Compounding Geometric Operations for Knowledge Graph Completion

Xiou Ge[1*], **Yun-Cheng Wang**[1], **Bin Wang**[2], **C.-C. Jay Kuo**[1]
[1]University of Southern California, Los Angeles, California, USA
[2]National University of Singapore, Singapore
{xiouge, yunchenw, jckuo}@usc.edu, bwang28c@gmail.com

Abstract

Geometric transformations including translation, rotation, and scaling are commonly used operations in image processing. Besides, some of them are successfully used in developing effective knowledge graph embedding (KGE). Inspired by the synergy, we propose a new KGE model by leveraging all three operations in this work. Since translation, rotation, and scaling operations are cascaded to form a composite one, the new model is named CompoundE. By casting CompoundE in the framework of group theory, we show that quite a few distanced-based KGE models are special cases of CompoundE. CompoundE extends the simple distance-based scoring functions to relation-dependent compound operations on head and/or tail entities. To demonstrate the effectiveness of CompoundE, we perform three prevalent KG prediction tasks including link prediction, path query answering, and entity typing, on a range of datasets. CompoundE outperforms extant models consistently, demonstrating its effectiveness and flexibility.[1]

1 Introduction

Knowledge graphs (KGs) such as DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), NELL (Carlson et al., 2010), Wikidata (Vrandečić and Krötzsch, 2014), Freebase (Bollacker et al., 2008), and ConceptNet (Speer et al., 2017) have been created and made available to the public to facilitate research on KG modeling and applications. Triples, denoted by (h, r, t), are basic elements of a KG, where h and t are head and tail entities while r is the relation connecting them. KG representation learning, also known as knowledge graph embedding (KGE), has been intensively studied in recent years. Yet, it remains one of the most fundamental problems in Artificial Intelligence (AI) research. KGE is critical to many downstream applications such as question answering (Guu et al., 2015) knowledge integration (Chen et al., 2017), text analysis (Li et al., 2019), entity classification (Zhao et al., 2020; Ge et al., 2022), etc. There are several challenges in the design of good KGE models. Complex relation patterns (e.g. 1-to-N, N-to-1, and N-to-N, antisymmetric, transitive, hierarchical relations, etc.) remain difficult to model. Also, each of extant KGE models has its own strengths and weaknesses. It is desired yet unclear how to design a KGE model that leverages strengths of some models and complements weaknesses of others.

Geometric operations such as translation and rotation belong to the family of affine transformations. These operations have been used to build effective KGE models such as TransE, RotatE, and PairRE. Previous KGEs often use a single type of operation to model all relation patterns with different properties. This could be problematic since each operator may have modeling limitations. A synergy of different transformations may complement the weaknesses of individual operators. In fact, generic compound operations yielded from a cascade of affine transformations find numerous applications in image processing (Pratt, 2013), including image warping (Wolberg, 1990), image morphing (Seitz and Dyer, 1996), and robot motion planning (LaValle, 2006). Motivated by the synergy, we propose a new KGE model to address the above-mentioned challenges. Since translation, rotation, and scaling operations are cascaded to form a compound operation, the proposed KGE model is named CompoundE. Compound operations inherit many desirable properties from the affine group, allowing CompoundE to model complex relations in different KGs.

There are four main contributions of this work. They are summarized below.

- We present a novel KG embedding model called CompoundE, which combines three

*Corresponding author
[1]Our source code is available at https://github.com/hughxiouge/CompoundE

fundamental operations in the affine group and offers a wide range of designs.

- It is proved mathematically that CompoundE can handle complex relation types in KG thanks to unique properties of the affine group.
- We apply CompoundE to perform three important KG prediction tasks, including link prediction, path query answering, and entity typing on widely adopted KG benchmarking datasets extracted from Freebase, WordNet, Wikidata, and YAGO. CompoundE consistently outperforms prior work.
- Against large-scale datasets containing millions of entities under the memory constraint, CompoundE outperforms other benchmarking methods by a big margin with fewer parameters.

The rest of this paper is organized as follows. Recent KGE models for both distance-based and entity-Transformation-based categories are first reviewed in Section 2. Then, we present CompoundE, show its relationship with previous KG embedding models, and explain the reason why it can model complex relations well in Section 3. Experiment details and performance comparisons are given in Section 4. Finally, concluding remarks are given and possible extensions are suggested in Section 5.

2 Related Work

2.1 Distance-based Models

Distance-based scoring function is a prevailing strategy in optimizing KGE. The main idea is to model a relation as a transformation that places head entity vectors in the proximity of their corresponding tail entity vectors, and vice versa. For a given triple, (h, r, t), the goal is to minimize the distance between h and t vectors after the transformation introduced by r. TransE (Bordes et al., 2013) is one of the earlier KGE models that interpret relations between entities as translation operations in the vector space. However, this simple approach is ineffective in modeling 1-to-N, N-to-1, N-to-N, and symmetric relations. To better encode complex relations, TransH (Wang et al., 2014) enables the relation-specific entity representation by projecting each entity to a relation-specific hyperplane. In order to represent symmetric relations, RotatE (Sun et al., 2019) models entities in the complex vector space and interprets a relation as a rotation instead of a translation. The self-adversarial negative sampling contributes to RotatE's performance improvement as compared to its predecessors. However, RotatE does not handle the hierarchical structure appearing in many KGs. MuRP (Balažević et al., 2019) and ROTH (Chami et al., 2020) leverage the power of hyperbolic curvature to better capture the hierarchical structure in KGs.

2.2 Entity-Transformation-based Models

Adding relation-specific transformation to baseline models is another popular line of work. Although TransH (Wang et al., 2014) enables the translation approach to model complex relations, the orthogonal projection prevents the model from encoding inverse and composition relations. Instead, TransR (Lin et al., 2015) models relations and entities in two different spaces. However, the relation-specific transformation is a dense matrix and significantly increase parameter complexity. To alleviate this problem, TransD (Ji et al., 2015) comes up with dynamic mapping matrices using relation and entity projection vectors. Similarly, TranSparse (Ji et al., 2016) enforces the relation projection matrix to be sparse. More recently, PairRE (Chao et al., 2021) performs transformations on both heads and tails. Specifically, head and tail entities had a Hadamard product with their respective weight vectors $\mathbf{r^H}$ and $\mathbf{r^T}$. This elementwise multiplication is nothing but the scaling operation. SFBR (Liang et al., 2021) introduces a semantic filter that includes a scaling and translation component. STaR (Li and Yang, 2022) adopts a similar approach but adds a rotation operator to design the bilinear product matrix for semantic matching scoring functions. ReflectE (Zhang et al., 2022a) models a relation as the Householder reflection. Different composition of geometric operations inspire us to invent CompoundE embedding in this paper.

3 Methodology

Translation, rotation, and scaling transformations appear frequently in engineering applications. In image processing, a cascade of translation, rotation, and scaling operations offers a set of image manipulation techniques. Such compound operations can be used to develop a new KGE model called CompoundE. We provide an illustration of CompoundE and comparison with previous KGE in Fig. 1.

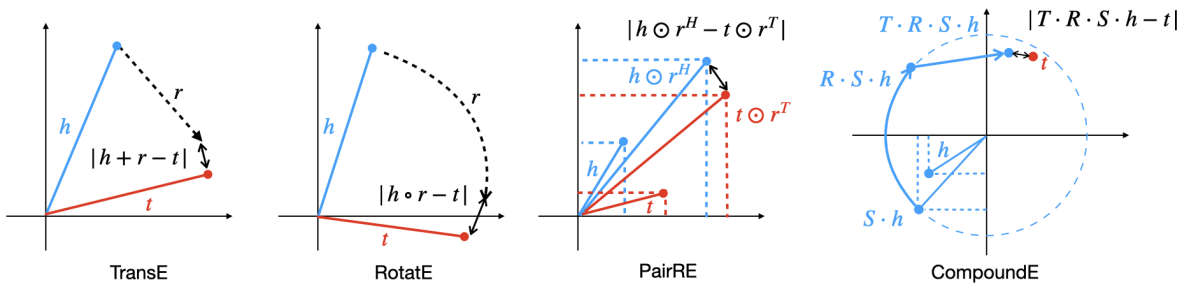


Figure 1: An illustration of previous distance-based KGE models and CompoundE.

3.1 Definition of CompoundE

Three forms of CompoundE scoring function can be written as

- CompoundE-Head

$$f_r(h,t) = \|\mathbf{T_r} \cdot \mathbf{R_r} \cdot \mathbf{S_r} \cdot \mathbf{h} - \mathbf{t}\|, \quad (1)$$

- CompoundE-Tail

$$f_r(h,t) = \|\mathbf{h} - \hat{\mathbf{T}}_\mathbf{r} \cdot \hat{\mathbf{R}}_\mathbf{r} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{t}\|, \quad (2)$$

- CompoundE-Complete

$$f_r(h,t) = \|\mathbf{T_r} \cdot \mathbf{R_r} \cdot \mathbf{S_r} \cdot \mathbf{h} - \hat{\mathbf{T}}_\mathbf{r} \cdot \hat{\mathbf{R}}_\mathbf{r} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{t}\|, \quad (3)$$

where $\mathbf{h}, \mathbf{t}$ denote head and tail entity embeddings, $\mathbf{T_r}, \mathbf{R_r}, \mathbf{S_r}$ denote the translation, rotation, and scaling operations for the head entity embedding, and $\hat{\mathbf{T}}_\mathbf{r}, \hat{\mathbf{R}}_\mathbf{r}, \hat{\mathbf{S}}_\mathbf{r}$ denote the counterparts for the tail entity embedding, respectively. These constituent operators are relation-specific. To generalize, any order or subset of translation, rotation, and scaling component can be a valid instance of CompoundE. Since matrix multiplications are non-commutative, different orders of cascading the constituent operators result in distinct CompoundE operators. Performance difference between these variations are discussed in Section B of the appendix.

