

PairRE: Knowledge Graph Embeddings via Paired Relation Vectors

Linlin Chao
AntGroup
chulin.ell@antgroup.com

Taifeng Wang
AntGroup
taifeng.wang@alibaba-inc.com

Jianshan He
AntGroup
yebai.hjs@antgroup.com

Wei Chu
AntGroup
wei.chu@alibaba-inc.com

ABSTRACT

Distance based knowledge graph embedding methods show promising results on link prediction task, on which two topics have been widely studied: one is the ability to handle complex relations, such as N-to-1, 1-to-N and N-to-N, the other is to encode various relation patterns, such as symmetry/antisymmetry. However, the existing methods fail to solve these two problems at the same time, which leads to unsatisfactory results. To mitigate this problem, we propose PairRE, a model with improved expressiveness and low computational requirement. PairRE represents each relation with paired vectors, where these paired vectors project connected two entities to relation specific locations. Beyond its ability to solve the aforementioned two problems, PairRE is advantageous to represent subrelation as it can capture both the similarities and differences of subrelations effectively. Given simple constraints on relation representations, PairRE can be the first model that is capable of encoding symmetry/antisymmetry, inverse, composition and subrelation relations. Experiments on link prediction benchmarks show PairRE can achieve either state-of-the-art or highly competitive performances. In addition, PairRE has shown encouraging results for encoding subrelation.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Knowledge graph embedding, logic rules

1 INTRODUCTION

Knowledge graphs store huge amounts of structured data as the form of triples, with projects such as WordNet [19], Freebase [2], YAGO [25] and DBpedia [17]. They have gained widespread traction from their successful use in tasks such as question answering [5], semantic parsing [1], named entity disambiguation [38], recommender systems [36] and so on.

Since most knowledge graphs often suffer from incompleteness, completing knowledge graphs and predicting missing links between entities have been a fundamental problem. This problem is named as link prediction or knowledge graph completion. Knowledge graph embedding methods, in which embed every entities and relations into a low dimensional space, have been proposed for this problem. The key goal of knowledge graph embedding is to preserve the

semantic information for the massive triples. Meanwhile, these methods should be efficient to scale up to the size of modern day knowledge graphs.

Distance based embedding methods from TransE [3] to the recent state-of-the-art RotatE [26] have shown substantial improvements on link prediction task. Two major problems have been widely studied. The first one refers to handling of 1-to-N, N-to-1, and N-to-N complex relations [3, 18]. In case of the 1-to-N relations, given triples like (*StevenSpielberg*, *DirectorOf*, ?), distance based models should make all the corresponding entities like *Jaws*, *JurassicPark* and *Schindler'sList*, which are all films directed by *StevenSpielberg*, have closer distance to entity *StevenSpielberg* after transformation via relation *DirectorOf*. The difficulty is that all these entities should have different representations. Same issue happens in the cases of 1-to-N and N-to-1 relations. The latter is learning and inferring relation patterns according to observed triples, as the success of knowledge graph completion heavily relies on this ability [3, 26]. There are various types of relation patterns: symmetry (e.g., *IsSimilarTo*), antisymmetry (e.g., *FatherOf*), inverse (e.g., *PeopleBornHere* and *PlaceOfBirth*), composition (e.g., my mother's father is my grandpa) and so on. Previous methods solve these two problems separately. TransH [33], TransR [18], TransD [13] all focus on ways to solve complex relations. However, these methods can only encode antisymmetry relations. The recent state-of-the-art RotatE shows promising results to encode symmetry/antisymmetry, inverse and composition relations. However, complex relations still remain challenging to predict.

Except the well solved symmetry/antisymmetry, inverse and composition relation patterns, subrelation is another important one. For example, the subrelation pair, *PlaceOfBirth* and *LivedIn*. Given America is the birth place of Donald Trump, we can conclude that Donald Trump once lived in America. How to represent subrelation is a challenge problem. As knowledge graphs are often incomplete, the subrelation pairs always have overlap in connected entities. When the overlap ratio is large enough, these embedded subrelations may become undistinguishable. This problem is similar to modeling of 1-to-N/N-to-1/N-to-N modeling problem, where the difficult is to distinguish relations rather than entities. Previous methods like TransE and RotatE, which take relation embedding as a single vector, pay less attention to subrelation. These models are not expressive enough to capture both the differences and similarities of subrelation pairs, just as they struggle to predict 1-to-N complex relations. Encoding subrelation pattern for knowledge graph embedding methods is also a challenge problem. As far as

