Block-Diagonal Orthogonal Relation and Matrix Entity for Knowledge Graph Embedding

Anonymous ACL submission

Abstract

The primary aim of Knowledge Graph Embeddings (KGE) is to learn low-dimensional representations of entities and relations for predicting missing facts. Although rotation-based methods like RotatE (Sun et al., 2019) and QuatE (Zhang et al., 2019) perform well in KGE, they face two challenges: limited model flexibility requiring proportional increases in relation size with entity dimension, and difficulties in generalizing the model for higher-dimensional rotations. To address these issues, we introduce OrthogonalE, a novel KGE model employing matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations. This approach not only enhances the generality and flexibility of KGE models but also captures several relation patterns that rotation-based methods can identify. Experimental results indicate that our new KGE model, OrthogonalE, offers generality and flexibility, captures several relation patterns, and significantly outperforms state-of-the-art KGE models while substantially reducing the number of relation parameters.

1 Introduction

The fundamental elements of knowledge graphs (KGs) are factual triples, each represented as (h, r, t), indicating a relationship r between head entity h and tail entity t. Notable examples include Freebase (Bollacker et al., 2008), Yago (Suchanek et al., 2007), and WordNet (Miller, 1995). KGs have practical applications in various fields such as question-answering (Hao et al., 2017), information retrieval (Xiong et al., 2017), recommender systems (Zhang et al., 2016), and natural language processing (Yang and Mitchell, 2019), garnering considerable interest in academic and commercial research.

Addressing the inherent incompleteness of KGs, link prediction has become a pivotal area of focus. Recent research (Bordes et al., 2013; Trouillon et al., 2016) has extensively leveraged Knowledge Graph Embedding (KGE) techniques, aiming to learn compact, low-dimensional representations of entities and relations. These approaches, marked by scalability and efficiency, have shown proficiency in modeling and deducing KG entities and relations from existing facts.

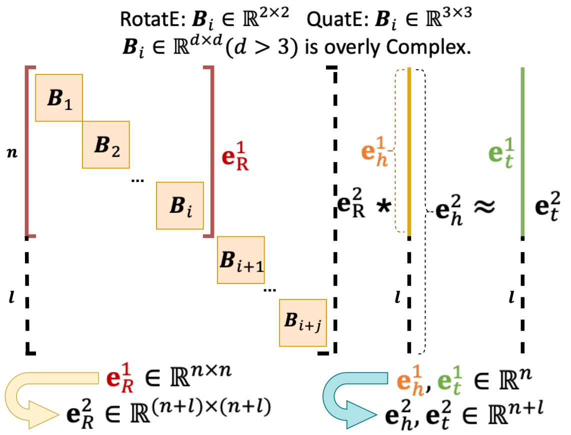


Figure 1: Fundamental operations ($\mathbf{e}_R^1 \cdot \mathbf{e}_h^1 \approx \mathbf{e}_t^1$) and inherent challenges of rotation-based KGE models. Rotation-based methods require increasing relation parameters for adequate entity representation and struggle with researching higher-dimensional rotation embeddings ($d > 3$) due to their complexity. OrthogonalE, depicted in Fig. 2, efficiently resolves these challenges.

Recently, rotation-based KGE methods have achieved notable success in the field. For instance, RotatE (Sun et al., 2019) conceptualizes relations as 2D rotations while QuatE (Zhang et al., 2019) employs 3D rotations to obtain a more expressive model than RotatE. Essentially, as illustrated in Fig. 1 , both operate by multiplying the relation matrix $\mathbf{e}_R^1 \in \mathbb{R}^{n\times n}$ composed of the block-diagonal Rotation matrix $\mathbf{B}_i \in \mathbb{R}^{d\times d}$ (RotatE: $\mathbb{R}^{2\times2}$, QuatE: $\mathbb{R}^{3\times3}$) with the head entity vector $\mathbf{e}_h^1 \in \mathbb{R}^n$.

However, these approaches face two primary issues, as illustrated in Fig. 1. First, the model's lack

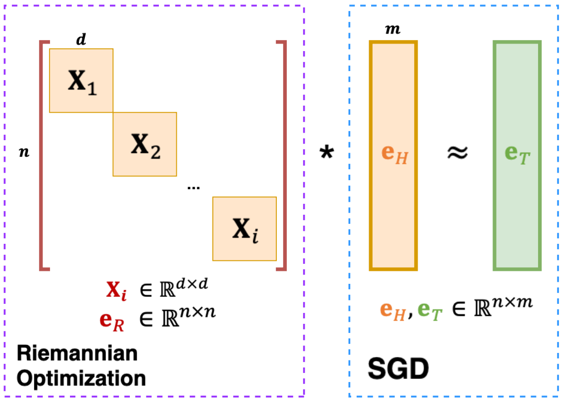


Figure 2: Diagram of the OrthogonalE approach. We employ matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations, thereby retaining the advantages of rotation-based method relation patterns while addressing its two main issues.

Notation	Explanation
$(h, r, t) \in \mathcal{E}$	Fact triples
$\mathcal{V}$	Entity sets
$\mathcal{R}$	Relation sets
$\mathbf{e}_v \in \mathbb{R}^n$	Entity vector rep
$\mathbf{e}_V \in \mathbb{R}^{n\times m}$	Entity matrix rep in OrthogonalE
$\mathbf{e}_R \in \mathbb{R}^{n\times n}$	Relation matrix rep
$\mathbf{B}_i \in \mathbb{R}^{d\times d}$	Block-diagonal rotation matrix
$\mathbf{X}_i \in \mathbb{R}^{d\times d}$	Block-diagonal orthogonal matrix
$n \in \mathbb{R}^1$	Row size of relation matrix rep
$m \in \mathbb{R}^1$	Column size of entity matrix rep
$d \in \mathbb{R}^1$	size of Block-diagonal matrix
$d^E(.,.)$	Euclidean distance
$b_v \in \mathbb{R}^1$	Entity bias
$\cdot$	Matrix multiplication
$s(h, r, t)$	Scoring function

Table 1: Notation summary. Within the table, $\mathbf{e}_v$ includes the head $\mathbf{e}_h$ and tail $\mathbf{e}_t$ entity vectors as used in traditional KGE methods, whereas $\mathbf{e}_V$ consists of the head $\mathbf{e}_H$ and tail $\mathbf{e}_T$ entity matrix representations in our OrthogonalE approach. Furthermore, 'rep' in the table denotes representation.

of flexibility necessitates increasing the overall relation matrix ($\mathbf{e}_R^1 \in \mathbb{R}^{n\times n} \rightarrow \mathbf{e}_R^2 \in \mathbb{R}^{(n+l)\times(n+l)}$) to meet entity dimension requirements ($\mathbf{e}_h^1 \in \mathbb{R}^n \rightarrow \mathbf{e}_h^2 \in \mathbb{R}^{n+l}$) for better represent entities. For example, when the entity vector changes ($\mathbf{e}_h^1 \in \mathbb{R}^{100} \rightarrow \mathbf{e}_h^2 \in \mathbb{R}^{1000}$) for better representation, the parameter increase is 900, but the corresponding change in the relation matrix ($\mathbf{e}_R^1 \in \mathbb{R}^{100\times 100} \rightarrow \mathbf{e}_R^2 \in \mathbb{R}^{1000\times 1000}$) results in a parameter increase of 990,000. This substantial increase leads to redundancy and inefficiency in representing relations.

Second, exploring high-dimensional rotational KGE models is challenging due to the significant computational demands and complexity of rotations in higher dimensions ($\mathbf{B}_i : \mathbb{R}^{2\times 2}, \mathbb{R}^{3\times 3} \rightarrow \mathbb{R}^{d\times d}, d > 3$), such as SO(4), SO(5), and SO(10). This restricts the development of more generalized and higher-dimensional rotation KGE approaches.

To overcome these two issues, we propose a highly general and flexible KGE model named OrthogonalE as shown in Fig. 2, and detailed notation details are shown in Table. 1. Firstly, by transforming entity vectors $\mathbf{e}_v \in \mathbb{R}^n$ into matrices $\mathbf{e}_V \in \mathbb{R}^{n\times m}$ for better represent entities, we control the entity dimension through variable m, avoiding unnecessary expansion of the relation size. Corresponding to the above example, we can maintain relation size ($\mathbf{e}_R \in \mathbb{R}^{100\times 100}$) and only modify entity matrix size ($\mathbf{e}_V^1 \in \mathbb{R}^{100\times 1} \rightarrow \mathbf{e}_V^2 \in \mathbb{R}^{100\times 10}, m : 1 \rightarrow 10$) to meet the requirements of entity representation. Secondly, leveraging the concept that rotation matrices are orthogonal, we replace rotation matrices $\mathbf{B}_i$ with orthogonal matrices $\mathbf{X}_i \in \mathbb{R}^{d\times d}$ of adaptable dimensions d, facilitating the exploration of higher-dimensional block-diagonal orthogonal matrix models. Lastly, for effective optimization, we employ Riemannian optimization for the relation matrix $\mathbf{e}_R \in \mathbb{R}^{n\times n}$ and Stochastic Gradient Descent (SGD) for the entity matrix $\mathbf{e}_V \in \mathbb{R}^{n\times m}$.

We evaluate the new model on two KGE datasets including WN18RR (Dettmers et al., 2018), FB15K-237 (Toutanova and Chen, 2015). Experimental results indicate that our new KGE model, OrthogonalE, offers generality and flexibility, captures several relation patterns, and significantly outperforms state-of-the-art KGE models while substantially reducing the number of relation parameters.