3.2 CompoundE as An Affine Group

Most analysis in previous work was restricted to the Special Euclidean Group $\mathbf{SE}(n)$ (Cao et al., 2022). Yet, we will show that CompoundE is not a special Euclidean group but an affine group. To proceed, we first formally introduce the lie group and three special groups as below.

Definition 3.1. *A Lie group is a continuous group that is also a differentiable manifold.*

Several Lie group examples are given below.

- The real vector space, $\mathbb{R}^n$, with the canonical addition as the group operation.
- The real vector space excluding zero, $(\mathbb{R}\backslash\{0\})$, with the element-wise multiplication as the group operation.
- The general linear group, $\mathbf{GL}_n(\mathbb{R})$, with the canonical matrix multiplication as the group operation.

Furthermore, the following three special groups are commonly used.

Definition 3.2. *The special orthogonal group is defined as*

$$\mathbf{SO}(n) = \left\{ \mathbf{A} \middle| \mathbf{A} \in \mathbf{GL}_n(\mathbb{R}), \mathbf{A}^\top \mathbf{A} = \mathbf{I}, \det(\mathbf{A}) = 1 \right\}. \quad (4)$$

Definition 3.3. *The special Euclidean group is defined as*

$$\mathbf{SE}(n) = \left\{ \mathbf{A} \middle| \mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{R} \in \mathbf{SO}(n), \mathbf{v} \in \mathbb{R}^n \right\}. \quad (5)$$

Definition 3.4. *The affine group is defined as*

$$\mathbf{Aff}(n) = \left\{ \mathbf{M} \middle| \mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{A} \in \mathbf{GL}_n(\mathbb{R}), \mathbf{v} \in \mathbb{R}^n \right\}. \quad (6)$$

By comparing Eqs. (5) and (6), we see that $\mathbf{SE}(n)$ is a subset of $\mathbf{Aff}(n)$.

Without loss of generality, consider $n = 2$. If $\mathbf{M} \in \mathbf{Aff}(2)$, we have

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{A} \in \mathbb{R}^{2\times 2}, \mathbf{v} \in \mathbb{R}^2. \quad (7)$$

The 2D translational matrix can be written as

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

Dataset	#Entities	#Relations	#Training	#Validation	#Test	Ave. Deg.	Scale
ogbl-wikikg2	2,500,604	535	16,109,182	429,456	598,543	12.2	Large
FB15k-237	14,541	237	272,115	17,535	20,466	19.74	Medium
WN18RR	40,943	11	86,835	3,034	3,134	2.19	Small

Table 1: Link Prediction Datasets Statistics.

while the 2D rotational matrix can be expressed as

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

It is easy to verify that they are both special Euclidean groups (i.e. $\mathbf{T} \in \mathbf{SE}(2)$ and $\mathbf{R} \in \mathbf{SE}(2)$). On the other hand, the 2D scaling matrix is in form of

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

It is not a special Euclidean group but an affine group of $n = 2$ (i.e., $\mathbf{S} \in \mathbf{Aff}(2)$).

Compounding translation and rotation operations, we can get a transformation in the special Euclidean group,

$$\begin{aligned} \mathbf{T} \cdot \mathbf{R} &= \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & v_x \\ \sin(\theta) & \cos(\theta) & v_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbf{SE}(2). \end{aligned} \tag{11}$$

Yet, if we add the scaling operation, the compound will belong to the Affine group. One of such compound operator can be written as

$$\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} = \begin{bmatrix} s_x \cos(\theta) & -s_y \sin(\theta) & v_x \\ s_x \sin(\theta) & s_y \cos(\theta) & v_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbf{Aff}(2). \tag{12}$$

When $s_x \neq 0$ and $s_y \neq 0$, the compound operator is invertible. It can be written in form of

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{13}$$

3.3 Relation with Other Distance-based KGE Models

CompoundE is a general form of quite a few distance-based KGE models. That is, we can derive their scoring functions from that of CompoundE by setting translation, scaling, and rotation operations to certain forms. Four examples are given below.

Derivation of TransE (Bordes et al., 2013). We begin with CompoundE-Head and set its rotation component to identity matrix $\mathbf{I}$ and scaling parameters to $\mathbf{1}$. Then, we get the scoring function of TransE as

$$\begin{aligned} f_r(h,t) &= \|\mathbf{T_r} \cdot \mathbf{I} \cdot \mathrm{diag}(\mathbf{1}) \cdot \mathbf{h} - \mathbf{t}\| \\ &= \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|. \end{aligned} \tag{14}$$

Derivation of RotatE (Sun et al., 2019). We can derive the scoring function of RotatE from CompoundE-Head by setting the translation component to $\mathbf{I}$ (translation vector $\mathbf{t} = \mathbf{0}$) and scaling component to $\mathbf{1}$.

$$\begin{aligned} f_r(h,t) &= \|\mathbf{I} \cdot \mathbf{R_r} \cdot \mathrm{diag}(\mathbf{1}) \cdot \mathbf{h} - \mathbf{t}\| \\ &= \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|. \end{aligned} \tag{15}$$

Derivation of LinearRE (Peng and Zhang, 2020). We can add back the translation component for the head transformation:

$$\begin{aligned} f_r(h,t) &= \|\mathbf{T_r} \cdot \mathbf{I} \cdot \mathbf{S_r} \cdot \mathbf{h} - \mathbf{I} \cdot \mathbf{I} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{t}\| \\ &= \|\mathbf{h} \odot \mathbf{r^H} + \mathbf{r} - \mathbf{t} \odot \mathbf{r^T}\|. \end{aligned} \tag{16}$$

Derivation of PairRE (Chao et al., 2021). CompoundE-Complete can be reduced to PairRE by setting both translation and rotation component to $\mathbf{I}$, for both head and tail transformation.

$$\begin{aligned} f_r(h,t) &= \|\mathbf{I} \cdot \mathbf{I} \cdot \mathbf{S_r} \cdot \mathbf{h} - \mathbf{I} \cdot \mathbf{I} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{t}\| \\ &= \|\mathbf{h} \odot \mathbf{r^H} - \mathbf{t} \odot \mathbf{r^T}\|. \end{aligned} \tag{17}$$

3.4 Properties of CompoundE

With a richer set of operations, CompoundE is more capable of modeling complex relations such as 1-to-N, N-to-1, and N-to-N relations in KG datasets. Modeling these relations are important since more than 98% of triples in FB15k-237 and WN18RR datasets involves complex relations. The importance of complex relation modeling is illustrated by two examples below. First, there is a need to distinguish different outcomes of relation compositions when modeling non-commutative relations. That is $r_1 \cdot r_2 \rightarrow r_3$ while $r_2 \cdot r_1 \rightarrow r_4$. For instance, r_1, r_2, r_3 and r_4 denote **isFatherOf**, **isMotherOf**, **isGrandfatherOf** and **isGrandmotherOf**, respectively. TransE and RotatE cannot make such

Datasets	FB15K-237				WN18RR			
Metrics	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**
text-based methods								
SimKGC (Wang et al., 2022)	0.666	0.587	0.717	0.800	0.336	0.249	0.362	0.511
KG-S2S (Chen et al., 2022)	0.574	0.531	0.595	0.661	0.336	0.257	0.373	0.498
embedding-based methods								
TransE (Bordes et al., 2013)	0.294	-	-	0.465	0.226	-	-	0.501
DistMult (Yang et al., 2015)	0.241	0.155	0.263	0.419	0.430	0.390	0.440	0.490
ComplEx (Trouillon et al., 2016)	0.247	0.158	0.275	0.428	0.440	0.410	0.460	0.510
RotatE (Sun et al., 2019)	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
ROTH (Chami et al., 2020)	0.348	0.252	0.384	0.540	**0.496**	0.449	**0.514**	**0.586**
PairRE (Chao et al., 2021)	0.351	0.256	0.387	0.544	-	-	-	-
FieldE (Nayyeri et al., 2021)	**0.36**	**0.27**	0.39	**0.55**	0.48	0.44	0.50	0.57
KGTuner (Zhang et al., 2022b)	0.352	0.263	0.387	0.530	0.484	0.440	0.506	0.562
IAS (Yang et al., 2022)	0.339	0.242	0.374	0.532	0.483	0.467	0.502	0.570
CAKE (Niu et al., 2022)	0.321	0.227	0.355	0.515	-	-	-	-
CompoundE	0.357	0.264	**0.393**	0.545	0.491	**0.450**	0.508	0.576

Table 2: Filtered ranking of link prediction for FB15k-237 and WN18RR.

distinctions since they are based on commutative relation embeddings. Second, to capture the hierarchical structure of relations, it is essential to build a good model for sub-relations. For example, r_1 and r_2 denote **isCapitalCityOf** and **cityLocatedInCountry**, respectively. Logically, **isCapitalCityOf** is a sub-relation of **cityLocatedInCountry** because if (h, r_1, t) is true, then (h, r_2, t) must be true. We provide mathematical proofs to show that CompoundE is capable of modeling symmetric/antisymmetric, inversion, composition, commutative/non-commutative, transitive, and sub-relations in Section A of the appendix.