Method	Score Function	Performance of complex relations	Relation Patterns				
			<i>Sym</i>	<i>Asym</i>	<i>Inv</i>	<i>Comp</i>	<i>Sub</i>
TransE	$- \mathbf{h} + \mathbf{r} - \mathbf{t} $	Low	✗	✓	✓	✓	✗
TransR	$- \mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t} $	High	✗	✓	✗	✗	✗
RotatE	$- \mathbf{h} \circ \mathbf{r} - \mathbf{t} $	Low	✓	✓	✓	✓	✗
PairRE	$- \mathbf{h} \circ \mathbf{r}^h - \mathbf{t} \circ \mathbf{r}^t $	High	✓	✓	✓	✓	✓*

Table 1: Comparison between PairRE and some representative distance based knowledge graph embedding methods. *Sym*, *Asym*, *Inv*, *Comp* and *Sub* are the abbreviations for symmetry, antisymmetry, inverse and subrelation respectively. ✓* means the model can build the specific capacity with some constraints.

we know, two ComplEx based models, ComplEx-NNE-AER [7] and SimpleE⁺ [8], give the sufficient conditions to encode subrelation. Although subrelations can be encoded, there are still flaws in encoding other relation patterns, e.g., composition. How to encode these four relation patterns simultaneously is still a challenge problem.

Here we present PairRE, an embedding method that is capable of encoding complex relations and multiple relation patterns. Different from homogeneous graphs, knowledge graphs usually contain rich relations. The expressiveness of relation representation is one of the key components to model performance improvement. Meanwhile, retaining a low complexity is necessary to scale up to the size of modern datasets. Considering the trade-off between expressiveness and complexity, we use two vectors for relation representation. These vectors project the corresponding head and tail entities to Euclidean space and the distance between the projected vectors is then minimized. This provides three important benefits: (1) projecting head and tail entities to relation specific locations overcomes the problem of modeling 1-to-N, N-to-1 and N-to-N complex relations; (2) the interactions among the paired relation vectors enable the model to encode three important relation patterns, symmetry/antisymmetry, inverse and composition; (3) by adding simple constraints on relation representations, the model can encode subrelation further. Besides, PairRE is a highly efficient model, which contributes to large scale datasets.

We evaluate PairRE on six standard knowledge graph benchmarks and show:

- PairRE can achieve either state-of-the-art or highly competitive performance;
- PairRE can better handle complex relations and has the ability to encode symmetry/antisymmetry, inverse and composition relations;
- By adding simple constraints on relation vectors, PairRE can encode subrelation further.

2 BACKGROUND AND NOTATION

Given a knowledge graph that is represented as a list of fact triples, knowledge graph embedding methods define scoring function to measure the plausibility of these triples. We denote a triple by (h, r, t) , where h represents head entity, r represents relation and t represents tail entity. The column vectors of entities and relations are represented by bold lower case letters. The scoring function is denoted as $f_r(\mathbf{h}, \mathbf{t})$. The entities and relations belong to the set \mathcal{E}

and \mathcal{R} respectively. We denote the set of all triples that are true in a world as \mathcal{T} .

The definition of complex relations [33] is defined as follows. For each relation r , we compute the average number of tails per head (tphr) and the average number of heads per tail (hptr). If tphr < 1.5 and hptr < 1.5, r is treated as 1-to-1; if tphr > 1.5 and hptr > 1.5, r is treated as a N-to-N; if tphr < 1.5 and hptr > 1.5, r is treated as 1-to-N.

In this paper, we focus on four important relation patterns, which includes: (1) **Symmetry/antisymmetry**. A relation r is symmetry if $\forall e_1, e_2 \in \mathcal{E}, (e_1, r, e_2) \in \mathcal{T} \iff (e_2, r, e_1) \in \mathcal{T}$ and is antisymmetry if $(e_1, r, e_2) \in \mathcal{T} \Rightarrow (e_2, r, e_1) \notin \mathcal{T}$; (2) **Inverse**. If $\forall e_1, e_2 \in \mathcal{E}, (e_1, r_1, e_2) \in \mathcal{T} \iff (e_2, r_2, e_1) \in \mathcal{T}$, then relation r_1 and r_2 are inverse relations; (3) **Composition**. If $\forall e_1, e_2, e_3 \in \mathcal{E}, (e_1, r_1, e_2) \in \mathcal{T} \wedge (e_2, r_2, e_3) \in \mathcal{T} \Rightarrow (e_1, r_3, e_3) \in \mathcal{T}$, then relation r_3 can be seen as the composition of relation r_1 and r_2 ; (4) **Sub-relation** [23]. If $\forall e_1, e_2 \in \mathcal{E}, (e_1, r_1, e_2) \in \mathcal{T} \Rightarrow (e_1, r_2, e_2) \in \mathcal{T}$, then relation r_2 can be seen as a subrelation of relation r_1 .