2 Related Work

Knowledge Graph Embedding Translation-based approaches are prominent in KGE, notably TransE (Bordes et al., 2013), which interprets relations as vector translations. TransH (Wang et al., 2014), TransR (Lin et al., 2015), and TransD (Ji et al., 2015) represent extensions of the translation-based method, building upon the foundational approach of TransE. ComplEx(Trouillon et al., 2016) advances this by embedding KGs in a complex space and using the Hermitian product for modeling antisymmetric patterns. Inspired by ComplEx, RotatE (Sun et al., 2019) then innovated by treating relations as rotations in a complex vector space, capable of capturing varied relation patterns like *Symmetry*, *Antisymmetry*, *Inversion*, and *Commutative Composition*. Following this, QuatE (Zhang et al., 2019) employed quaternion operations (3D rotations) for even better expressiveness than RotatE. DensE (Lu et al., 2022) employed various techniques for 3D rotation implementation and proposed that 3D rotation could handle the relation pattern of *non-commutative composition*. HopfE (Bastos et al., 2021) seeks to employ SO(4) rather than SO(3) for KG representation, which is directly connected to the generality issue discussed in our research. We are also keen on investigating rotations in higher dimensions. Nonetheless, progressing to SO(5) or even SO(10) poses substantial difficulties.

In conclusion, considering the two major disadvantages of rotation-based methods mentioned in the Introduction 1, we need to refine our model to make it more general and flexible.

Optimization on the orthogonal manifold In optimization on the orthogonal manifold, transitioning from X^t to X^{t+1} while remaining on the manifold necessitates a method known as retraction (Absil and Malick, 2012). Prior research has effectively adapted several standard Euclidean function minimization algorithms to Riemannian manifolds. Notable examples include gradient descent ((Absil et al., 2008); (Zhang and Sra, 2016)), second-order quasi-Newton methods ((Absil et al., 2007); (Qi et al., 2010)), and stochastic approaches (Bonnabel, 2013), crucial in deep neural network training.

Meanwhile, we often use Riemannian optimization for the orthogonal manifold, which has also progressed in deep learning, especially in CNNs and RNNs. (Cho and Lee, 2017) innovatively substituted CNN's Batch Normalization layers with Riemannian optimization on the Grassmann manifold for parameter normalization. Additionally, significant strides in stabilizing RNN training have been made by (Vorontsov et al., 2017), (Wisdom et al., 2016), (Lezcano-Casado and Martınez-Rubio, 2019), and (Helfrich et al., 2018), through the application of Riemannian optimization to unitary matrices.

As this paper primarily focuses on KGE, we do not delve deeply into Riemannian optimization.

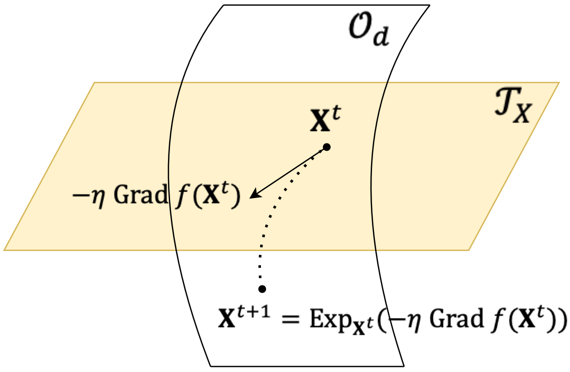


Figure 3: Abstract representation of Riemannian gradient descent iteration on orthogonal manifold

Instead, we utilize the retraction with exponential map for iterative optimization, sourced from Geoopt (Kochurov et al., 2020).

3 Problem Formulation and Background

We present the KGE problem and describe Optimization on the orthogonal manifold before our approach part.

3.1 Knowledge Graph Embedding

In a KG consisting of fact triples $(h, r, t) \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, with $\mathcal{V}$ and $\mathcal{R}$ denoting entity and relation sets, the objective of KGE is to map entities $v \in \mathcal{V}$ to $k_{\mathcal{V}}$-dimensional embeddings $\mathbf{e}_v$, and relations $r \in \mathcal{R}$ to $k_{\mathcal{R}}$-dimensional embeddings $\mathbf{e}_r$.

A scoring function $s : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \to \mathbb{R}$ evaluates the difference between transformed and target entities, quantified as a Euclidean distance:

$$d^E(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$$

3.2 Optimization on the orthogonal manifold

In optimization on the orthogonal manifold, the core problem is formulated as:

$$\min_{X \in \mathcal{O}_d} f(X), \tag{1}$$

Here, f is a differentiable function mapping elements of $\mathbb{R}^{d\times d}$ to $\mathbb{R}$, and the *orthogonal manifold* $\mathcal{O}_d$ is defined as $\mathcal{O}_d \triangleq \{X \in \mathbb{R}^{d\times d} \mid XX^\top = I_d\}$. Moreover, the tangent space at X, denoted by $\mathcal{T}_X$, is the set $\mathcal{T}_X = \{\xi \in \mathbb{R}^{d\times d} \mid \xi X^\top + X\xi^\top = 0\}$.

To address the problem 1 more efficiently, recent studies suggest optimization of the orthogonal manifold with retractions as an effective approach (Ablin and Peyré, 2022). In this work, we primarily employ the retraction with exponential map for iterative optimization, as illustrated in Fig. 3. The key iteration formula for this method is:

$$X^{t+1} = \mathrm{Exp}_{X^t}(-\eta \mathrm{Grad} f(X^t)), \tag{2}$$

Where t indexes the iteration steps, $\mathrm{Exp}_{X^t}(\xi)$ denotes the exponential map, and η represents the learning rate. $\mathrm{Grad} f(\cdot)$ is the Riemannian gradient. Subsequent sections will delve into the computation of $\mathrm{Exp}_{X^t}(\xi)$ and $\mathrm{Grad} f(\cdot)$.

The exponential map allows movement in a specified direction on the manifold. Starting from X with initial velocity ξ, the exponential map for the orthogonal matrices manifold is represented by (Massart and Abrol, 2022):

$$\mathrm{Exp}_X(\xi) = X \mathrm{expm}(X^\top \xi), \forall \xi \in \mathcal{T}_X,$$

where $\mathrm{expm}(\cdot)$ denotes the matrix exponential.

On the orthogonal manifold, the Riemannian gradient $\mathrm{Grad} f(\cdot)$ is calculated as (Absil et al., 2008):

$$\mathrm{Grad} f(X) = P_{\mathcal{T}_X}(\bigtriangledown f(X)),$$

Where $\bigtriangledown f(X)$ is Euclidean gradient of $f(X)$, and the calculation formula for $P_{\mathcal{T}_X}(\cdot)$ is:

$$P_{\mathcal{T}_X}(Y) = X(\frac{X^\top Y - Y^\top X}{2}), Y \in \mathbb{R}^{d \times d}$$

4 Approach

Our approach is developed to acquire both a flexible and general KGE model and ensure that this model can concurrently represent several relation patterns. This is achieved by employing matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations. Figure 2 illustrates the OrthogonalE approach, and Table 1 provides the details of the notations used.

4.1 Orthogonal Matrices for Relations

To address the challenge of exploring high-dimensional rotational KGE models mentioned in the introduction, we exploit the orthogonality of rotation matrices, substituting rotation matrices ($\mathbf{B}_i \in \mathbb{R}^{d \times d}$) with orthogonal matrices ($\mathbf{X}_i \in \mathbb{R}^{d \times d}$) of corresponding dimensions d. Consequently, our relation embedding ($\mathbf{e}_R \in \mathbb{R}^{n \times n}$) are composed of n/d block-diagonal orthogonal matrices $\mathbf{X}_i$ as illustrated in Fig. 2:

$$\mathbf{e}_R = diag(\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_{n/d}) \tag{3}$$

e	Model	Number of Parameters		
		Normal	$m=1$ 5.2.2	Fixed $\mathbf{e}_V$ 5.2.4
$\mathbf{e}_R$	RotatE	$\frac{n}{2}$	$\frac{n}{2}$	$(\frac{n}{2}) * m$
	QuatE	n	n	$n * m$
	OrthogonalE(d)	$\frac{(d-1)n}{2}$	$\frac{(d-1)n}{2}$	$\frac{(d-1)n}{2}$
$\mathbf{e}_V$	RotatE	n	n	$n * m$
	QuatE	n	n	$n * m$
	OrthogonalE(d)	$n * m$	n	$n * m$

Table 2: The parameter calculations for the KGE models. For all models, the relation matrix size is n. The block-diagonal matrix size is 2 for RotatE, 3 for QuatE, and d for OrthogonalE($d \times d$), with an entity matrix column size of m for OrthogonalE. In the table, "Normal" represents the standard parameter calculation, "$m = 1$" constrains the column size of the entity matrix to 1 to explore the impact of block-diagonal orthogonal matrices on the model, as analyzed in section 5.2.2. "Fixed $\mathbf{e}_V$" ensures that the entity dimensions are consistent across all models to demonstrate the parameter savings in the relation matrix when using the entity matrix in OrthogonalE, as discussed in section 5.2.4.

Where the number of relation parameters is $\frac{d(d-1)}{2} * \frac{n}{d} = \frac{(d-1)n}{2}$, which shown in Table. 2. And this aspect allows OrthogonalE to gain generality, adapting to datasets with diverse complexities by modifying the block-diagonal matrices' dimension d. Additionally, the employed relation structure facilitates the model's capability to concurrently capture *Symmetry*, *Antisymmetry*, *Inversion*, and *Non-commutative Composition* relation patterns, as substantiated in Appendix A.3, and detailed introduction of relation patterns refer to Appendix A.5.