Optimization We follow RotatE's negative sampling loss and the self-adversarial training strategy. The loss function of CompoundE can be written as

$$L_{\text{KGE}} = -\log \sigma(\zeta_1 - f_r(h,t)) - \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(f_r(h'_i, t'_i) - \zeta_1), \tag{18}$$

where σ is the sigmoid function, ζ_1 is a fixed margin hyperparameter, (h'_i, r, t'_i) is the i-th negative triple, and $p(h'_i, r, t'_i)$ is the probability of drawing negative triple (h'_i, r, t'_i). Given a positive triple, (h_i, r, t_i), the negative sampling distribution is

$$p(h'_j, r, t'_j | \{(h_i, r, t_i)\}) = \frac{\exp \alpha_1 f_r(h'_j, t'_j)}{\sum_i \exp \alpha_1 f_r(h'_i, t'_i)}, \tag{19}$$

where α_1 is the temperature of sampling.

4 Experiments

4.1 Link Prediction

Datasets. We conduct experiments on three widely used benchmarking datasets: ogbl-wikikg2, FB15k-237, and WN18RR. ogbl-wikikg2 is one of Open Graph Benchmark dataset (Hu et al., 2020) extracted from the Wikidata (Vrandečić and Krötzsch, 2014) KG. Its challenge is with designing embedding models that can scale to large KGs. FB15k-237 and WN18RR are extracted from the Freebase (Bollacker et al., 2008) and the WordNet (Miller, 1995), respectively. Inverse relations are removed from both to avoid the data leakage problem. Their main challenge lies in modeling symmetry/antisymmetry and composition relation patterns. The detailed statistics of the three datasets are shown in Table 1.

Evaluation Protocol. To evaluate the link prediction performance of CompoundE, we compute the rank of the ground truth entity in the list of top candidates. Since embedding models tend to rank entities observed in the training set higher, we compute the filtered rank to prioritize candidates that would result in unseen triples. We follow the convention (Wang et al., 2017; Ji et al., 2021) and adopt the Mean Reciprocal Rank (MRR) and Hits@k metrics to compare the quality of different KGE models. Higher MRR and Hits@k values indicate better model performance.

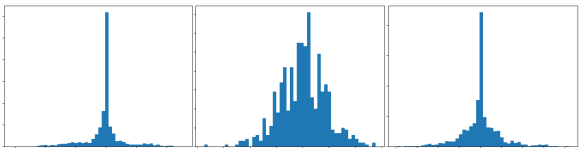
Performance Benchmarking. Tables 2 and 3 show the best performance of CompoundE and other benchmarking models for FB15k-237,

Datasets	ogbl-wikikg2		
Metrics	Dim	Valid MRR	Test MRR
AutoSF+NodePiece	100	0.5806	0.5703
ComplEx-RP	50	<u>0.6561</u>	<u>0.6392</u>
TransE	500	0.4272	0.4256
DistMult	500	0.3506	0.3729
ComplEx	250	0.3759	0.4027
RotatE	250	0.4353	0.4353
PairRE	200	0.5423	0.5208
TripleRE	200	0.6045	0.5794
CompoundE	100	**0.6704**	**0.6515**

Table 3: Filtered ranking of link prediction on ogbl-wikikg2. Best results are in bold and best known published results are underlined.

WN18RR and ogbl-wikikg2 datasets, respectively. The best results are shown in bold fonts whereas the second best are underlined. CompoundE is a competitive model among embedding-based methods across all three datasets. As shown in Table 3, the results of CompoundE are much better than previous KGE models while the embedding dimension and the model parameter numbers are significantly lower for the ogbl-wikikg2 dataset. This implies lower computation and memory costs of CompoundE. We see from Table 2 that CompoundE has achieved significant improvement over distance-based KGE models using a single operation, either translation (TransE), rotation (RotatE), or scaling (PairRE). This confirms that cascading geometric transformations is an effective strategy for designing KG embeddings. In Table 2, we also compare CompoundE with text-based methods for link prediction on FB15k-237 and WN18RR datasets. Note that text-based methods require entity textual descriptions to make meaningful prediction. It is worth noting that CompoundE can still outperform transformer-based approaches significantly for WN18RR dataset, without having access to large amount of pretraining corpus.

Performance on Different Relation Types. To gain insights into the superior performance of CompoundE, we examine the performance of CompoundE on each type of relations. KG relations can be categorized into 4 types: 1) 1-to-1, 2) 1-to-N, 3) N-to-1, and 4) N-to-N. We classify the relations based on the following rule. For each relation, r, we compute the average number of subject (head) entities per object (tail) entity as hpt_r and the average number of object entities per subject as



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

Figure 2: Distribution of relation embedding values for "**friends**" relation in FB15k-237, obtained using $\|\mathbf{S_r} \cdot \mathbf{R_r} \cdot \mathbf{T_r} \cdot \mathbf{h} - \mathbf{t}\|$.

tph_r. Then, with a specific threshold η,

$$\begin{cases} hpt_r < \eta \text{ and } tph_r < \eta \implies r \text{ is 1-to-1} \\ hpt_r < \eta \text{ and } tph_r \geq \eta \implies r \text{ is 1-to-N} \\ hpt_r \geq \eta \text{ and } tph_r < \eta \implies r \text{ is N-to-1} \\ hpt_r \geq \eta \text{ and } tph_r \geq \eta \implies r \text{ is N-to-N.} \end{cases} \tag{20}$$

We set $\eta = 1.5$ as a logical threshold by following the convention. Table 4 compares the MRR scores of CompoundE with benchmarking models on 1-to-1, 1-to-N, N-to-1, and N-to-N relations in head and tail entities prediction performance for the FB15k237 dataset. We see that CompoundE consistently outperforms benchmarking models in all relation categories. The filtered MRR scores on each relation type of the WN18RR dataset are given in Table 5. We observe that CompoundE has a significant advantage over benchmarking models for certain 1-to-N relations such as "**member_of_domain_region**" ($+84.8\%$) and for some N-to-1 relations such as "**synset_domain_topic_of**" ($+12.7\%$). CompoundE is more effective than previous KGE models in modeling complex relations.

In Fig. 2, we visualize relation embedding for the "**friend**" relation in FB15k-237 by plotting the histogram of translation, rotation, and scaling parameter values. Since "**friend**" is a symmetric relation, we expect the translation value to be close to zero, which is consistent with Fig. 2 (a). Also, since "**friend**" is an N-to-N relation, we expect the Compound operation to be singular. Actually, most of the scaling values are zero as shown in Fig. 2 (c). They support our theoretical analysis of CompoundE's properties.

4.2 Path Query Answering

Path query is important since it is often desired to perform complex queries on knowledge graph. For example, one might ask "where did Michelle Obama's spouse live in?". To obtain the answer, a

Task	Predicting Head				Predicting Tail			
Type	**1-to-1**	**1-to-N**	**N-to-1**	**N-to-N**	**1-to-1**	**1-to-N**	**N-to-1**	**N-to-N**
TransE	0.492	0.454	0.081	0.252	0.485	0.072	0.740	0.367
RotatE	0.493	0.471	0.088	0.259	0.491	0.072	0.748	0.370
PairRE	0.496	0.476	0.117	0.274	0.492	0.073	0.763	0.387
CompoundE	**0.501**	**0.488**	**0.123**	**0.279**	**0.497**	**0.074**	**0.783**	**0.394**

Table 4: Filtered MRR on four relation types of FB15k-237.

Relation	Type	TransE	RotatE	Ours
similar_to	1-to-1	0.294	**1.000**	**1.000**
verb_group	1-to-1	0.363	0.961	**0.974**
member_meronym	1-to-N	0.179	**0.259**	0.254
has_part	1-to-N	0.117	**0.200**	**0.200**
member_of_domain_usage	1-to-N	0.113	0.297	**0.309**
member_of_domain_region	1-to-N	0.114	0.217	**0.401**
hypernym	N-to-1	0.059	0.156	**0.179**
instance_hypernym	N-to-1	0.289	0.322	**0.351**
synset_domain_topic_of	N-to-1	0.149	0.339	**0.382**
also_see	N-to-N	0.227	0.625	**0.629**
derivationally_related_form	N-to-N	0.440	**0.957**	0.956

Table 5: Filtered MRR on each relation type of WN18RR.

model first need to correctly predict the fact that (Michelle Obama, spouse, Barack Obama), and then predict (Barack Obama, livedIn, Chicago). CompoundE has the property to perform well on this task since it is capable of modeling the non-commutative relation compositions.

