ricardo Campos, G. Dias, Alípio Mário Jorge, and Adam Jatowt. 2014. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47:1 – 41.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2020. Pairre: Knowledge graph embeddings via paired relation vectors. *arXiv preprint arXiv:2011.03798*.

Kai Chen, Ye Wang, Yitong Li, and Aiping Li. 2022. Rotateqvs: representing temporal information as rotations in quaternion vector space for temporal knowledge graph completion. *arXiv preprint arXiv:2203.07993*.

Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Conference on Empirical Methods in Natural Language Processing*.

Xiou Ge, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. 2022. Compounde: Knowledge graph embedding with translation, rotation and scaling compound operations. *arXiv preprint arXiv:2207.05324*.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3988–3995.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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)*, pages 687–696.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1807–1810.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *ArXiv*, abs/2004.04926.

Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. Temporal diversity in recommender systems. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217.

Jennifer Lautenschlager, Steve Shellman, and Michael Ward. 2015. Icews event aggregations.

Steven M LaValle. 2006. *Planning Algorithms*. Cambridge University Press.

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 Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Jiang Li, Xiangdong Su, and Guanglai Gao. 2023. 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)*, pages 15460–15474.

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 AAAI conference on artificial intelligence*, volume 29.

Feifei Liu, Mingtong Liu, Meiting Li, Yuwei Xin, Dongping Gao, Jun Wu, and Jiaan Zhu. 2023. Automatic knowledge extraction from chinese electronic medical records and rheumatoid arthritis knowledge graph construction. *Quantitative Imaging in Medicine and Surgery*, 13:3873 – 3890.

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

Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6471–6479.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Sanju Tiwari, Fatima N Al-Aswadi, and Devottam Gaurav. 2021. Recent trends in knowledge graphs: theory and practice. *Soft Computing*, 25:8337–8355.

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

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575.

Fuwei Zhang, Zhao Zhang, Xiang Ao, Fuzhen Zhuang, Yongjun Xu, and Qing He. 2022. Along the time: Timeline-traced embedding for temporal knowledge graph completion. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2529–2538.

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

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3065–3072.

Appendix

A Definition of Relation Patterns

Definition 1 *A relation $\hat{r}$ is symmetric, if $\forall s, o, \tau, (s, \hat{r}, o, \tau) \wedge (o, \hat{r}, s, \tau) \in \mathcal{G}$*

Definition 2 *A relation $\hat{r}$ is asymmetric, if $\forall s, o, \tau, (s, \hat{r}, o, \tau) \in \mathcal{G} \wedge (o, \hat{r}, s, \tau) \notin \mathcal{G}$*

Definition 3 *Relation $\hat{r}_1$ is the inverse of $\hat{r}_2$, if $\forall s, o, \tau, (s, \hat{r}_1, o, \tau) \wedge (o, \hat{r}_2, s, \tau) \in \mathcal{G}$*

Definition 4 *Relation $\hat{r}_1$ and $\hat{r}_2$ are evolving over time from timestamp τ_1 to timestamp τ_2, if $\forall s, o, (s, \hat{r}_1, o, \tau_1) \wedge (s, \hat{r}_2, o, \tau_2) \in \mathcal{G}$.*

The above describes various relational patterns from a mathematical point of view. We will illustrate various relational patterns next. For symmetric patterns, *(Canada, Consult, France)* and *(France, Consult, Canada)* show that *Consult* is symmetric. For asymmetric patterns, *is father of* is an asymmetric relation, because *(personA, is father of, personB)* and *(personB, is father of, personA)* can't both be true. *is father of* and *is son of* are inverse relations. Temporal evolution patterns detail can be found at Figure 1.

B Proof of Propositions 1

Let M denote the compound operation for head entity. Following ComplEx, we employ the standard dot product $< a, b > = a \circ b = \sum_k a_k b_k$. For $< a, b > = < c, d >$, there are special cases that the previous equation holds true when $a_i b_i == c_i d_i$, where $i \in [0, k]$. For symmetric pattern, we can get $\phi(s, \hat{r}, o, \tau) = \phi(s, \hat{r}, o, \tau)$ based on definition of symmetric pattern. According to score function of TCompoundE, we get:

$$< Me_s, e_o > = < Me_o, e_s > \Rightarrow \\ Me_s \circ e_o = Me_o \circ e_s \qquad (12)$$

If the matrix M is invertible,then we can get:

$$Me_s \circ e_o = Me_o \circ e_s \Rightarrow$$
$$e_s \circ e_o = M^{-1}Me_o \circ e_s \Rightarrow \qquad (13)$$
$$e_s \circ e_o = e_o \circ e_s$$

Therefore, TCompoundE can model symmetric pattern when matrix M is invertible.