4.2 Matrices Representation for Entities

Inspired by (Miyato et al., 2022), transforms vector embeddings into matrix embeddings to improve embedding effectiveness. In our work, to enhance OrthogonalE's flexibility, we aim to regulate entity dimension using variable m and transform entity vectors $\mathbf{e}_v \in \mathbb{R}^n$ into matrices $\mathbf{e}_V \in \mathbb{R}^{n \times m}$ as shown in Fig. 2, thus preventing unnecessary expansion of the relation size. This part allows OrthogonalE to acquire flexibility, adapting to diverse datasets with varying relation and entity parameters, rather than indiscriminately increase both. And the number of entity parameters is $n * m$.

4.3 Scoring function and Loss

We utilize the Euclidean distance between the transformed head entity $\mathbf{e}_R \cdot \mathbf{e}_H$ and the tail entity $\mathbf{e}_T$

as the scoring function:

$$s(h,r,t) = -d^E(\mathbf{e}_R \cdot \mathbf{e}_H, \mathbf{e}_T) + b_h + b_t \quad (4)$$

Here, $b_v(v \in \mathcal{V})$ denotes the entity bias, incorporated as a margin in the scoring function, following methodologies from (Tifrea et al., 2018; Balazevic et al., 2019). Furthermore, we opt for uniform selection of negative samples for a given triple (h,r,t) by altering the tail entity, rather than employing alternative negative sampling techniques. The loss function defined as follows:

$$L = \sum_{t'} \log\left(1 + \exp\left(y_{t'} \cdot s\left(h, r, t'\right)\right)\right) \quad (5)$$

$$y_{t'} = \begin{cases} -1, & \text{if } t' = t \\ 1, & \text{otherwise} \end{cases}$$

4.4 Optimization

Traditional KGE models train and optimize relations and entities jointly. In contrast, our study aims to achieve more effective optimization of the block-diagonal orthogonal matrices of relation embeddings $\mathbf{X}_i \in \mathbb{R}^{d \times d}$ by separately optimizing relations and entities, utilizing Riemannian optimization for the relation matrix $\mathbf{e}_R \in \mathbb{R}^{n \times n}$ and SGD for the entity matrix $\mathbf{e}_V \in \mathbb{R}^{n \times m}$.

Initially, when optimizing relations, all entity parameters are fixed, rendering the entity embeddings analogous to the function $f(\cdot)$ in the problem 1. Notably, each block-diagonal orthogonal matrix $\mathbf{X}_i$ within the relation embedding $\mathbf{e}_R$ optimized by individual Riemannian optimization using RiemannianAdam (Kochurov et al., 2020), which is a Riemannian version (equation 2) of the popular Adam optimizer (Kingma and Ba, 2014). These are then concatenated in a block-diagonal way according to equation 3 to complete the process. After optimizing the relation parameters $\mathbf{e}_R \in \mathbb{R}^{n \times n}$, they are held constant while the entity parameters $\mathbf{e}_V \in \mathbb{R}^{n \times m}$ are optimized using Stochastic Gradient Descent (SGD), specifically employing the Adagrad optimizer (Duchi et al., 2011).

5 Experiment

We expect that our proposed OrthogonalE model, employing matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations, will outperform baseline models. Also, we anticipate that OrthogonalE is a general and flexible KGE model and can represent several relation patterns simultaneously. Our goal is to validate these through empirical testing.

5.1 Experiment Setup

Dataset. We evaluate our proposed method on two KG datasets, including WN18RR (Dettmers et al., 2018) (license: Apache 2.0), FB15K-237 (Toutanova and Chen, 2015) (license: CC-BY-4.0). The details of these datasets are shown in Table 4. More detail is given in A.1.

Evaluation metrics. To predict the tail entity from a given head entity and relation, we rank the correct tail entity among all possible entities using two established ranking metrics. The first is the mean reciprocal rank (MRR), the average inverse ranking of the correct entities, calculated as $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\text{Rank}_i}$. Second is Hits@K for $K \in \{1, 3, 10\}$, the frequency of correct entities ranking within the top K positions.

Baselines. We compare our new model with several classic methods, including TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), and ConvE (Dettmers et al., 2018). Additionally, we include rotation-based KGE methods such as RotatE (Sun et al., 2019), QuatE (Zhang et al., 2019), HopfE (Bastos et al., 2021), and DensE (Lu et al., 2022) as baselines. In addition to these methods and our OrthogonalE$(d \times d)$, we introduce comparative models Gram-Schmidt$(d \times d)$ utilizing the Gram-Schmidt process for generating orthogonal matrices and SGD for joint relation-entity training. OrthogonalE further differentiates by employing orthogonal matrices of varying sizes to discuss performance nuances.

Implementation The key hyperparameters of our implementation include the learning rate for RiemannianAdam (Kochurov et al., 2020) and Adagrad (Duchi et al., 2011), negative sample size, and batch size. To determine the optimal hyperparameters, we performed a grid search using the validation data. More detail refers to A.1.

5.2 Results

We first analyzed the overall accuracy for all baseline models and OrthogonalE, then separately examined the impacts of block-diagonal Orthogonal matrices, Riemannian Optimization for relations, and entity matrices on the model from various experimental results. Finally, we utilize several relation histograms to verify our model can capture these relation patterns.

Model	WN18RR MRR	WN18RR H@1	WN18RR H@3	WN18RR H@10	FB15K-237 MRR	FB15K-237 H@1	FB15K-237 H@3	FB15K-237 H@10
TransE ♢	.226	-	-	.501	.294	-	-	.465
DistMult ♢	.430	.390	.440	.490	.241	.155	.263	.419
ComplEx ♢	.440	.410	.460	.510	.247	.158	.275	.428
ConvE ♢	.430	.400	.440	.520	.325	.237	.356	.501
RotatE ♢	.470	.422	.488	.565	.297	.205	.328	.480
QuatE ♢	.481	.436	.500	.564	.311	.221	.342	.495
HopfE (Bastos et al., 2021)	.472	.413	.500	**.586**	**.343**	**.247**	**.379**	**.534**
DensE (Lu et al., 2022)	.486	-	-	.572	.306	-	-	.481
Gram-Schmidt(2×2)	.475	.434	.489	.556	.317	.226	.344	.502
Gram-Schmidt(3×3)	.487	.445	.500	.568	.322	.232	.350	.504
OrthogonalE(2×2)	.490	.445	.503	.573	.330	.239	.368	.516
OrthogonalE(3×3)	.493	**.450**	**.508**	.580	.331	.240	.359	.513
OrthogonalE(4×4)	.493	.446	.506	.578	.332	.240	.363	.517
OrthogonalE(10×10)	**.494**	.446	**.508**	.573	.334	.242	.367	.518

Table 3: Link prediction accuracy results of two datasets, **Bold** indicates the best score, and underline represents the second-best score. For a fair comparison, we standardized m at 1 for Gram-Schmidt and all OrthogonalE sizes. The entity dimension for WN18RR was set at approximately 500 (for example, 501 for 3×3 blocks to ensure experimental feasibility) and around 1000 for FB15K-237. [♢]: The results are sourced from (Zhang et al., 2019). For a fair comparison, the results of RotatE, QuatE, HopfE, and DensE are reported without self-adversarial negative sampling, type constraints, semantics, or reciprocals. More baseline results are shown in Appendix A.6.

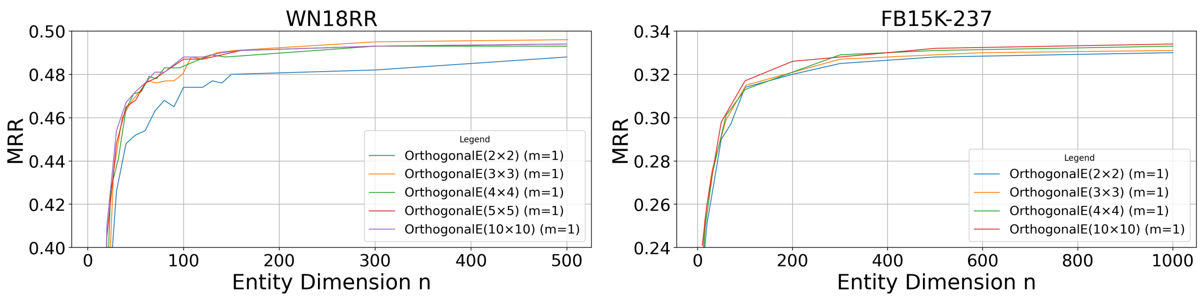


Figure 4: MRR accuracy comparison of OrthogonalE models with different block-diagonal orthogonal matrices across varying entity dimensions ($n * 1$, we set $m = 1$) on WN18RR and FB15K-237.

Dataset	Entities	Relations	Train	Validation	Test
WN18RR	40,943	11	86,835	3,034	3,134
FB15K-237	14,541	237	272,115	17,535	20,466

Table 4: Details of the two datasets.