In Path Query Answering (PQA), a tuple (s, P, t) is given, where s and t denote the source and target entities and $P = \{r_1, \ldots, r_k\}$ denotes the relation path consisting of a sequence of relations that links $s \to r_1 \to r_2 \cdots \to r_k \to t$. PQA tests that after traversing through the relation path from a given source entity, whether the model is able to predict the correct target entity. During testing, the ground truth t is hidden and we compute the score for all candidate target entities and evaluate the quantile of ground truth, which is the fraction of irrelevant candidates that's ranked lower than the ground truth. Mean quantile of all test paths are reported. In particular, type match paths are excluded since those are trivial for prediction. Specifically, we use both the KG triples and sampled paths with length $|P| \in \{2, 3, 4, 5\}$ to train the embedding, which is also referred to as the "comp" setting (Guu et al., 2015). We use CompoundE to perform PQA on the Freebase and WordNet datasets prepared by (Guu et al., 2015). Statistics of these two datasets are shown in Table 10. Performance comparison with previous models on the PQA task under the "comp" setting is shown in Table 6. Results show that CompoundE is very competitive for the PQA task among pure embedding models.

	WordNet		Freebase	
	MQ	H@10	MQ	H@10
Bilinear	0.894	0.543	0.835	0.421
TransE	0.933	0.435	0.880	0.505
DistMult	0.904	0.311	0.848	0.386
RotatE	0.947	0.653	0.901	0.601
Rotate3D	0.949	0.671	0.905	0.621
CompoundE	**0.951**	**0.674**	**0.913**	**0.650**

Table 6: Performance comparison for path query answering.

4.3 KG Entity Typing

KG Entity typing predicts class labels for nodes in knowledge graph. Entity type provides semantic signals for information extraction tasks such as relation extraction (Yaghoobzadeh et al., 2017), entity linking (Gupta et al., 2017; Choi et al., 2018) and coreference resolution (Durrett and Klein, 2014). Entity typing is challenging since each entity may be associated with a large number of type labels. We show that CompoundE can also be effective for performing entity typing.

We perform entity typing using CompoundE embedding on the FB15k-ET and YAGO43k-ET dataset prepared by (Moon et al., 2017). Statistics of these datasets are shown in Table 9. In addition to RDF triples (h, r, t), entity and entity type pairs (e, t) are added to these entity typing datasets. Since the type can be interpreted as "isA" relationship between e and t, we add a "type" relation between the e and t pair and treat that as a special triple. Performance comparison with existing work is shown in Table 7. The optimal configuration is shown in Table 13. Similar to link prediction, we also report the MRR and Hits@k scores. Results show that CompoundE achieves significant improvement over other models especially for the YAGO43k-ET dataset, even without giving special

Datasets	FB15k-ET				YAGO43k-ET			
Metrics	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
TransE (Bordes et al., 2013)	0.45	31.51	51.45	73.93	0.21	12.63	23.24	38.93
TransE-ET (Moon et al., 2017)	0.46	33.56	52.96	71.16	0.18	9.19	19.41	35.58
ETE (Moon et al., 2017)	0.50	38.51	55.33	71.93	0.23	13.73	26.28	42.18
HMGCN (Jin et al., 2019)	0.51	39.02	54.75	72.36	0.25	14.21	27.34	43.69
ConnectE (Zhao et al., 2020)	0.59	49.55	64.32	79.92	0.28	16.01	30.85	47.92
CORE (Ge et al., 2022)	0.60	48.91	66.30	81.60	0.35	24.17	39.18	54.95
AttEt (Zhuo et al., 2022)	0.62	51.66	67.68	82.13	0.35	24.43	41.31	56.48
CompoundE	**0.64**	**52.49**	**71.88**	**85.89**	**0.48**	**36.36**	**55.80**	**70.31**

Table 7: Entity typing performance comparison for FB15k-ET and YAGO43k-ET datasets. Best result are in bold and second best result are underlined.

treatment to the representation of entity types. This observation supports the claim that CompoundE is strongly capable of representing entity semantics.

4.4 Complexity Analysis.

We compare the computational complexity of CompoundE and several popular KGE models in Table 8. The last column gives the estimated number of free parameters used by different models to achieve the best performance for the ogbl-wikikg2 dataset. CompoundE cuts the number of parameters at least by half while achieving much better performance. In the table, n, m, and d denote the entity number, the relation number, and their embedding dimension, respectively. Since $n \gg m$ in most datasets, we can afford to increase the complexity of relation embedding for better link prediction result without significantly increasing the overall space complexity. In Fig. 3, we compare the MRR scores of CompoundE and previous SOTA embedding models on the ogbl-wikikg2 dataset under different dimension settings $d \in \{10, 20, 50, 100, 150, 200, 250, 300\}$. CompoundE significantly outperforms benchmarking methods, even under low dimension setting.

Hyperparameters. We conduct two sets of controlled experiments to find the best model configurations for ogbl-wikikg2, FB15k-237, and WN18RR datasets. For the first set, we evaluate the effect of different combinations of learning rates and embedding dimensions while keeping other hyperparameters constant. For the second set, we evaluate the effect of different combinations of the training batch size and the negative sample size, while keeping other hyperparameters constant. The optimal model configurations for three datasets are given in Table 11 of the appendix.

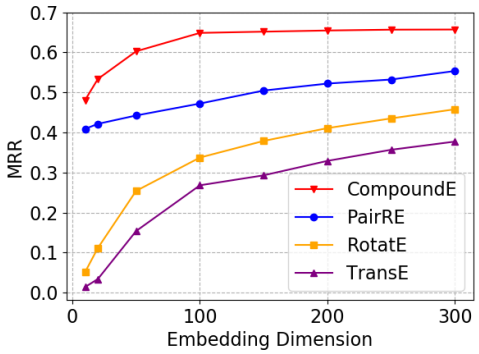


Figure 3: MRR scores on ogbl-wikikg2 dataset.

5 Conclusion and Future Work

A new KGE model called CompoundE was proposed in this work. We showed that quite a few distance-based KGE models are special cases of CompoundE. Extensive experiments were conducted for three different knowledge graph prediction tasks including link prediction, path query answering, and entity typing. Competitive experimental results demonstrate the effectiveness of CompoundE. We also mathematically prove the properties of CompoundE and its capability of modeling different relation patterns. We also explain the performance difference of different CompoundE forms, especially for the complex relation patterns.

We are interested in exploring two topics as future extensions. First, we may consider more complex operations in CompoundE. For example, there is a recent trend to extend 2D rotations to 3D rotations for rotation-based embeddings such as RotatE3D (Gao et al., 2020), SU2E (Yang et al., 2020). It is worthwhile to explore CompoundE3D. Second, CompoundE is expected to be useful in many

Model	Ent. emb.	Rel. emb.	Scoring Function	Space	# Params
TransE	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$O((m+n)d)$	1251M
ComplEx	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\text{Re}\left(\sum_{k=1}^{K} \mathbf{r}_k \mathbf{h}_k \bar{\mathbf{t}}_k\right)$	$O((m+n)d)$	1251M
RotatE	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$-\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|$	$O((m+n)d)$	1250M
PairRE	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}^{\mathbf{H}}, \mathbf{r}^{\mathbf{T}} \in \mathbb{R}^d$	$-\|\mathbf{h} \odot \mathbf{r}^{\mathbf{H}} - \mathbf{t} \odot \mathbf{r}^{\mathbf{T}}\|$	$O((m+n)d)$	500M
CompoundE-Head	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{T}[:, d-1], diag(\mathbf{S}) \in \mathbb{R}^d, \theta \in \mathbb{R}^{d/2}$	$-\|\mathbf{T} \cdot \mathbf{R}(\theta) \cdot \mathbf{S} \cdot \mathbf{h} - \mathbf{t}\|$	$O((m+n)d)$	250.1M
CompoundE-Tail	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\hat{\mathbf{T}}[:, d-1], diag(\hat{\mathbf{S}}) \in \mathbb{R}^d, \theta \in \mathbb{R}^{d/2}$	$-\left\|\mathbf{h} - \hat{\mathbf{T}} \cdot \hat{\mathbf{R}}(\theta) \cdot \hat{\mathbf{S}} \cdot \mathbf{t}\right\|$	$O((m+n)d)$	250.1M
CompoundE-Complete	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{T}/\hat{\mathbf{T}}[:, d-1], diag(\mathbf{S}/\hat{\mathbf{S}}) \in \mathbb{R}^d, \theta \in \mathbb{R}^{d/2}$	$-\left\|\mathbf{T} \cdot \mathbf{R}(\theta) \cdot \mathbf{S} \cdot \mathbf{h} - \hat{\mathbf{T}} \cdot \hat{\mathbf{R}}(\theta) \cdot \hat{\mathbf{S}} \cdot \mathbf{t}\right\|$	$O((m+n)d)$	250.3M

Table 8: Complexity comparison of KGE models.

downstream tasks. This conjecture has to be verified. If this is the case, CompoundE can offer a low memory solution to these tasks in realistic settings.