3 RELATED WORK

Distance based models. Distance based models measure plausibility of fact triples as distance between entities. TransE, the most representative one, interprets relation as a translation vector r so that entities can be connected, i.e., $h + r \approx t$. TransE efficient, though cannot model symmetry relations and remains difficult in modeling complex relations. Several models are proposed for improving TransE to deal with complex relations, including TransH, TransR, TransD, TransSparse [14] and so on. All these methods project the entities to relation specific hyperplanes or spaces first, then translate these entities with relation vectors. By projecting entities to different spaces or hyperplanes, the ability to handle complex relations is improved. However, with the added projecting parameters, these models are unable ability to encode inverse and composition relations.

The recent state-of-the-art, RotatE, which can encode symmetry/antisymmetry, inverse and composition relation patterns, utilizes rotation based translational method in a complex space. Although expressiveness for different relation patterns, complex relations still remain challenging. GC-OTE [27] proposes to improve complex relation modeling ability of RotatE by introducing graph context for entity embedding. However, the calculation of graph contexts for head and tail entities is time consuming, which is almost infeasible for large scale knowledge graphs, e.g. ogbl-wikikg [12].

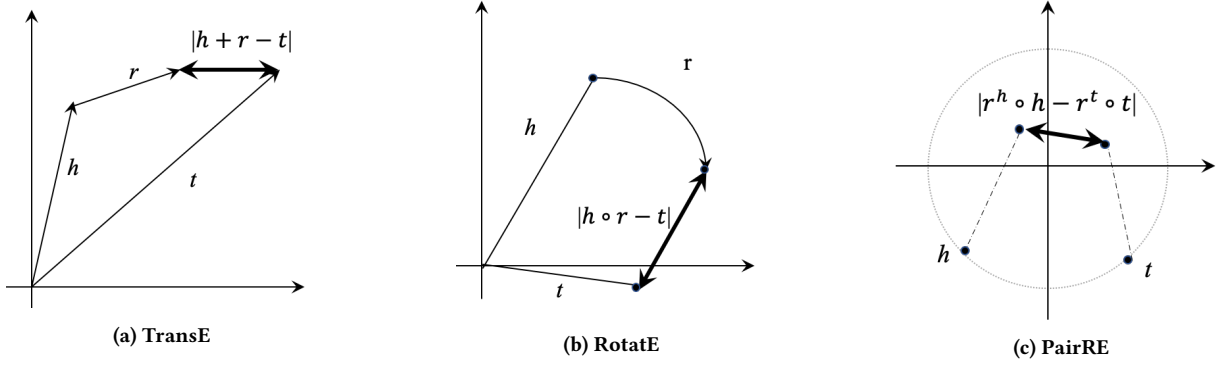


Figure 1: Illustration of TransE, RotatE and PairRE when the entities stay in a plane. For PairRE, all entities are on the unit circle. The relation vectors project entities to different locations.

Another related work is SE [4], which utilizes two separate relation matrixes to project head and tail entities. As pointed out by [26], this model is not able to encode symmetry/antisymmetry, inverse and composition relations.

Table 1 shows comparison between our method and some representative distance based knowledge graph embedding methods. As the table shows, our model is the most expressive one, with the ability to handle complex relations and encode four key relation patterns.

Semantic matching models. Semantic matching models exploit similarity-based scoring functions, which can be divided into bilinear models and neural network based models. As the models have been developed, such as RESCAL [22], DistMult [35], HolE [21], ComplEx [29] and QuatE [37], the key relation encoding abilities are enriched. However, all these models have the flaw in encoding composition relations [26].

RESCAL, ComplEx and Simple [16] are all proved to be fully expressive when embedding dimensions fulfill some requirements [16, 29, 32]. The fully expressiveness means these models can express all the ground truth existed in the data, including complex relations. However, these requirements are hardly fulfilled in practical use. It is proved by [32] that, to achieve complete expressiveness, the embedding dimension should larger than $N/32$, where N is the number of entities in dataset.

Neural networks based methods, e.g., convolution neural networks [6], graph convolutional networks [24] show promising performances. However, they are difficult to analyze as they work as a black box.