5.2.1 Overall Accuracy

Table 3 presents link prediction accuracies for the WN18RR and FB15K-237 datasets. The OrthogonalE model demonstrates superior performance in the WN18RR dataset and achieves results on the FB15K-237 dataset that are only marginally lower than those of HopfE (Bastos et al., 2021), outperforming all other compared models, highlighting its superior representational ability by employing matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations. Moreover, the OrthogonalE model with 2×2 and 3×3 configurations yields significantly better performance than the corresponding sizes of the Gram-Schmidt method, and notably exceeds RotatE and QuatE, respectively, showcasing the enhanced efficacy of the KGE model. Finally, since the WN18RR and FB15K-237 datasets are relatively small, the performance differences among OrthogonalE models with (2×2), (3×3), (4×4), and (10×10) are not significant when using sufficient dimensions (WN18RR: 500, FB15K-237: 1000). We will discuss the performance at different dimensions in detail in section 5.2.2.

5.2.2 Block-diagonal Orthogonal matrices

Fig. 4 shows MRR accuracy comparison of OrthogonalE models with different block-diagonal orthogonal matrices in varying entity dimensions ($n * 1$,

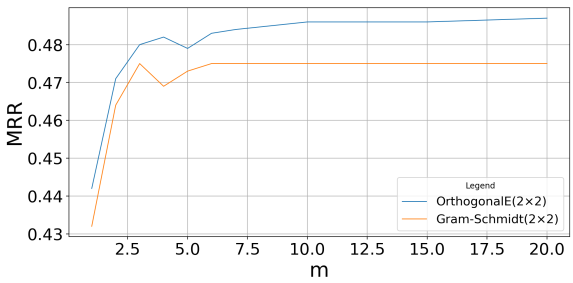


Figure 5: MRR accuracy comparison of OrthogonalE(2×2) and Gram-Schmidt(2×2) models across varying entity dimensions (m) with fixed relation matrix (40×40) on WN18RR.

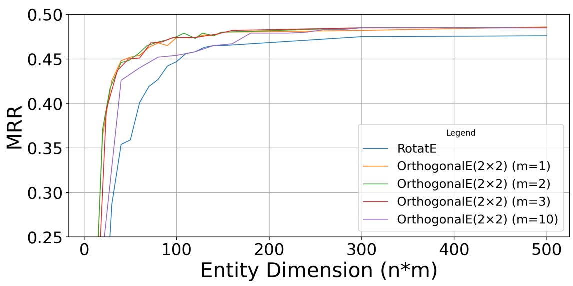


Figure 6: MRR accuracy comparison of RotatE and OrthogonalE(2×2) models across varying entity dimensions ($n*m$) on WN18RR.

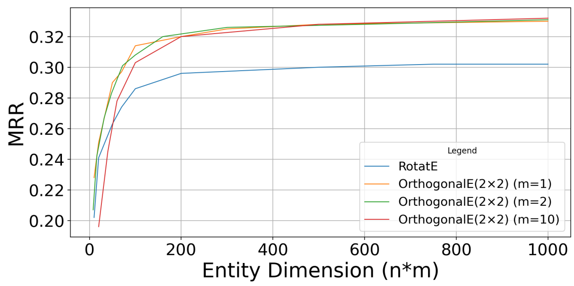


Figure 7: MRR accuracy comparison of RotatE and OrthogonalE(2×2) models across varying entity dimensions ($n*m$) on FB15K-237.

we set $m = 1$) on WN18RR and FB15K-237.

An initial dataset analysis reveals WN18RR has 40,943 entities with just 11 relations (about 3,722 entities per relation), while FB15K-237 includes 14,541 entities and 237 relations (around 61 entities per relation). This implies that WN18RR requires a more sophisticated representation capability compared to FB15K-237.

Our results (Fig. 4) confirm our dataset analysis. For WN18RR, the performance is similar for block sizes from 3×3 to 10×10, all outperforming 2×2 blocks, showcasing 2×2 blocks are not enough for its relation representation. However, for FB15K-237, performance is stable across all block sizes, indicating 2×2 blocks are enough for its relations representation. These results show WN18RR requires more complex blocks for adequate representation, and illustrate that the OrthogonalE model is general, which can adapt to datasets of various complexities by adjusting the dimension d of the block-diagonal matrices.

5.2.3 Riemannian Optimization for relations

Fig. 5 compares MRR accuracies of OrthogonalE (2×2) and Gram-Schmidt (2×2) across entity dimensions (m) with a constant relation matrix (40×40) on WN18RR, assessing the efficacy of orthogonal optimization beyond the Gram-Schmidt method for block-diagonal orthogonal matrices. The result demonstrates that OrthogonalE's Riemannian optimization significantly exceeds Gram-Schmidt, underscoring its necessity.

5.2.4 Entity matrix

In OrthogonalE, we maintained a constant entity dimension ($n * m$) while varying m to assess the impact of entity shape. Fig. 6 compares the MRR accuracies of RotatE with OrthogonalE (2×2) over different fixed entity dimensions $n*m$ in WN18RR. OrthogonalE models with $m = 1$, 2, or 3 perform similarly and better than $m = 10$, and all significantly outperform RotatE across dimensions. Notably, their relation parameter is $1/m$ of RotatE's, which is shown in Table. 2. These results demonstrate OrthogonalE's efficacy in saving relation parameters while outperforming RotatE, highlighting our model's flexibility in controlling entity dimension through variable m without unnecessarily increasing relation size.

Besides the comparison of RotatE and OrthogonalE(2×2), Fig. 7 shows comparison of RotatE and OrthogonalE (2×2) in FB15K-237. The experimental results, consistent with those discussed in the previous paragraph. More details refer to Appendix A.2.

5.2.5 Relation Pattern

Following the proof of relation patterns in Appendix A.3, Fig. 8 shows histograms of relation embeddings for different relation patterns. We provide several relation patterns examples and discussion of *non-commutative composition* in Appendix A.4

Symmetry and Antisymmetry In OrthogonalE, the *symmetry* relation pattern is encoded when the $\mathbf{e}_R$ embedding satisfies $\mathbf{e}_R \cdot \mathbf{e}_R = \mathrm{I}$, in accordance

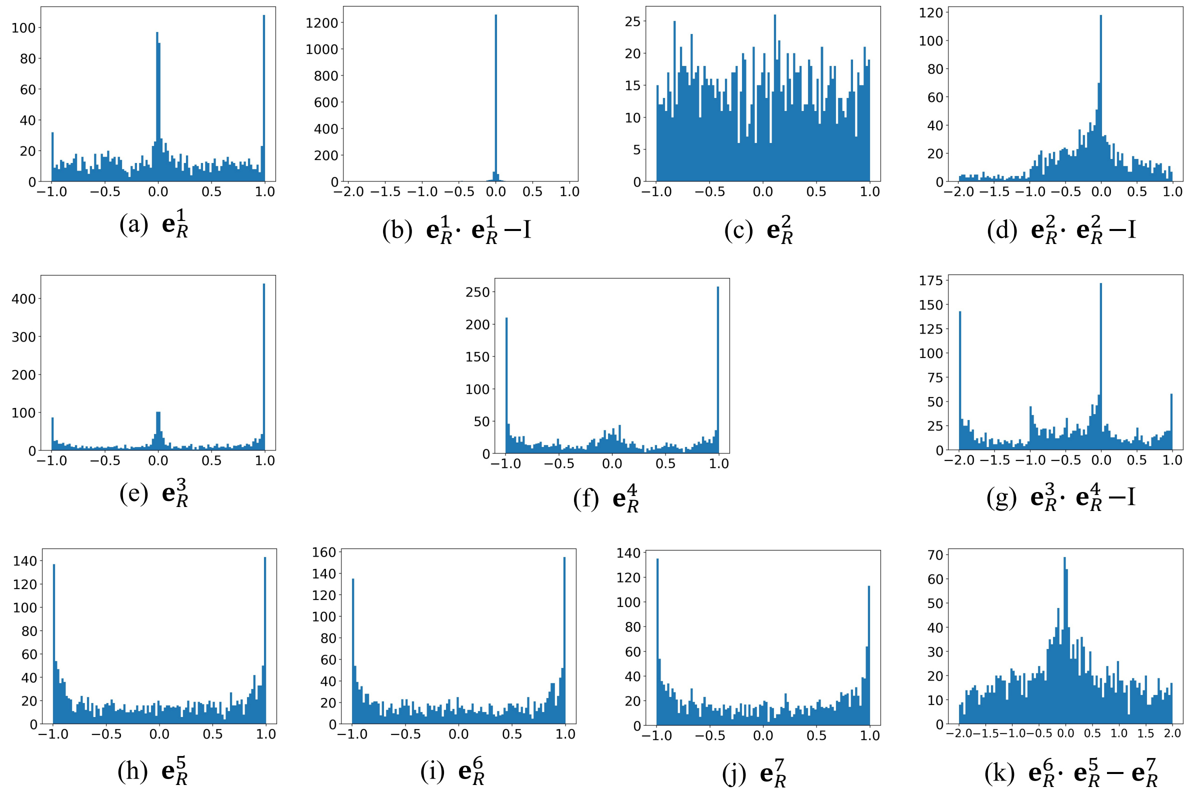


Figure 8: Histograms of relation embeddings for different relation patterns, where $\mathbf{e}_R^1$ represents `_similar_to`, $\mathbf{e}_R^2$ represents `_member_of_domain_region`, $\mathbf{e}_R^3$ represents `/film/film/genre`, $\mathbf{e}_R^4$ represents `/media_common/netflix_genre/titles`, $\mathbf{e}_R^5$ represents `/location/administrative_division/country`, $\mathbf{e}_R^6$ represents `/location/hud_county_place/place`, and $\mathbf{e}_R^7$ represents `/base/aareas/schema/administrative_area/capital`. From the WN18RR dataset, we select $\mathbf{e}_R^1$ and $\mathbf{e}_R^2$ to represent *Symmetry* and *Antisymmetry*, respectively, and obtain their relation embeddings using the OrthogonalE(3×3) model with n=500 and m=1. Similarly, from the FB15K-237 dataset, we select $\mathbf{e}_R^3$, $\mathbf{e}_R^4$, and $\mathbf{e}_R^5$, $\mathbf{e}_R^6$, $\mathbf{e}_R^7$ as representations for *Inversion* and *Composition*, respectively, and acquire their relation embeddings under the OrthogonalE(2×2) model with n=1000 and m=1.

with Equation 6. Figs. 8(a) and (b) illustrate the embeddings of $\mathbf{e}_R^1$ and $\mathbf{e}_R^1 \cdot \mathbf{e}_R^1 - \mathrm{I}$, respectively. From Fig. 8(b), we observe that nearly all values are concentrated around 0, thereby indicating that OrthogonalE's relations exhibit *symmetry* properties. Correspondingly, the multitude of nonzero values in Fig. 8(d) indicates that OrthogonalE's relations also can represent *antisymmetry* properties.