Limitations

Similar to many knowledge graph embedding models, our proposed method is yet to handle link prediction under inductive settings. One possible future extension is to leverage entity description information to generate textual features and use CompoundE as a decoder to handle unseen entities. Also, the affine operators we use are limited to translation, rotation, and scaling and this may limit the number of different relation patterns we can handle. In the future, we can include all affine transformations and investigate their difference. Also, because we use 2D givens rotation matrix, the embedding dimension setting needs to be a factor of 2. We can explore higher dimensional transformations such as 3D transformations and compare the modeling power.

Acknowledgment

The authors acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources that have contributed to the research results reported within this publication. URL: https://carc.usc.edu.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735. Springer.

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Multi-relational Poincaré graph embeddings. In *Adv. Neural Info. Process. Syst. 32 (NeurIPS 2019)*, volume 32.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. 2008 ACM SIGMOD Int. Conf. Manage. Data (SIGMOD '08)*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Adv. Neural Info. Process. Syst. 26 (NeurIPS 2013)*, page 2787–2795.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2022. Geometry interaction knowledge graph embeddings. In *Proc. 36th AAAI Conf. Artif. Intell.*

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proc. 24th AAAI Conf. Artif. Intell.*, page 1306–1313.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proc. 58th Annu. Meet. Assoc. Comput. Linguist. (ACL 2020)*, pages 6901–6914.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4360–4369, Online. Association for Computational Linguistics.

Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. 2022. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4005–4017.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the association for computational linguistics*, 2:477–490.

Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3D: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage. (CIKM'20)*, pages 385–394.

Xiou Ge, Yun-Cheng Wang, Bin Wang, and CC Jay Kuo. 2022. CORE: A knowledge graph entity type prediction method via complex space regression and embedding. *Pattern Recognit. Lett.*, 157:97–103.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *EMNLP*.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proc. 53rd Annu. Meet. Assoc. Comput. Linguist. (ACL 2015)*, volume 1, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proc. 30st AAAI Conf. Artif. Intell.*, pages 985–991.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2019. Fine-grained entity typing via hierarchical multi graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4969–4978, Hong Kong, China. Association for Computational Linguistics.

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

Dingcheng Li, Siamak Zamani, Jingyuan Zhang, and Ping Li. 2019. Integration of knowledge graph embedding into topic modeling with hierarchical Dirichlet process. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 940–950, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiayi Li and Yujiu Yang. 2022. STaR: Knowledge graph embedding by scaling, translation and rotation. In *Artificial Intelligence and Mobile Services–AIMS 2022: 11th International Conference, Held as Part of the Services Conference Federation, SCF 2022, Honolulu, HI, USA, December 10–14, 2022, Proceedings*, pages 31–45. Springer.

Zongwei Liang, Junan Yang, Hui Liu, and Keju Huang. 2021. A semantic filter based on relations for knowledge graph completion. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7920–7929.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proc. 29th AAAI Conf. Artif. Intell.*, pages 2181–2187.

George A Miller. 1995. WordNet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

Changsung Moon, others, Paul Jones, and Nagiza F Samatova. 2017. ru. In *CIKM*.

Mojtaba Nayyeri, Chengjin Xu, Franca Hoffmann, Mirza Mohtashim Alam, Jens Lehmann, and Sahar Vahdati. 2021. Knowledge graph representation learning using ordinary differential equations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9529–9548.

Guanglin Niu, Bo Li, Yongfei Zhang, and Shiliang Pu. 2022. CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2867–2877, Dublin, Ireland. Association for Computational Linguistics.

Yanhui Peng and Jing Zhang. 2020. LineaRE: Simple but powerful knowledge graph embedding for link prediction. In *Proc. 2020 IEEE 20th Int. Conf. Data Min. (ICDM '20)*, pages 422–431.

William K Pratt. 2013. *Introduction to digital image processing*. CRC press.

Steven M Seitz and Charles R Dyer. 1996. View morphing. In *Proc. 23rd Annu. Conf. Comput. Graphics Interactive Techniques (SIGGRAPH'96)*, pages 21–30.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proc. 31st AAAI Conf. Artif. Intell.*, page 4444–4451.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge. In *Proc. 16th Int. Conf. World Wide Web (WWW'07)*, 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. In *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, page 1–18.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proc. 33rd Int. Conf. Mach. Learn. (ICML 2016)*, pages 2071–2080.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledge base. *Commun. ACM*, 57(10):78–85.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proc. 28th AAAI Conf. Artif. Intell.*, pages 1112–1119.

George Wolberg. 1990. *Digital image warping*, volume 10662. IEEE Computer Society Press Los Alamitos, CA.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, pages 1–13.

Jinfa Yang, Xianghua Ying, Yongjie Shi, Xin Tong, Ruibin Wang, Taiyan Chen, and Bowei Xing. 2022. Knowledge graph embedding by adaptive limit scoring loss using dynamic weighting strategy. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1153–1163.

Tong Yang, Long Sha, and Pengyu Hong. 2020. Nage: Non-Abelian group embedding for knowledge graphs. In *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage. (CIKM'20)*, pages 1735–1742.

Qianjin Zhang, Ronggui Wang, Juan Yang, and Lixia Xue. 2022a. Knowledge graph embedding by reflection transformation. *Knowl.-Based Syst.*, 238:107861.

Yongqi Zhang, Zhanke Zhou, Quanming Yao, and Yong Li. 2022b. Efficient hyper-parameter search for knowledge graph embedding. In *ACL (long paper)*, pages 2715–2735.

Yu Zhao, Anxiang Zhang, Ruobing Xie, Kang Liu, and Xiaojie Wang. 2020. Connecting embeddings for knowledge graph entity typing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6419–6428, Online. Association for Computational Linguistics.

Jianhuan Zhuo, Qiannan Zhu, Yinliang Yue, Yuhong Zhao, and Weisi Han. 2022. A neighborhood-attention fine-grained entity typing for knowledge graph completion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1525–1533.

A Properties of CompoundE

Let $\mathbf{M}$ and $\hat{\mathbf{M}}$ denote the compound operation for the head and tail entity embeddings, respectively. In the following, we will prove nine properties of CompoundE.

Proposition A.1. *CompoundE can model 1-N relations.*

Proof. A relation r is an 1-N relation iff there exist at least two distinct tail entities t_1 and t_2 such that (h, r, t_1) and (h, r, t_2) both hold. Then we have:

$$\begin{aligned} \mathbf{M} \cdot \mathbf{h} = \hat{\mathbf{M}} \cdot \mathbf{t_1}, \quad \mathbf{M} \cdot \mathbf{h} = \hat{\mathbf{M}} \cdot \mathbf{t_2} \\ \hat{\mathbf{M}} \cdot (\mathbf{t_1} - \mathbf{t_2}) = 0 \end{aligned} \tag{21}$$

Since $\mathbf{t_1} \neq \mathbf{t_2}$, CompoundE can model 1-N relations when $\hat{\mathbf{M}}$ is singular. □

Proposition A.2. *CompoundE can model N-1 relations.*

Proof. A relation r is an N-1 relation iff there exist at least two distinct head entities h_1 and h_2 such that (h_1, r, t) and (h_1, r, t) both hold. Then we have:

$$\begin{aligned} \mathbf{M} \cdot \mathbf{h_1} = \hat{\mathbf{M}} \cdot \mathbf{t}, \quad \mathbf{M} \cdot \mathbf{h_2} = \hat{\mathbf{M}} \cdot \mathbf{t} \\ \mathbf{M} \cdot (\mathbf{h_1} - \mathbf{h_2}) = 0 \end{aligned} \tag{22}$$

Since $\mathbf{h_1} \neq \mathbf{h_2}$, CompoundE can model N-1 relations when $\mathbf{M}$ is singular. □

Proposition A.3. *CompoundE can model N-N relations.*

Proof. By the proof for Prop.A.1 and A.2, N-N relations can be modeled when both $\mathbf{M}$ and $\hat{\mathbf{M}}$ are singular. □

Proposition A.4. *CompoundE can model symmetric relations.*

Proof. A relation r is a symmetric relation iff (h, r, t) and (t, r, h) holds simultaneously. Then we have:

$$\begin{aligned} \mathbf{M}\cdot\mathbf{h} = \hat{\mathbf{M}}\cdot\mathbf{t} &\implies \mathbf{h} = \mathbf{M}^{-1}\hat{\mathbf{M}}\cdot\mathbf{t} \\ \mathbf{M}\cdot\mathbf{t} = \hat{\mathbf{M}}\cdot\mathbf{h} &\implies \mathbf{M}\cdot\mathbf{t} = \hat{\mathbf{M}}\mathbf{M}^{-1}\hat{\mathbf{M}}\cdot\mathbf{t} \\ \mathbf{M}\hat{\mathbf{M}}^{-1} &= \hat{\mathbf{M}}\mathbf{M}^{-1} \end{aligned} \tag{23}$$

Therefore, CompoundE can model symmetric relations when $\mathbf{M}\hat{\mathbf{M}}^{-1} = \hat{\mathbf{M}}\mathbf{M}^{-1}$. □