C Proof of Propositions 2

By definition of asymmetric pattern, we can get $\phi(s, \hat{r}, o, \tau) \neq \phi(s, \hat{r}, o, \tau)$. By similar proof for Propositions 1, TCompoundE can model asymmetric pattern when matrix is not invertible.

D Proof of Propositions 3

Based on definition of inverse pattern, we have $\phi(s, \hat{r}_1, o, \tau) = \phi(o, \hat{r}_2, s, \tau)$. Hence, we get

$$M_1 e_s \circ e_o = M_2 e_o \circ e_s \qquad (14)$$

If the matrix M_1 or M_2 is invertible, the we can get:

$$e_s \circ e_o = M_1^{-1} M_2 e_o \circ e_s \Rightarrow$$
$$M_1^{-1} M_2 = I \quad or \quad M_2^{-1} M_1 = I \qquad (15)$$

Therefore, TCompoundE can model symmetric pattern when M_1 and M_2 are inverse matrices.

E Proof of Propositions 4

For temporal evolution pattern, we can get $\phi(s, \hat{r}_1, o, \tau_1) = \phi(s, \hat{r}_2, o, \tau_2)$ based on definiton of temporal evolution pattern. Then through the score function, we can get:

$$M_{\hat{r}_1 \tau_1} e_s \circ e_o = M_{\hat{r}_2 \tau_2} e_o \circ e_s \qquad (16)$$

where $M_{\hat{r}_1 \tau_1} = S_{\tau_1} T_{\tau_1} S_{\hat{r}_1} T_{\hat{r}_1}$ and $M_{\hat{r}_2 \tau_2} = S_{\tau_2} T_{\tau_2} S_{\hat{r}_2} T_{\hat{r}_2}$. Then we can get:

$$e_s \circ e_o = M_{\hat{r}_1 \tau_1}^{-1} M_{\hat{r}_2 \tau_2} \circ e_s \Rightarrow$$
$$M_{\hat{r}_1 \tau_1}^{-1} M_{\hat{r}_2 \tau_2} = I \quad or \quad M_{\hat{r}_2 \tau_2}^{-1} M_{\hat{r}_1 \tau_1} = I \qquad (17)$$

We can observe from the above formula: TCompoundE can model temporal evolution pattern when $M_{\hat{r}_1 \tau_1}$ and $M_{\hat{r}_2 \tau_2}$ are inverse matrices.

F Introduction of Variant Models

We introduce $R_{\hat{r}}$ and R_τ in the variant model to represent the relation-specific and time-specific rotation operation. To better explain the differences between variant models, we introduce a more general formula instead of Formula 7:

$$e_s^{\hat{r}\tau} = R_{\hat{r}\tau} \cdot S_{\hat{r}\tau} \cdot T_{\hat{r}\tau} \cdot e_s \qquad (18)$$

where $R_{\hat{r}\tau}$ represents relation-specific rotation operation that incorporates temporal information. Next we will look at the formula differences between TCompoundE and its variant models in obtaining the $R_{\hat{r}\tau}$, $S_{\hat{r}\tau}$ and $T_{\hat{r}\tau}$.

	$R_{\hat{r}\tau}$	$S_{\hat{r}\tau}$	$T_{\hat{r}\tau}$
TCompoundE	I	$S_\tau \cdot T_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$
V1	I	I	$S_\tau \cdot T_\tau \cdot T_{\hat{r}}$
V2	I	$S_{\hat{r}}$	$S_\tau \cdot T_\tau \cdot T_{\hat{r}}$
V3	$R_{\hat{r}}$	$S_{\hat{r}}$	$S_\tau \cdot T_\tau \cdot T_{\hat{r}}$
V4	I	$S_\tau \cdot T_\tau \cdot S_{\hat{r}}$	I
V5	$R_{\hat{r}}$	$S_\tau \cdot T_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$
V6	$S_\tau \cdot T_\tau \cdot R_{\hat{r}}$	I	I
V7	$S_\tau \cdot T_\tau \cdot R_{\hat{r}}$	$S_{\hat{r}}$	$T_{\hat{r}}$

Table 5: Variant models of the first group.