Inversion The *inversion* relation pattern is encoded when the $\mathbf{e}_R^3$ and $\mathbf{e}_R^4$ satisfies $\mathbf{e}_R^3 \cdot \mathbf{e}_R^4 = \mathrm{I}$, according to Equation 8. Even though $\mathbf{e}_R^3$ and $\mathbf{e}_R^4$ are responsible for additional relation patterns, which results in a cluster of values around -2 in Fig 8 (g), the majority of values still converge towards or equal 0. This suggests that OrthogonalE's relations have the *inversion* property.

Composition The *composition* relation pattern is encoded when the $\mathbf{e}_R^5$, $\mathbf{e}_R^6$, and $\mathbf{e}_R^7$ embedding satisfy $\mathbf{e}_R^6 \cdot \mathbf{e}_R^5 = \mathbf{e}_R^7$, in accordance with Equation 9. The majority of data in Fig. 8 (k) converge towards or are equal to 0, indicating that OrthogonalE's relations can represent the *composition* relation pattern.

6 Conclusion

In this study, we propose the OrthogonalE model to acquire a flexible and general KGE model with employing matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations. Experimental results indicate that our new KGE model offers generality and flexibility, captures several relation patterns, and outperforms SoTA rotation-based KGE models while substantially reducing the number of relation parameters.

Limitations

Even though the block-diagonal orthogonal relation with Riemannian optimization makes KGE models more general and improves their performance, the computation of exponential retraction in the orthogonal manifold for Riemannian optimization is costly. In practical model training, with the same entity dimension, our OrthogonalE (2×2) training time is 4 times longer than that of RotatE. In future research directions, we will continue to explore this limitation, such as by employing the landing algorithm (Ablin and Peyré, 2022) for retraction on orthogonal manifolds to reduce computational complexity.

Ethics Statement

This study complies with the ACL Ethics Policy.

Acknowledgements

References

Pierre Ablin and Gabriel Peyré. 2022. Fast and accurate optimization on the orthogonal manifold without retraction. In *International Conference on Artificial Intelligence and Statistics*, pages 5636–5657. PMLR.

P-A Absil, Christopher G Baker, and Kyle A Gallivan. 2007. Trust-region methods on riemannian manifolds. *Foundations of Computational Mathematics*, 7:303–330.

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. 2008. *Optimization algorithms on matrix manifolds*. Princeton University Press.

P-A Absil and Jérôme Malick. 2012. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*, 32.

Anson Bastos, Kuldeep Singh, Abhishek Nadgeri, Saeedeh Shekarpour, Isaiah Onando Mulang, and Johannes Hoffart. 2021. Hopfe: Knowledge graph representation learning using inverse hopf fibrations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 89–99.

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

Silvere Bonnabel. 2013. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.

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

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2020. Hitter: Hierarchical transformers for knowledge graph embeddings. *arXiv preprint arXiv:2008.12813*.

Minhyung Cho and Jaehyung Lee. 2017. Riemannian approach to batch normalization. *Advances in Neural Information Processing Systems*, 30.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. *Proceedings of the AAAI conference on artificial intelligence*, 32(1).

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

Chi Han, Qizheng He, Charles Yu, Xinya Du, Hanghang Tong, and Heng Ji. 2023. Logical entity representation in knowledge-graphs for differentiable rule learning. *arXiv preprint arXiv:2305.12738*.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231.

Jiabang He, Liu Jia, Lei Wang, Xiyao Li, and Xing Xu. 2023. Mocosa: Momentum contrast for knowledge graph completion with structure-augmented pre-trained language models. *arXiv preprint arXiv:2308.08204*.

Kyle Helfrich, Devin Willmott, and Qiang Ye. 2018. Orthogonal recurrent neural networks with scaled cayley transform. In *International Conference on Machine Learning*, pages 1969–1978. PMLR.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. 2020. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*.

Mario Lezcano-Casado and David Martınez-Rubio. 2019. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning*, pages 3794–3803. PMLR.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the AAAI conference on artificial intelligence*, 29(1).

Haonan Lu, Hailin Hu, and Xiaodong Lin. 2022. Dense: An enhanced non-commutative representation for knowledge graph embedding with adaptive semantic hierarchy. *Neurocomputing*, 476:115–125.

Estelle Massart and Vinayak Abrol. 2022. Coordinate descent on the orthogonal group for recurrent neural network training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7744–7751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Takeru Miyato, Masanori Koyama, and Kenji Fukumizu. 2022. Unsupervised learning of equivariant structure from sequences. *Advances in Neural Information Processing Systems*, 35:768–781.

Chunhong Qi, Kyle A Gallivan, and P-A Absil. 2010. Riemannian bfgs algorithm with applications. In *Recent Advances in Optimization and its Applications in Engineering: The 14th Belgian-French-German Conference on Optimization*, pages 183–192. Springer.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

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*.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2018. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66.

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.

Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. 2017. On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning*, pages 3570–3578. PMLR.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022a. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. *arXiv preprint arXiv:2203.02167*.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022b. Language models as knowledge embeddings. *arXiv preprint arXiv:2206.12617*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI conference on artificial intelligence*, 28(1).

Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. 2016. Full-capacity unitary recurrent neural networks. *Advances in neural information processing systems*, 29.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279.

Bishan Yang and Tom Mitchell. 2019. Leveraging knowledge bases in lstms for improving machine reading. *arXiv preprint arXiv:1902.09091*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

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

Hongyi Zhang and Suvrit Sra. 2016. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638. PMLR.

Ningyu Zhang, Xin Xie, Xiang Chen, Yongheng Wang, Xu Cheng, and Huajun Chen. 2022. Reasoning through memorization: Nearest neighbor knowledge graph embeddings. *arXiv preprint arXiv:2201.05575*.

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

Dataset	model	lr-entity	lr-relation	optimizer	negative samples
WN18RR(dim=500)	TransE	0.001	-	Adam	300
	RotatE	0.1	-	Adagrad	300
	QuatE	0.2	-	Adagrad	300
	OrthogonalE(2×2)	0.2	0.02	-	300
	OrthogonalE(3×3)	0.2	0.02	-	300
FB15k-237(dim=1000)	TransE	0.05	-	Adam	300
	RotatE	0.1	-	Adagrad	300
	QuatE	0.2	-	Adagrad	300
	OrthogonalE(2×2)	0.5	0.06	-	300
	OrthogonalE(3×3)	0.5	0.06	-	300

Table 5: Best hyperparameters of our approach and several composite models. In the table, the lr-entity values corresponding to TransE, RotatE, and QuatE refer to the learning rate for the entire model. For the OrthogonalE model, we employ RiemannianAdam for relation optimization and Adagrad for entity optimization, as detailed in the Approach section.

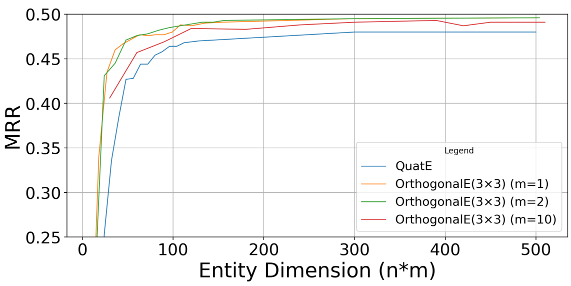


Figure 9: MRR accuracy comparison of QuatE and OrthogonalE(3×3) models across varying entity dimensions $(n * m)$ on WN18RR.

A Appendix

A.1 More information about Experiment setup

Dataset WN18RR is a subset of WN18 (Dettmers et al., 2018) which is contained in WordNet (Miller, 1995). FB15K-237 is a subset of FB15K, whichh is a subset of Freebase (Bollacker et al., 2008), a comprehensive KG including data about common knowledge. All three datasets were designed for KGE, and we employ them for KGE tasks, and all three datasets have no individual people or offensive content.

Implementation The training of models was carried out on two A6000 GPUs, which boasts 48GB of memory. Specifically, for the OrthogonalE model and its related flexible versions, the training durations were roughly 5 hour for the WN18RR dataset, 30 hours for FB15K-237. Our experiments were facilitated by leveraging PyTorch and Numpy as essential tools. Furthermore, We use ChatGPT in our paper writing. Finally, we obtain results by selecting the maximum values from three random seeds for Table 3 and using a single run for other results.