Proposition A.5. *CompoundE can model antisymmetric relations.*

Proof. A relation r is a antisymmetric relation iff (h, r, t) holds but (t, r, h) does not. By similar proof for Proposition A.4, CompoundE can model symmetric relations when $\mathbf{M}\hat{\mathbf{M}}^{-1} \neq \hat{\mathbf{M}}\mathbf{M}^{-1}$. □

Proposition A.6. *CompoundE can model inversion relations.*

Proof. A relation r_2 is the inverse of relation r_1 iff (h, r_1, t) and (t, r_2, h) holds simultaneously. Then we have:

$$\begin{aligned} \mathbf{M_1}\cdot\mathbf{h} = \hat{\mathbf{M}}_\mathbf{1}\cdot\mathbf{t} &\implies \mathbf{h} = \mathbf{M_1^{-1}}\hat{\mathbf{M}}_\mathbf{1}\cdot\mathbf{t} \\ \mathbf{M_2}\cdot\mathbf{t} = \hat{\mathbf{M}}_\mathbf{2}\cdot\mathbf{h} &\implies \mathbf{M_2}\cdot\mathbf{t} = \hat{\mathbf{M}}_\mathbf{2}\mathbf{M_1^{-1}}\hat{\mathbf{M}}_\mathbf{1}\cdot\mathbf{t} \\ \hat{\mathbf{M}}_\mathbf{2}^{-1}\mathbf{M_2} &= \mathbf{M_1^{-1}}\hat{\mathbf{M}}_\mathbf{1} \end{aligned} \tag{24}$$

Therefore, CompoundE can model inversion relations when $\hat{\mathbf{M}}_\mathbf{2}^{-1}\mathbf{M_2} = \mathbf{M_1^{-1}}\hat{\mathbf{M}}_\mathbf{1}$. □

Proposition A.7. *CompoundE can model relation compositions.*

Proof. r_3 is a composition of r_1 and r_2 iff (e_1, r_1, e_2), (e_2, r_2, e_3), and (e_1, r_3, e_3) hold simultaneously. Then we have:

$$\begin{aligned} \mathbf{M_1}\cdot\mathbf{e_1} = \hat{\mathbf{M}}_\mathbf{1}\cdot\mathbf{e_2} &\implies \mathbf{e_1} = \mathbf{M_1^{-1}}\hat{\mathbf{M}}_\mathbf{1}\cdot\mathbf{e_2} \\ \mathbf{M_2}\cdot\mathbf{e_2} = \hat{\mathbf{M}}_\mathbf{2}\cdot\mathbf{e_3} &\implies \mathbf{e_3} = \hat{\mathbf{M}}_\mathbf{2}^{-1}\mathbf{M_2}\cdot\mathbf{e_2} \\ \mathbf{M_3}\cdot\mathbf{e_1} &= \hat{\mathbf{M}}_\mathbf{3}\cdot\mathbf{e_3} \\ \mathbf{M_3}\mathbf{M_1^{-1}}\hat{\mathbf{M}}_\mathbf{1}\cdot\mathbf{e_2} &= \hat{\mathbf{M}}_\mathbf{3}\hat{\mathbf{M}}_\mathbf{2}^{-1}\mathbf{M_2}\cdot\mathbf{e_2} \\ \hat{\mathbf{M}}_\mathbf{3}^{-1}\mathbf{M_3} &= (\hat{\mathbf{M}}_\mathbf{2}^{-1}\mathbf{M_2})(\hat{\mathbf{M}}_\mathbf{1}^{-1}\mathbf{M_1}) \end{aligned} \tag{25}$$

Therefore, CompoundE can model relations composition when $\hat{\mathbf{M}}_\mathbf{3}^{-1}\mathbf{M_3} = (\hat{\mathbf{M}}_\mathbf{2}^{-1}\mathbf{M_2})(\hat{\mathbf{M}}_\mathbf{1}^{-1}\mathbf{M_1})$. □

Proposition A.8. *CompoundE can model both both commutative and non-commutative relations.*

Proof. Since the general form of affine group is non-commutative, our proposed CompoundE is non-commutative i.e.

$$(\mathbf{M_1}\hat{\mathbf{M}}_\mathbf{1}^{-1})(\mathbf{M_2}\hat{\mathbf{M}}_\mathbf{2}^{-1}) \neq (\mathbf{M_2}\hat{\mathbf{M}}_\mathbf{2}^{-1})(\mathbf{M_1}\hat{\mathbf{M}}_\mathbf{1}^{-1}) \tag{26}$$

where each $\mathbf{M}$ consists of translation, rotation, and scaling component. However, in special cases, when our relation embedding has only one of the translation, rotation, or scaling component, then the relation embedding becomes commutative again. □

Proposition A.9. *CompoundE can model transitive relations.*

Proof. r is a transitive relation iff (e_1, r, e_2), (e_2, r, e_3), and (e_1, r, e_3) hold simultaneously. Consider the CompoundE variant, and let $\mathbf{R} = \hat{\mathbf{R}}$, $\mathbf{S}$ be a idempotent matrix.

$$\begin{aligned} f_r(h,t) &= \|\mathbf{S}\cdot\mathbf{R}\cdot\mathbf{h} - \hat{\mathbf{R}}\cdot\mathbf{t}\| \\ &= \|\mathbf{R}\cdot(\mathbf{R^{-1}SR}\cdot\mathbf{h} - \mathbf{t})\| \\ &= \|\mathbf{R^{-1}SR}\cdot\mathbf{h} - \mathbf{t}\| \end{aligned} \tag{27}$$

Let $\mathbf{M}_r = \mathbf{R^{-1}SR}$. Then it is easy to see that

$$\begin{aligned} &\mathbf{M}_r\cdot\mathbf{M}_r\cdot\cdots\cdot\mathbf{M}_r \\ =&(\mathbf{R^{-1}SR})\cdot(\mathbf{R^{-1}SR})\cdot\cdots\cdot(\mathbf{R^{-1}SR}) \\ =&\mathbf{R^{-1}SR} \\ =&\mathbf{M}_r \end{aligned} \tag{28}$$

Therefore, CompoundE can model transitive relations. □

Proposition A.10. *CompoundE can model sub-relations.*

Proof. A relation r_1 is a sub-relation of r_2 if (h, r_2, t) implies (h, r_1, t). Without loss of generality, suppose our compounding operation takes the following form

$$\mathbf{M} = \mathbf{T}\cdot\mathbf{R}\cdot\mathbf{S},\ \hat{\mathbf{M}} = \hat{\mathbf{T}}\cdot\hat{\mathbf{R}}\cdot\hat{\mathbf{S}}, \tag{29}$$

and suppose

$$\begin{aligned} &\mathbf{T_1} = \mathbf{T_2}, \hat{\mathbf{T}}_\mathbf{1} = \hat{\mathbf{T}}_\mathbf{2}, \\ &\mathbf{R_1} = \mathbf{R_2}, \hat{\mathbf{R}}_\mathbf{1} = \hat{\mathbf{R}}_\mathbf{2}, \\ &\mathbf{S_1} = \gamma\mathbf{S_2}, \hat{\mathbf{S}}_\mathbf{1} = \gamma\hat{\mathbf{S}}_\mathbf{2}, \gamma \leq 1. \end{aligned} \tag{30}$$

With these conditions, we can compare the CompoundE scores generated by (h, r_1, t) and (h, r_2, t) as follows:

$$\begin{aligned}
&f_{r_1}(h,t) - f_{r_2}(h,t) \\
=&\|\mathbf{T_1}\cdot\mathbf{R_1}\cdot\mathbf{S_1}\cdot\mathbf{h} - \hat{\mathbf{T}}_\mathbf{1}\cdot\hat{\mathbf{R}}_\mathbf{1}\cdot\hat{\mathbf{S}}_\mathbf{1}\cdot\mathbf{t}\| - \\
&\|\mathbf{T_2}\cdot\mathbf{R_2}\cdot\mathbf{S_2}\cdot\mathbf{h} - \hat{\mathbf{T}}_\mathbf{2}\cdot\hat{\mathbf{R}}_\mathbf{2}\cdot\hat{\mathbf{S}}_\mathbf{2}\cdot\mathbf{t}\| \\
=&\|\mathbf{T_2}\cdot\mathbf{R_2}\cdot(\gamma\mathbf{S_2})\cdot\mathbf{h} - \hat{\mathbf{T}}_\mathbf{2}\cdot\hat{\mathbf{R}}_\mathbf{2}\cdot(\gamma\hat{\mathbf{S}}_\mathbf{2})\cdot\mathbf{t}\| - \\
&\|\mathbf{T_2}\cdot\mathbf{R_2}\cdot\mathbf{S_2}\cdot\mathbf{h} - \hat{\mathbf{T}}_\mathbf{2}\cdot\hat{\mathbf{R}}_\mathbf{2}\cdot\hat{\mathbf{S}}_\mathbf{2}\cdot\mathbf{t}\| \\
=&\|\gamma(\mathbf{T_2}\cdot\mathbf{R_2}\cdot\mathbf{S_2}\cdot\mathbf{h} - \hat{\mathbf{T}}_\mathbf{2}\cdot\hat{\mathbf{R}}_\mathbf{2}\cdot\hat{\mathbf{S}}_\mathbf{2}\cdot\mathbf{t})\| - \\
&\|\mathbf{T_2}\cdot\mathbf{R_2}\cdot\mathbf{S_2}\cdot\mathbf{h} - \hat{\mathbf{T}}_\mathbf{2}\cdot\hat{\mathbf{R}}_\mathbf{2}\cdot\hat{\mathbf{S}}_\mathbf{2}\cdot\mathbf{t}\| \leq 0
\end{aligned} \tag{31}$$