	$R_{\hat{r}\tau}$	$S_{\hat{r}\tau}$	$T_{\hat{r}\tau}$
TCompoundE	I	$S_\tau \cdot T_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$
V8	I	$S_{\hat{r}}$	$T_\tau \cdot T_{\hat{r}}$
V9	I	$S_{\hat{r}}$	$S_\tau \cdot T_{\hat{r}}$
V10	I	$S_{\hat{r}}$	$R_\tau \cdot T_{\hat{r}}$
V11	I	$S_{\hat{r}}$	$R_\tau \cdot S_\tau \cdot T_\tau \cdot T_{\hat{r}}$
V12	I	$T_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$
V13	I	$S_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$
V14	I	$R_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$
V15	I	$R_\tau \cdot S_\tau \cdot T_\tau \cdot S_{\hat{r}}$	$T_{\hat{r}}$

Table 6: Variant models of the second group.

In keeping with section 6.2, the variant model is also divided into two groups, where the variant model in the first group keeps the time-specific operations unchanged, and changes the relation-specific operations; The second group keeps the relation-specific operations of the same and change the time-specific operations.

Mapping on	Relation			ICEWS14				ICEWS05 − 15			
	$T_{\hat{r}}$	$S_{\hat{r}}$	$R_{\hat{r}}$	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
$S_{\hat{r}}$	✔	✔		**0.644**	**0.561**	**0.694**	**0.795**	**0.692**	**0.612**	**0.743**	**0.837**
$T_{\hat{r}}$	✔			0.304	0.128	0.412	0.632	0.374	0.194	0.490	0.702
	✔	✔		0.576	0.476	0.639	0.755	0.623	0.521	0.690	0.809
	✔		✔	0.547	0.442	0.612	0.740	0.606	0.522	0.655	0.760
	✔	✔	✔	0.571	0.468	0.636	0.757	0.620	0.517	0.686	0.807
$S_{\hat{r}}$		✔		0.602	0.518	0.650	0.762	0.676	0.595	0.727	0.824
		✔	✔	0.580	0.478	0.644	0.770	0.649	0.556	0.713	0.818
	✔	✔	✔	0.576	0.485	0.630	0.746	0.652	0.571	0.702	0.801
$R_{\hat{r}}$			✔	0.601	0.515	0.651	0.761	0.634	0.555	0.680	0.781
		✔	✔	0.587	0.491	0.644	0.763	0.636	0.546	0.687	0.804
	✔		✔	0.548	0.445	0.607	0.742	0.606	0.522	0.655	0.760
	✔	✔	✔	0.582	0.491	0.638	0.752	0.638	0.550	0.690	0.804

Table 7: Supplementary results of different relation-specific operations on ICEWS14 and ICEWS05-15

Mapping on	Time			ICEWS14				ICEWS05 − 15			
	T_{τ}	S_{τ}	R_{τ}	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
$S_{\hat{r}}$	✔	✔		**0.644**	**0.561**	**0.694**	**0.795**	0.692	**0.612**	0.743	0.837
$T_{\hat{r}}$	✔			0.578	0.478	0.640	0.756	0.621	0.519	0.685	0.807
		✔		0.577	0.478	0.639	0.754	0.620	0.517	0.685	0.807
			✔	0.559	0.460	0.619	0.737	0.584	0.482	0.646	0.771
	✔		✔	0.563	0.461	0.629	0.746	0.601	0.494	0.670	0.795
		✔	✔	0.565	0.466	0.623	0.763	0.607	0.505	0.672	0.793
	✔	✔	✔	0.564	0.462	0.628	0.747	0.606	0.505	0.670	0.794
$S_{\hat{r}}$	✔			0.634	0.548	0.687	0.791	**0.693**	0.607	**0.750**	**0.846**
		✔		0.638	0.558	0.685	0.783	0.682	0.606	0.735	0.832
			✔	0.587	0.499	0.642	0.743	0.639	0.557	0.688	0.786
	✔		✔	0.620	0.538	0.670	0.768	0.680	0.598	0.734	0.826
		✔	✔	0.577	0.498	0.621	0.722	0.671	0.592	0.720	0.814
	✔	✔	✔	0.582	0.505	0.627	0.734	0.673	0.595	0.723	0.815