A.2 More results about Entity Matrix

Fig. 9 compare MRR accuracies of QuatE and OrthogonalE (3×3) over different fixed entity dimensions $n*m$ on WN18RR. The experimental results, consistent with those discussed in the section 5.2.4, demonstrate our model's flexibility in controlling entity dimension through variable m without unnecessarily increasing relation size.

Furthermore, for OrthogonalE(2×2) on WN18RR dataset, Fig. 5 result (with $m = 7$ yielding MRR=0.483) suggests that a relation matrix of 40×40 (20 parameters), compared to a dimension of 500×500 (250 parameters) in Table 3, can achieve comparably high performance, thus demonstrating that entity matrix method significantly reduces the need for excessive relation parameters.

A.3 Proof of Relation Patterns

OrthogonalE is capable of representing the four kinds of relational patterns: *Symmetry*, *Antisymmetry*, *Inversion*, and *Non-commutative Composition*. We present the following propositions to formalize this capability:

Proposition 1. *OrthogonalE can represent Symmetry relation pattern.*

Proof. If $(\mathbf{e}_H, \mathbf{e}_R, \mathbf{e}_T) \in \mathcal{E}$ and $(\mathbf{e}_T, \mathbf{e}_R, \mathbf{e}_H) \in$

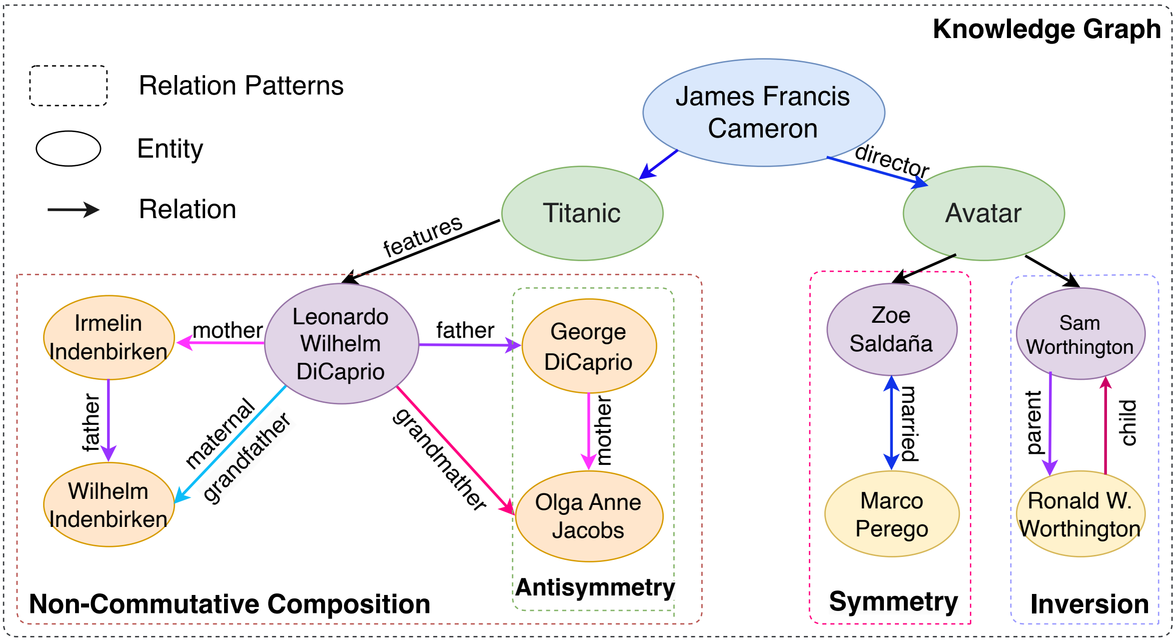


Figure 10: Toy examples for *symmetry*, *antisymmetry*, *inversion*, and *non-commutative composition* relation patterns.

$\mathcal{E}$, we have

$$\begin{aligned} &\mathbf{e}_R \cdot \mathbf{e}_H = \mathbf{e}_T \wedge \mathbf{e}_R \cdot \mathbf{e}_T = \mathbf{e}_H \\ &\Rightarrow \mathbf{e}_R \cdot \mathbf{e}_R = \mathrm{I} \\ &\Rightarrow \mathbf{X}_i \cdot \mathbf{X}_i = \mathrm{I} \end{aligned} \tag{6}$$

Proposition 2. *OrthogonalE can represent Antisymmetry relation pattern.*

Proof. If $(\mathbf{e}_H, \mathbf{e}_R, \mathbf{e}_T) \in \mathcal{E}$ and $(\mathbf{e}_T, \mathbf{e}_R, \mathbf{e}_H) \notin \mathcal{E}$, we have

$$\begin{aligned} &\mathbf{e}_R \cdot \mathbf{e}_H = \mathbf{e}_T \wedge \mathbf{e}_R \cdot \mathbf{e}_T \neq \mathbf{e}_H \\ &\Rightarrow \mathbf{e}_R \cdot \mathbf{e}_R \neq \mathrm{I} \\ &\Rightarrow \mathbf{X}_i \cdot \mathbf{X}_i \neq \mathrm{I} \end{aligned} \tag{7}$$

Proposition 3. *OrthogonalE can represent Inversion relation pattern.*

Proof. If $(\mathbf{e}_H, \mathbf{e}_R^1, \mathbf{e}_T) \in \mathcal{E}$ and $(\mathbf{e}_T, \mathbf{e}_R^2, \mathbf{e}_H) \in \mathcal{E}$, we have

$$\begin{aligned} &\mathbf{e}_R^1 \cdot \mathbf{e}_H = \mathbf{e}_T \wedge \mathbf{e}_R^2 \cdot \mathbf{e}_T = \mathbf{e}_H \\ &\Rightarrow \mathbf{e}_R^1 \cdot \mathbf{e}_R^2 = \mathrm{I} \\ &\Rightarrow \mathbf{X}_i^1 \cdot \mathbf{X}_i^2 = \mathrm{I} \end{aligned} \tag{8}$$

Proposition 4. *OrthogonalE can represent Non-commutative Composition relation pattern.*

Proof. If $(\mathbf{e}_H, \mathbf{e}_R^1, \mathbf{e}_T) \in \mathcal{E}$ and $(\mathbf{e}_T, \mathbf{e}_R^2, \mathbf{e}_P) \in \mathcal{E}$ and $(\mathbf{e}_H, \mathbf{e}_R^3, \mathbf{e}_P) \in \mathcal{E}$, we have

$$\begin{aligned} &\mathbf{e}_R^1 \cdot \mathbf{e}_H = \mathbf{e}_T \wedge \mathbf{e}_R^2 \cdot \mathbf{e}_T = \mathbf{e}_P \wedge \mathbf{e}_R^3 \cdot \mathbf{e}_H = \mathbf{e}_P \\ &\Rightarrow \mathbf{e}_R^2 \cdot \mathbf{e}_R^1 = \mathbf{e}_R^3 \\ &\Rightarrow \mathbf{X}_i^2 \cdot \mathbf{X}_i^1 = \mathbf{X}_i^3 \end{aligned} \tag{9}$$

$$\mathbf{X}_i \in \mathbb{R}^{d \times d} \begin{cases} \text{is Commutative}, & \text{if } d = 2 \\ \text{is Non-commutative}, & \text{if } d > 2 \end{cases} \tag{10}$$

The property of non-commutative composition dictates that the sequence of $\mathbf{X}_i^1$ and $\mathbf{X}_i^2$ cannot be exchanged. Given that $\mathbf{X}_i \in \mathbb{R}^{d \times d}$ represents an orthogonal matrix, we consider two situations. In the first scenario, when $d = 2$, $\mathbf{X}_i$ is a special case corresponding to a 2-dimensional rotation matrix, analogous to the RotatE (Sun et al., 2019), and is therefore commutative, not exhibiting non-commutative composition. However, for $d > 2$, $\mathbf{X}_i$ is non-commutative, thus can represent non-commutative composition relation pattern.

To gain a clearer understanding of the proof process, we use symmetry as an illustrative example to introduce the proof section, specifically referring to equation 6 in the paper. Initially, we assume that if a relation $\mathbf{e}_R$ in the OrthogonalE model exhibits the property of symmetry, then we can identify two related KG triples: $(\mathbf{e}_H, \mathbf{e}_R, \mathbf{e}_T) \in \mathcal{E}$ and

$(\mathbf{e}_T, \mathbf{e}_R, \mathbf{e}_H) \in \mathcal{E}$. For instance, $\mathbf{e}_H$ ($\mathbf{e}_R$: is similar to) $\mathbf{e}_T$ and $\mathbf{e}_T$ ($\mathbf{e}_R$: is similar to) $\mathbf{e}_H$. Since both triples are trained by the OrthogonalE model, they adhere to the OrthogonalE equation (as depicted in Fig. 2). Consequently, we can derive that $\mathbf{e}_R{\cdot}\mathbf{e}_H = \mathbf{e}_T$ and $\mathbf{e}_R{\cdot}\mathbf{e}_T = \mathbf{e}_H$. By combining and simplifying these two equations, we can conclude that $\mathbf{e}_R \cdot \mathbf{e}_R = \mathrm{I}$ (Identity matrix). This means that if we can identify such an $\mathbf{e}_R$ that satisfies $\mathbf{e}_R \cdot \mathbf{e}_R = \mathrm{I}$, it demonstrates that the OrthogonalE model can capture the symmetry relation pattern. For $\mathbf{e}_R{\cdot}\mathbf{e}_R = \mathrm{I}$, we understand that $\mathbf{e}_R$ is composed of several block-diagonal orthogonal matrices $\mathbf{X}_i$, as shown in Fig. 2). Ultimately, this reduces to finding $\mathbf{X}_i \cdot \mathbf{X}_i = \mathrm{I}$ to satisfy $\mathbf{e}_R \cdot \mathbf{e}_R = \mathrm{I}$. We can identify corresponding orthogonal matrices $\mathbf{X}_i$ that satisfy $\mathbf{X}_i \cdot \mathbf{X}_i = \mathrm{I}$, which demonstrates that OrthogonalE can fulfill the property of symmetry.