This means that (h, r_1, t) generates a smaller error score than (h, r_2, t). If (h, r_2, t) holds, (h, r_1, t) must also holds. Therefore, r_1 is a sub-relation of r_2. □

B Performance comparison for different variations of CompoundE

We investigate the performance difference of CompoundE variants. Specifically, the different forms of CompoundE have visible difference in different relation types. We conduct experiment on YAGO3-10 dataset and compare the performance of CompoundE-left, CompoundE-right, CompoundE-Complete for 1-to-1, 1-to-N, and N-to-1 relations. In particular, when evaluating the 1-to-N relations, we focus on predicting $(?, r, t)$ while for N-to-1 relations we focus on predicting $(h, r, ?)$ to correctly reflect the performance on respective relation types. Performance comparison is shown in 4. We observe that for CompoundE-Complete has advantage over other forms for 1-to-1 relations. CompoundE-left and CompoundE-right are the better performing forms for 1-to-N and N-to-1 relations respectively. This observation is consistent with the discussion of the modeling capability of CompoundE. It still remains a questions that how different order of operator composition will affect the performance of CompoundE and we will address that in future work.

C Visualization of Embedding

We provide a 2D t-SNE visualization of the entity embedding generated by CompoundE for FB15k-237 in Fig. 5. Each entity is colored with its respective entity type. As shown in the figure, some entity type class are well separated while others are not. This scatter plot shows that entity representations extracted by CompoundE can capture the semantics of the entity. Thus, their embeddings can be used in various downstream tasks such as KG entity typing and similarity based recommendations.

Besides the histograms shown in the main paper, we add more plots to visualize CompoundE relation embedding values. In Fig. 6, we show the embedding values for the "friends" relation in the FB15k-237. We use the CompoundE-Complete variant ($\|\mathbf{S_r}\cdot\mathbf{R_r}\cdot\mathbf{T_r}\cdot\mathbf{h} - \hat{\mathbf{S}}_\mathbf{r}\cdot\hat{\mathbf{R}}_\mathbf{r}\cdot\hat{\mathbf{T}}_\mathbf{r}\cdot\mathbf{t}\|$) to generate the embedding. We plot the translation and scaling components for both the head and the tail. We only show a single plot for the rotation component since the rotation parameter is shared between the head and the tail. Different from the CompoundE-head ($\|\mathbf{S_r}\cdot\mathbf{R_r}\cdot\mathbf{T_r}\cdot\mathbf{h} - \mathbf{t}\|$), we see two modes (instead of only one mode) in CompoundE-Complete's plots. One conjecture for this difference is that CompoundE-Complete has a pair of operations on both the head and the tail, the distribution of values need to have two modes to maintain the symmetry. Similar to CompoundE-head, the scaling parameters of CompoundE-Complete have a large amount of zeros to maintain the singularity of compounding operators and help learn the N-to-N complex relations.

Fig. 7 and Fig. 8 display the histogram of relation embeddings for "**instance_hypernym**" relation and "**similar_to**" relation in WN18RR, respectively. The real (in blue) and the imaginary (in orange) parts are overlaid in each plot. Notice that "**instance_hypernym**" is an antisymmetric relation while "**similar_to**" is a symmetric relation. This relation pattern is reflected on the embedding histogram since the translation and the scaling histograms for the head and the tail are different in "**instance_hypernym**". In contrast, the translation and scaling histograms for the head and the tail are almost identical in "**similar_to**".

D Datasets

The path query dataset can be obtained from the link[2]. The original github repo[3] has the MIT license. The entity typing dataset can be found here[4]. Statistics of these datasets are shown in Table 9 and

[2] https://worksheets.codalab.org/worksheets/0xfcace41fdeec45f3bc6ddf31107b829f

[3] https://github.com/kelvinguu/traversing-knowledge-graphs

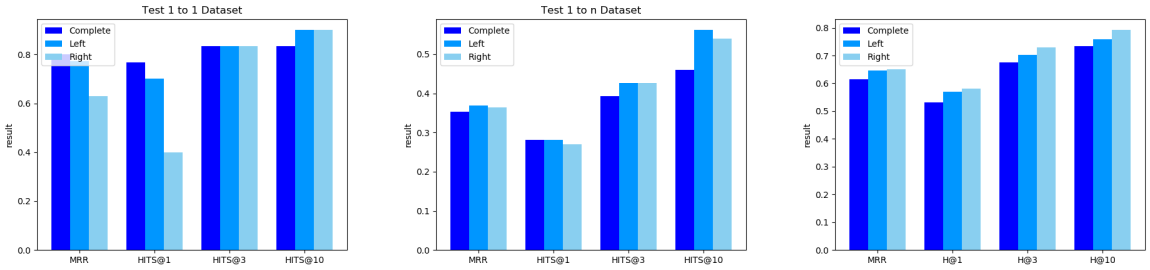
[4] https://github.com/cmoon2/knowledge_graph/tree/master/datasets

Dataset	#Ent	#Rel	#Type	#KG Triples			#Entity Type Pairs		
				#Train	#Valid	#Test	#Train	#Valid	#Test
FB15k-ET	14,951	1,345	3,851	483,142	50,000	59,071	136,618	15,749	15,780
YAGO43k-ET	42,335	37	45,182	331,687	29,599	29,593	375,853	42,739	42,750

Table 9: Entity Typing Datasets Statistics.

Dataset	#Ent	#Rel	#KG Triples			#Path		
			#Train	#Valid	#Test	#Train	#Valid	#Test
WordNet	38,551	11	110,361	2,602	10,462	2,129,539	11,277	46,577
Freebase	75,043	13	316,232	5,908	23,733	6,266,058	27,163	109,557

Table 10: Path Query Answering Datasets Statistics.



(a) 1 to 1 (b) 1 to n (c) n to 1

Figure 4: Comparing the performance of different CompoundE forms.

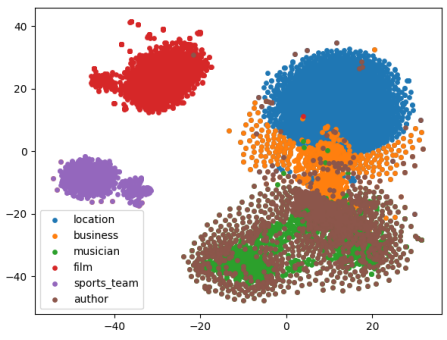


Figure 5: t-SNE visualization of entity embedding in the 2D space for some major entity types in FB15K-237.

Table 10.

E Implementation and Optimal Configurations

To form relation specific transformation matrices for high dimensional entity vectors, we can first cascade translation, rotation, and scaling operator to yield a compound operator $\mathbf{O_{r,i}} = \mathbf{T_{r,i}} \cdot \mathbf{R_{r,i}} \cdot \mathbf{S_{r,i}}$ In the actual implementation, we use the operator's representation in regular Cartesian coordinate instead of the homogeneous coordinate. Furthermore, a high-dimensional relation operator can be represented as a block diagonal matrix in the form of

$$\mathbf{M_r} = \mathbf{diag}(\mathbf{O_{r,1}}, \mathbf{O_{r,2}}, \ldots, \mathbf{O_{r,n}}), \qquad (32)$$

where $\mathbf{O_{r,i}} \in \mathbb{R}^{2\times 2}$ is the compound operator of the i-th block.

In our implementation, we normalize all entity embeddings to unit vectors before applying compound operations. The optimal configurations of CompoundE are given in Table 11. The implementation of the rotation operation in the optimal CompoundE configuration for the WN18RR dataset is adapted from RotatE.

All experiments were conducted on a NVIDIA V100 GPU with 32GB memory. GPUs with larger memory such as NVIDIA A100 (40GB), NVIDIA A40 (48GB) are only needed for hyperparameter sweep when the dimension, the negative sample size, and the batch size are high. We should point out that such settings are not essential for CompoundE to obtain good results. They were used to search for the optimal configurations. We have considered the following set of numbers as our parameter search space to obtain the best performance we can for each dataset and tasks.

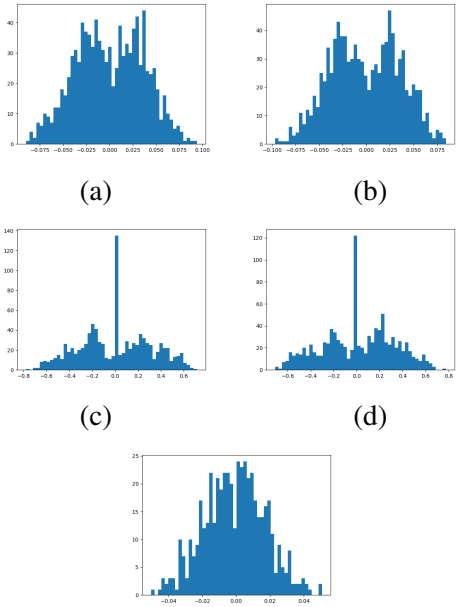
Link Prediction.