Table 8: Supplementary results of different relation-specific operations on ICEWS14 and ICEWS05-15

From the first group of variant models introduced in Table 5, we can see that V1, V2 and V3 all apply time-specific operations to relation-specific translation operations; V4 and V5 apply time-specific operations to relation-specific scaling operations; V6 and V7 apply it to relationship-specific rotation operations.

A formula introduction for the second set of variant models can be viewed in Table 6. Where V8, V9, V10 and V11 all make time-specific operations specific to relation-specific translation operations; V12, V13, V14 and V15 apply it to relationship-specific scaling operations.

G Different Scoring Function

The semantic similarity scoring function is shown in function 8. The distance scoring function is defined as follows:

$$\phi(s, \hat{r}, o, \tau) = \|e_s^{\hat{r}\tau} - e_o\|_2 \qquad (19)$$

From the definition of the distance scoring func-

tion, we can see that in the distance scoring function, we use the L2 norm to find the Euclidean distance of $e_s^{\hat{r}\tau}$ and e_o.

	MRR	H@1	H@3	H@10
Distance	0.446	0.333	0.506	0.667
Similarity	**0.644**	**0.561**	**0.694**	**0.795**

Table 9: Distance and similarity scoring function result on ICEWS14.

We can observe that the semantic similarity scoring function is more advantageous in Table 9. In addition, the broadcast mechanism of the tensor can be better utilized by using the semantic similarity scoring function. It takes up less memory at runtime and has a shorter run time.

H More result of variant of TCompound

In the experimental analysis section, we examined the utilization of single operations as well as

variants incorporating all operations. This section presents additional experimental results, encompassing variants employing various combinations of two geometric operations, such as translation and rotation, and scaling and rotation. These combinations are applied to both relation-specific and time-specific operations. Our model consistently outperforms all aforementioned variants across all metrics on the ICEWS14 and ICEWS05-15 datasets, as illustrated in Tables 7 and 8. These findings confirm the soundness of TCompoundE's design.

I Case Study

We have selected some of the quadruples used in Section 6.3 experiments as examples in the case study. The experimental results are shown in the table 10, where e_s, $T_{\hat{r}\tau}e_s$, and $S_{\hat{r}\tau}T_{\hat{r}\tau}e_s$ respectively represent the three stages in Section 6.3.

	e_s	$T_{\hat{r}\tau}e_s$	$S_{\hat{r}\tau}T_{\hat{r}\tau}e_s$
C1	72	38	**1**
C2	6	4	**1**
C3	72	38	**1**
C4	7100	7072	**2**
C5	431	152	**3**
C6	87	17	**1**
C7	65	99	**1**
C8	300	9	**1**
C9	47	41	**1**
C10	216	105	**1**

Table 10: Case study on ICEWS14.

The table shows the ranking in each stage. We can observe from the table that the ranking of the sampled quadruples will have a greater improvement after the two stages than in the initial stage. The details of C1 - C10 are shown as follows:
(South Korea, Express intent to meet or negotiate, China, 2014-09-30)
(South Korea, Criticize or denounce, North Korea,2014-07-03)
(South Korea, Express intent to meet or negotiate, China, 2014-09-30)
(South Korea, Host a visit, Barack Obama, 2014-04-23)
(South Korea, Express intent to cooperate, North Korea, 2014-03-27)
(South Korea, Express intent to cooperate, North Korea, 2014-03-27)
(South Korea, Host a visit, North Korea, 2014-10-08)
(South Korea, Express intent to cooperate, China, 2014-07-20)
(South Korea, Make an appeal or request, China, 2014-06-25)
(South Korea, Consult, Head of Government (Egypt), 2014-11-23)