A.4 More experiments on relation pattern

Symmetry and Antisymmetry Fig. 11 shows histograms of additional examples of relation embeddings for *symmetry* and *antisymmetry* relation patterns. Furthermore, it displays examples of two *symmetry* and two *antisymmetry* relations from both the WN18RR and FB15K-237 datasets.

Composition Firstly, we add $\mathbf{e}_R^1$, $\mathbf{e}_R^2$, and $\mathbf{e}_R^3$ from OrthogonalE(3×3) for comparison with the three composition relations in Fig. 8 from OrthogonalE(2×2). From Fig. 12(a, b, c, d), we observe that OrthogonalE(2×2) performs better in composition than OrthogonalE(3×3).

Secondly, we aim to explore more about the *non-commutative composition* relation pattern, so we select $\mathbf{e}_R^4$, $\mathbf{e}_R^5$, and $\mathbf{e}_R^6$, three *non-commutative composition* relations, for our study. Notably, $\mathbf{e}_R^7$, $\mathbf{e}_R^8$, and $\mathbf{e}_R^9$ share the same relational meanings as $\mathbf{e}_R^4$, $\mathbf{e}_R^5$, and $\mathbf{e}_R^6$, respectively, with the distinction that the former are relations within OrthogonalE(3×3), while the latter are within OrthogonalE(2×2). Figs. 12(h, l), using $\mathbf{e}_R^5 \cdot \mathbf{e}_R^4 - \mathbf{e}_R^6$ and $\mathbf{e}_R^4 \cdot \mathbf{e}_R^5 - \mathbf{e}_R^6$ respectively, show nearly indistinguishable histograms, indicating that swapping the sequence of the two relations in OrthogonalE(2×2) does not affect the outcome of the composition, suggesting it is commutative. Conversely, Figs. 12(m, n), using $\mathbf{e}_R^8 \cdot \mathbf{e}_R^7 - \mathbf{e}_R^9$ and $\mathbf{e}_R^7 \cdot \mathbf{e}_R^8 - \mathbf{e}_R^9$, show that the former results in a trend closer to or equal to 0 more distinctly than the latter, implying that changing the sequence of relations affects the outcome, thereby demonstrating the non-commutative nature of relations in OrthogonalE(3×3). In conclusion, even though OrthogonalE(2×2) generally outperforms OrthogonalE(3×3) in composition relation patterns, the comparative analysis reveals that OrthogonalE(3×3) indeed possesses non-commutative composition properties, following the equation 9 and 10.

A.5 Introduction of Relation Patterns

We can observe several relation patterns in KGs, including *symmetry*, *antisymmetry*, *inversion*, and *composition* (both commutative and non-commutative). Detailed examples have been shown in Fig. 10.

Symmetry and Antisymmetry Certain relations demonstrate *symmetry*, indicating that the validity of a relation between entities x and y $((r_1(x, y) \Rightarrow r_1(y, x)))$ (for instance, *is married to*) is equally valid in the opposite direction (namely, from y to x). Conversely, other relations are characterized by *antisymmetry* $((r_1(x, y) \Rightarrow \neg r_1(y, x)))$, signifying that if a relation is applicable between x and y (such as *is father of*), it is inapplicable in the reverse direction (from y to x).

Inversion Relations can also exhibit inversion $((r_1(x, y) \Leftrightarrow r_2(y, x)))$, where reversing the direction of a relation effectively transforms it into another relation (for example, *is child of* and *is parent of*).

Composition The composition of relations $((r_1(x, y) \cap r_2(y, z) \Rightarrow r_3(x, z)))$ signifies a crucial pattern wherein merging two or more relations facilitates the deduction of a novel relation. Such compositions might be commutative, where the sequence of relations is irrelevant, or non-commutative, where the sequence significantly influences the outcome. In scenarios where the order of relations is pivotal, as illustrated by the relationship where B is the mother of A's father and E is the father of A's mother, non-commutative composition $((r_1(x, y) \cap r_2(y, z) \neq (r_2(x, y) \cap r_1(y, z))$ becomes essential. While commutative compositions would consider B and E as identical, non-commutative compositions recognize them as distinct.

A.6 Other baseline KGE model

In recent times, several significant performance methods have been developed, as detailed for

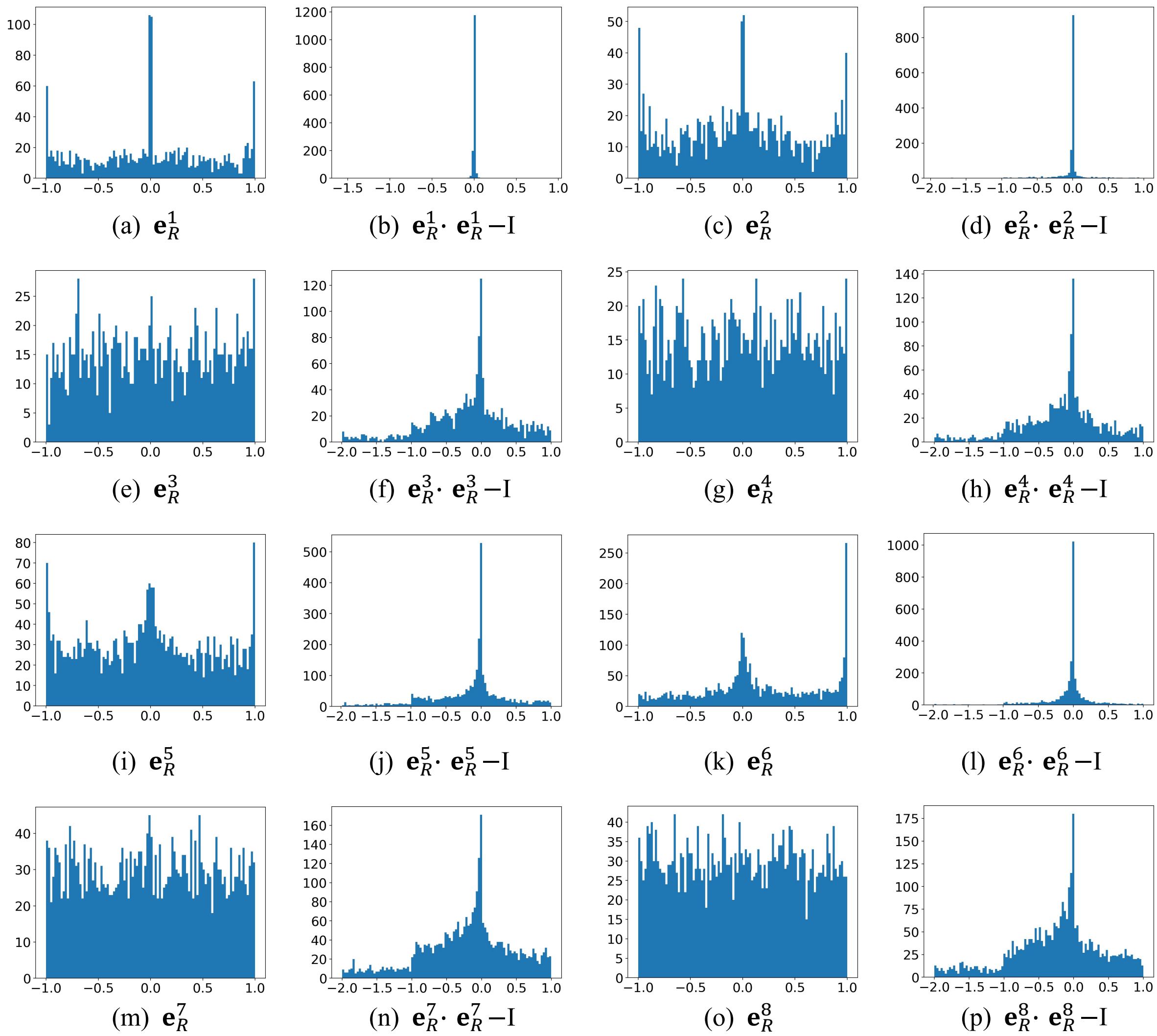
KGE Model	Description	MRR Accuracy
MoCoSA(He et al., 2023)	Language Models	.696
SimKGC(Wang et al., 2022a)	Language Models	.671
LERP(Han et al., 2023)	Additional Contextual Information (Logic Rules)	.622
C-LMKE(Wang et al., 2022b)	Language Models	.598
KNN-KGE(Zhang et al., 2022)	Language Models	.579
HittER(Chen et al., 2020)	Language Models	.503
OrthogonalE(10×10)	-	.494

Table 6: Other baseline models in WN18RR dataset.