Encoding Subrelation. Existing methods encode subrelations mainly by utilizing first order logic rules. One way is to augment knowledge graphs with grounding of the rules, including subrelation rules. RUGE [10] and pLogictNet [23] are two representative works. These methods are not efficient or able to guarantee the subrelations in completed knowledge graph. The other way is adding constraints on entity and relation representations, e.g., ComplEx-NNE-AER and Simple⁺. These methods enrich the expressiveness of knowledge graph methods with relatively low cost. In this paper, we show that PairRE can encode subrelation with constraints

on relation representations while keeping the ability to encode symmetry/antisymmetry, inverse and composition relations.

4 METHODOLOGY

To overcome the problem of modeling 1-to-N/N-to-1/N-to-N complex relations and enrich the capabilities for different relation patterns, we propose a model with paired vectors for each relation. Given a training triple (h, r, t) composed of two entities $h, t \in \mathcal{E}$ and a relation $r \in \mathcal{R}$, our model learns vector embeddings of entities and relation in real space. Specially, PairRE takes relation embedding as paired vectors, which is represented as $[r^h, r^t]$. r^h and r^t project head entity h and tail entity t to Euclidean space respectively. The projecting operation is the Hadamard product¹ between these two vectors. PairRE then computes distance of the two projected vectors as plausibility of the triple. We want that $h \circ r^h \approx t \circ r^t$ when (h, r, t) holds, while $h \circ r^h$ should be far away from $t \circ r^t$ otherwise. In this paper, we take the L2-norm to measure the distance. We also add additional constraints on entity embeddings, that the L2-norm of h, t are 1. The scoring function is defined as follows:

$$f_r(h, t) = -||h \circ r^h - t \circ r^t||^2, \quad (1)$$

where $h, r^h, r^t, t \in \mathbb{R}^d$ and $||h||^2 = ||t||^2 = 1$. The model parameters are, all the entities' embeddings, $\{e_j\}_{j=1}^{\mathcal{E}}$ and all the relations' embeddings, $\{r_j\}_{j=1}^{\mathcal{R}}$.

Illustration of the proposed PairRE is shown in Figure 1. Compared to TransE/RotatE, PairRE enables an entity to have distributed representations when involved in different relations. By introducing the mechanism of projecting entities to relation specific locations, the model can alleviate the modeling problem for complex relations. Meanwhile, without adding a relation specific translational vector enables the model to encode several key relation patterns. We show these capabilities below (which are also proved in Appendix A).

PROPOSITION 1. *PairRE can encode symmetry/antisymmetry relation pattern.*

PROPOSITION 2. *PairRE can encode inverse relation pattern.*

PROPOSITION 3. *PairRE can encode composition relation pattern.*

¹Hadamard product means entry-wise product.

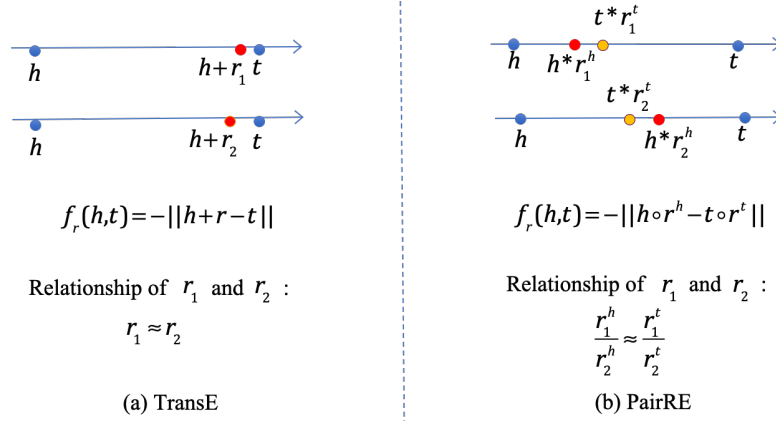


Figure 2: Examples of TransE and PairRE to represent subrelation. The dimension of entity embedding equals to one. Triples (h, r_1, t) and (h, r_2, t) are examples for subrelations r_1 and r_2 . For TransE, the learned embeddings of r_1 and r_2 can be very close. While PairRE can capture similarities of r_1 and r_2 by forcing $r_1^h/r_2^h \approx r_1^t/r_2^t$. The relation embeddings of r_1 and r_2 in PairRE can be quite different, which enables them to capture more specific information.

Subrelation. Besides the above three relation patterns, we further analyze another important relation pattern, subrelation. Currently, most models try to solve the 1-to-N/N-to-1/N-to-N complex relations. However, due to the incompleteness of knowledge graphs, the connected entities of a particular subrelation pair always have large overlap. This may lead representations of the subrelation pair undistinguishable. As Figure 2 shows, with limited embedding dimension, subrelations r_1 and r_2 in TransE may take the same value. While the paired vectors of relation embedding enables PairRE to represent the subrelations in fine-grained way. As shown in Figure 2, the similarities of subrelations r_1 and r_2 can be captured by $r_1^h \circ r_2^t - r_1^t \circ r_2^h$. The differences can be fully represented by their embeddings since they can be totally different. The ability to distinguish subrelations enables PairRE to better model triples. Besides, PairRE can encode subrelations given simple constraints on relation representations.