Table 11: Optimal Configurations for Link Prediction Tasks. B denotes the batch size and N denotes the negative sample size.

Dataset	CompoundE Variant	#Dim	lr	B	N	α	ζ
ogbl-wikikg2	$\|\mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{T}} \cdot \hat{\mathbf{R}} \cdot \mathbf{t}\|$	100	0.005	4096	250	1	7
FB15k-237	$\|\mathbf{S} \cdot \mathbf{R} \cdot \mathbf{T} \cdot \mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{R}} \cdot \hat{\mathbf{T}} \cdot \mathbf{t}\|$	600	0.00005	1024	125	1	6
WN18RR	$\|\mathbf{R} \cdot \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{T}} \cdot \mathbf{t}\|$	500	0.00007	1024	256	0.5	6

Table 12: Optimal Configurations for Path Query Answering. B denotes the batch size and N denotes the negative sample size.

Dataset	CompoundE Variant	#Dim	lr	B	N	α	ζ
Freebase	$\|\mathbf{S} \cdot \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{h} - \mathbf{t}\|$	1500	0.00002	1024	256	1	6
WordNet	$\|\mathbf{S} \cdot \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{h} - \mathbf{t}\|$	1000	0.00005	1024	256	1	6



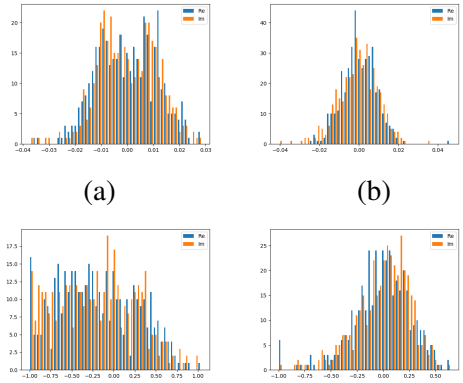
(a) (b) (c) (d) (e)

Figure 6: FB15k-237 "**friends**" relation embedding obtained using $\|\mathbf{S} \cdot \mathbf{R} \cdot \mathbf{T} \cdot \mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{R}} \cdot \hat{\mathbf{T}} \cdot \mathbf{t}\|$: (a) distribution of head translation values, (b) distribution of tail translation values, (c) distribution of head scaling values, (d) distribution of tail scaling values, and (e) distribution of rotation angle values.

Wikikg2

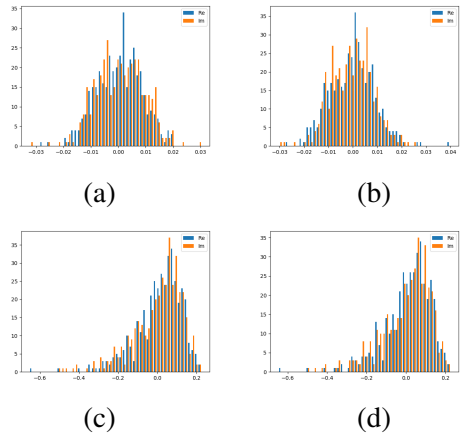
- $d \in \{50, 100, 150, 200, 250, 300, 400\}$
- $lr \in \{0.0005, 0.001, 0.005, 0.01\}$
- $\zeta \in \{5, 6, 7, 8, 9\}$
- batch $\in \{256, 512, 1024, 2048\}$
- negative sample $\in \{256, 512, 1024, 2048\}$

FB15k-237



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

Figure 7: WN18RR "**instance_hypernym**" relation: (a) distribution of head translation values, (b) distribution of tail translation values, (c) distribution of head scaling values, and (d) distribution of tail scaling values.



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

Figure 8: WN18RR "**similar_to**" relation: (a) distribution of head translation values, (b) distribution of tail translation values, (c) distribution of head scaling values, and (d) distribution of tail scaling values.

Table 13: Optimal Configurations for Entity Typing. *B* denotes the batch size and *N* denotes the negative sample size.

Dataset	CompoundE Variant	#Dim	*lr*	*B*	*N*	α	ζ
FB15k-ET	$\|\mathbf{R}\cdot\mathbf{T}\cdot\mathbf{S}\cdot\mathbf{h}-\hat{\mathbf{R}}\cdot\hat{\mathbf{T}}\cdot\hat{\mathbf{S}}\cdot\mathbf{t}\|$	1500	0.00005	2048	512	1	10
YAGO43k-ET	$\|\mathbf{h}-\hat{\mathbf{T}}\cdot\hat{\mathbf{S}}\cdot\hat{\mathbf{R}}\cdot\mathbf{t}\|$	1000	0.00005	1024	256	1	6

- $d \in \{100, 200, 300, 400\}$
- $lr \in \{0.00001, 0.00005, 0.0001, 0.0005\}$
- $\zeta \in \{4, 5, 6, 7, 8, 9\}$
- batch $\in \{256, 512, 1024, 2048\}$
- negative sample $\in \{256, 512, 1024, 2048\}$

WN18RR

- $d \in \{100, 200, 300, 400\}$
- $lr \in \{0.00001, 0.00005, 0.0001, 0.0005\}$
- $\zeta \in \{5, 6, 7, 8, 9\}$
- batch $\in \{256, 512, 1024, 2048\}$
- negative sample $\in \{256, 512, 1024, 2048\}$

Path Query Answering.

Freebase

- $d \in \{500, 1000, 1500, 2000\}$
- $lr \in \{0.00001, 0.00002, 0.00005, 0.0001\}$
- $\zeta \in \{6, 9, 12, 15\}$
- batch $\in \{512, 1024\}$
- negative sample $\in \{256, 512\}$

WordNet

- $d \in \{500, 1000, 1500, 2000\}$
- $lr \in \{0.00001, 0.00002, 0.00005, 0.0001\}$
- $\zeta \in \{6, 9, 12, 15\}$
- batch $\in \{512, 1024\}$
- negative sample $\in \{256, 512\}$

Entity typing.

FB15k-ET

- $d \in \{500, 1000, 1500\}$
- $lr \in \{0.00001, 0.00005, 0.0001, 0.0005\}$
- $\zeta \in \{8, 9, 10, 11\}$
- batch $\in \{1024, 2048, 4096, 8192\}$
- negative sample $\in \{256, 512, 1024, 2048\}$

YAGO43k-ET

- $d \in \{500, 1000, 1500\}$
- $lr \in \{0.00001, 0.00005, 0.0001, 0.0005\}$
- $\zeta \in \{19, 20, 21, 22\}$
- batch $\in \{1024, 2048\}$
- negative sample $\in \{256, 512\}$

F Comparing CompoundE and STaR

The main difference between CompoundE and STaR is that STaR embedding uses a bilinear product and adopts a semantic matching approach while CompoundE's scoring function is a distance-based metric. Because of this, the optimization strategy for CompoundE is the self-adversrial negative sampling loss whereas STaR uses the regularized cross-entropy loss. More importantly, CompoundE embedding has clear and intuitive geometric interpretations whereas the design of STaR is less intuitive since it is unclear what composition of operators means in the context of a bilinear product. We also shed light on the superior capability of CompoundE to model relation compositions and entity semantics through PQA and entity typing experiments. Lastly, we can incorporate reflection and shear operators below who also belong to the affine operator family. The reflection matrix can be defined as

$$\mathbf{F} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ \sin(\phi) & -\cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{33}$$

And the shear matrices on two different directions can be defined as

$$\mathbf{H}_x = \begin{bmatrix} 1 & \tan(\psi_x) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{34}$$

Datasets	ogbl-wikikg2				
Metrics	Dim	Valid	Test		
		MRR	MRR	Hit@1	Hit@3	Hit@10
$\|\mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{T}} \cdot \hat{\mathbf{R}} \cdot \mathbf{t}\|$	100	0.6704	0.6515	0.5843	0.6781	0.7872
$\|\mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{T}} \cdot \hat{\mathbf{F}} \cdot \hat{\mathbf{R}} \cdot \mathbf{t}\|$	100	0.6694	0.6509	0.5844	0.6760	0.7865
$\|\mathbf{h} - \hat{\mathbf{S}} \cdot \hat{\mathbf{H}}_{\mathbf{x}} \cdot \hat{\mathbf{H}}_{\mathbf{y}} \cdot \hat{\mathbf{T}} \cdot \hat{\mathbf{R}} \cdot \mathbf{t}\|$	100	0.6701	0.6539	0.5865	0.6805	0.7906

Table 14: Preliminary comparison after adding reflection and shear operators.

$$\mathbf{H}_y = \begin{bmatrix} 1 & 0 & 0 \\ \tan(\psi_y) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{35}$$

We have done preliminary experiments on Wikikg2 but reflection and shear operators and result is shown in Table 14. We will further improve the result in the future.