WN18RR in Table 6. Among these, MoCoSA(He et al., 2023), SimKGC(Wang et al., 2022a), C-LMKE(Wang et al., 2022b), KNN-KGE(Zhang et al., 2022), and HittER(Chen et al., 2020) mainly utilize Language Models (LMs) to enrich dataset semantic information, thereby achieving superior outcomes. Conversely, LERP(Han et al., 2023) does not employ LMs but leverages additional contextual information (logic rules) beyond the dataset to fill information gaps in entities and relations. On the other hand, methods such as TransE(Bordes et al., 2013), RotatE(Sun et al., 2019), and the OrthogonalE method introduced in this paper depend solely on the inherent data and information of the KGE dataset itself. These methods, based on specific mathematical rules and algorithms, do not incorporate any external information and thus do not operate as black-box approaches like LLMs. Consequently, these dataset-dependent methods remain highly valuable for KGE research.

A.7 hyperparameter

All the hyperparameter settings have been shown in Table 5



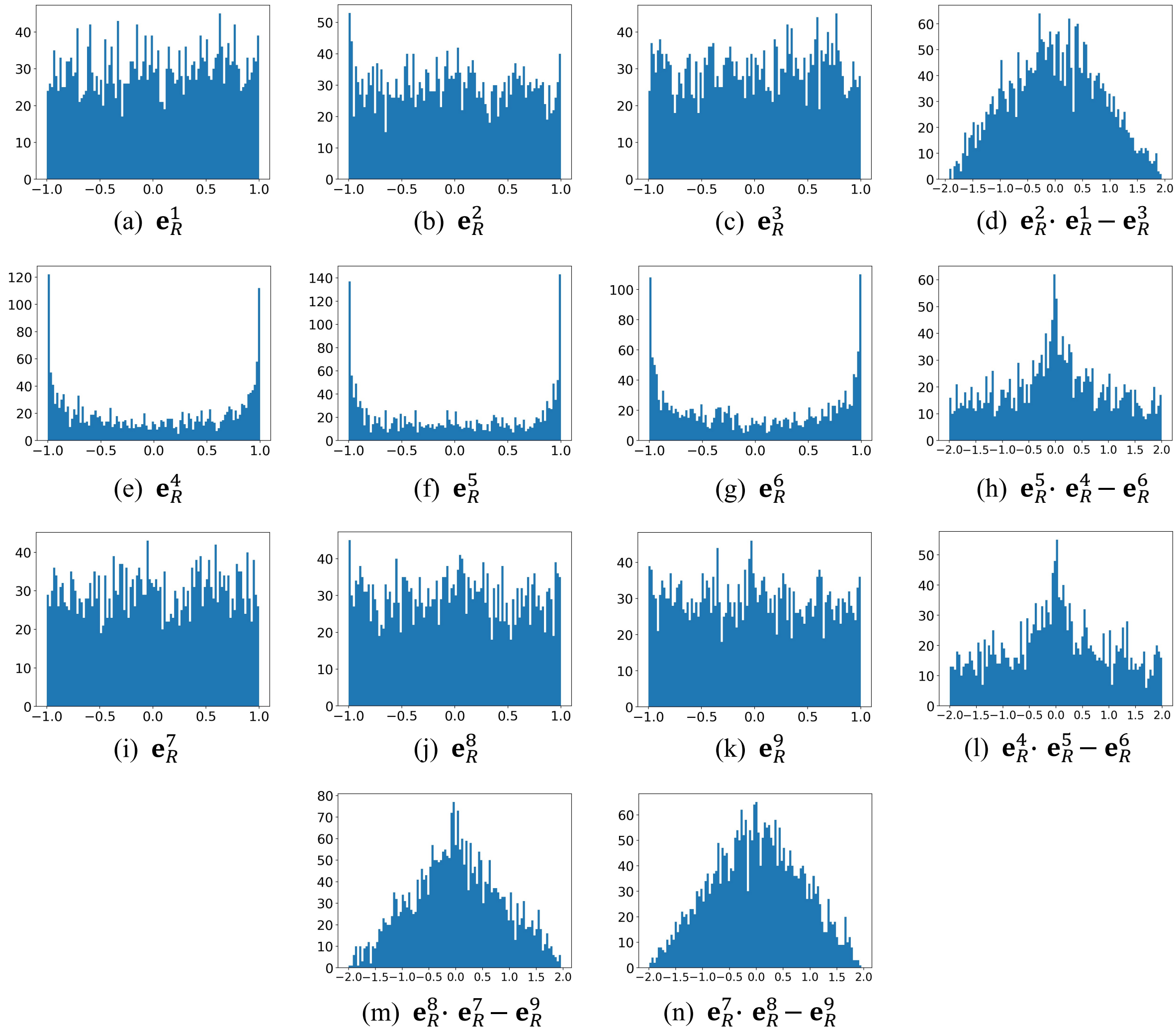
(a) $\mathbf{e}_R^1$ (b) $\mathbf{e}_R^1 \cdot \mathbf{e}_R^1 - \mathrm{I}$ (c) $\mathbf{e}_R^2$ (d) $\mathbf{e}_R^2 \cdot \mathbf{e}_R^2 - \mathrm{I}$

(e) $\mathbf{e}_R^3$ (f) $\mathbf{e}_R^3 \cdot \mathbf{e}_R^3 - \mathrm{I}$ (g) $\mathbf{e}_R^4$ (h) $\mathbf{e}_R^4 \cdot \mathbf{e}_R^4 - \mathrm{I}$

(i) $\mathbf{e}_R^5$ (j) $\mathbf{e}_R^5 \cdot \mathbf{e}_R^5 - \mathrm{I}$ (k) $\mathbf{e}_R^6$ (l) $\mathbf{e}_R^6 \cdot \mathbf{e}_R^6 - \mathrm{I}$

(m) $\mathbf{e}_R^7$ (n) $\mathbf{e}_R^7 \cdot \mathbf{e}_R^7 - \mathrm{I}$ (o) $\mathbf{e}_R^8$ (p) $\mathbf{e}_R^8 \cdot \mathbf{e}_R^8 - \mathrm{I}$

Figure 11: Histograms of relation embeddings for *symmetry* and *antisymmetry* relation patterns, where e_R^1 represents `_derivationally_related_form`, e_R^2 represents `_instance_hypernym`, e_R^3 represents `_also_see`, e_R^4 represents `_verb_group`, e_R^5 represents `/media_common/netflix_genre/titles`, e_R^6 represents `/film/film/genre`, e_R^7 represents `/award/award_category/category_of` , and e_R^8 represents `/people/person/gender`. From the WN18RR dataset, we select e_R^1, e_R^2 e_R^3, $\mathrm{e}_R^4$and to represent *Symmetry* and *Antisymmetry*, respectively, and obtain their relation embeddings using the OrthogonalE(3×3) model with n=501 and m=1. Similarly, from the FB15K-237 dataset, we select e_R^5, e_R^6, and e_R^7, e_R^8 as representations for *symmetry* and *antisymmetr*, respectively, and acquire their relation embeddings under the OrthogonalE(3×3) model with n=999 and m=1.



(a) $\mathbf{e}_R^1$ (b) $\mathbf{e}_R^2$ (c) $\mathbf{e}_R^3$ (d) $\mathbf{e}_R^2 \cdot \mathbf{e}_R^1 - \mathbf{e}_R^3$

(e) $\mathbf{e}_R^4$ (f) $\mathbf{e}_R^5$ (g) $\mathbf{e}_R^6$ (h) $\mathbf{e}_R^5 \cdot \mathbf{e}_R^4 - \mathbf{e}_R^6$

(i) $\mathbf{e}_R^7$ (j) $\mathbf{e}_R^8$ (k) $\mathbf{e}_R^9$ (l) $\mathbf{e}_R^4 \cdot \mathbf{e}_R^5 - \mathbf{e}_R^6$

(m) $\mathbf{e}_R^8 \cdot \mathbf{e}_R^7 - \mathbf{e}_R^9$ (n) $\mathbf{e}_R^7 \cdot \mathbf{e}_R^8 - \mathbf{e}_R^9$

Figure 12: Histograms of relation embeddings for *composition* relation patterns, where $\mathbf{e}_R^1$ represents `/location/administrative_division/country`, $\mathbf{e}_R^2$ represents `/location/hud_county_place/place`, $\mathbf{e}_R^3$ represents `/base/aareas/schema/administrative_area/capital`, $\mathbf{e}_R^4$ represents `/award/award_nominee/award_nominations./award/award_nomination/nominated_for`, $\mathbf{e}_R^5$ represents `/award/award_category/winners./award/award_honor/award_winner`, and $\mathbf{e}_R^6$ represents `/award/award_category/nominees./award/award_nomination/nominated_for`. $\mathbf{e}_R^7$, $\mathbf{e}_R^8$, and $\mathbf{e}_R^9$ have the same relational meanings as $\mathbf{e}_R^4$, $\mathbf{e}_R^5$, and $\mathbf{e}_R^6$, respectively, the difference lies in that the former are relations within the OrthogonalE(3×3) model, while the latter are from the OrthogonalE(2×2) model. All these relations are selected from the FB15K-237 dataset. $\mathbf{e}_R^1$, $\mathbf{e}_R^2$, $\mathbf{e}_R^3$, $\mathbf{e}_R^7$, $\mathbf{e}_R^8$, and $\mathbf{e}_R^9$ are relation embeddings under the OrthogonalE(3×3) model with n=999 and m=1, while $\mathbf{e}_R^4$, $\mathbf{e}_R^5$, and $\mathbf{e}_R^6$ are relation embeddings under the OrthogonalE(2×2) model with n=1000 and m=1