PROPOSITION 4. *PairRE can encode subrelation relation pattern using inequality constraints.*

PROOF. For subrelations $r_1 \rightarrow r_2$, we impose the following constraints:

$$r_{2,i}^h / r_{1,i}^h = r_{2,i}^t / r_{1,i}^t = \alpha_i, |\alpha_i| \leq 1, \quad (2)$$

where $0 \leq i \leq d$. Then we can get

$$\begin{aligned} f_{r_2}(h, t) - f_{r_1}(h, t) &= ||h \circ r_1^h - t \circ r_1^t|| - ||h \circ r_2^h - t \circ r_2^t|| \\ &= ||h \circ r_1^h - t \circ r_1^t|| - ||\alpha \circ (h \circ r_1^h - t \circ r_1^t)|| \geq 0, \end{aligned} \quad (3)$$

for any two entities $h, t \in \mathcal{E}$. That is, when the constraints are satisfied, the plausibility of fact triple (h, r_2, t) is more than triple (h, r_1, t) . \square

Dataset	$ \mathcal{R} $	$ \mathcal{E} $	Train	Valid	Test
ogbl-wikikg	535	2,500k	16,109k	429k	598k
ogbl-biokg	51	94k	4,763k	163k	163k
FB15k	13k	15k	483k	50k	59k
FB15k-237	237	15k	272k	18k	20k
DB100k	470	100k	598k	50k	50k
Sports	4	1039	1312	-	307

Table 2: Number of entities, relations, and observed triples in each split for the six benchmarks.

Optimization. To optimize the model, we utilize the self-adversarial negative sampling loss [26] as objective for training:

$$L = -\log \sigma(\gamma - f_r(h, t)) - \sum_{i=1}^n p(h'_i, r, t'_i) \log \sigma(f_r(h'_i, r, t'_i) - \gamma), \quad (4)$$

where γ is a fixed margin and σ is the sigmoid function. (h'_i, r, t'_i) is the i_{th} negative triple and $p(h'_i, r, t'_i)$ represents the weight of this negative sample. $p(h'_i, r, t'_i)$ is defined as follows:

$$p((h'_i, r, t'_i) | (h, r, t)) = \frac{\exp f_r(h'_i, t'_i)}{\sum_j \exp f_r(h'_j, t'_j)}. \quad (5)$$

5 EXPERIMENTAL RESULTS

5.1 Experimental setup

We evaluate the proposed method on link prediction tasks. At first, we validate the ability to deal with complex relations and symmetry/antisymmetry, inverse and composition relations on four benchmarks. Then we validate our model on two subrelation specific benchmarks. Statistics of these benchmarks are shown in Table 2. We show these benchmarks below:

-	ogbl-wikikg			ogbl-biokg		
Model	#Dim	Test MRR	Valid MRR	#Dim	Test MRR	Valid MRR
TransE	100	0.2535 \pm 0.004	0.4587 \pm 0.003	-	-	-
DistMult	100	0.3434 \pm 0.008	0.3412 \pm 0.007	-	-	-
ComplEx	50	0.3877 \pm 0.005	0.3612 \pm 0.006	-	-	-
RotatE	50	0.2681 \pm 0.005	0.3613 \pm 0.003	-	-	-
PairRE	100	0.4912 \pm 0.004	0.5013 \pm 0.004	-	-	-
TransE	600 \dagger	0.4536 \pm 0.003	0.4587 \pm 0.003	2000	0.7452 \pm 0.0004	0.7456 \pm 0.0003
DistMult	600 \dagger	0.3612 \pm 0.003	0.3403 \pm 0.001	2000	0.8043 \pm 0.0003	0.8055 \pm 0.0003
ComplEx	300 \dagger	0.4028 \pm 0.003	0.3787 \pm 0.004	1000	0.8095 \pm 0.0007	0.8105 \pm 0.0001
RotatE	300 \dagger	0.3626 \pm 0.004	0.3613 \pm 0.003	1000	0.7989 \pm 0.0004	0.7997 \pm 0.0002
PairRE	200	0.5289 \pm 0.003	0.5529 \pm 0.001	2000	0.8164 \pm 0.0005	0.8172 \pm 0.0005

Table 3: Link prediction results on ogbl-wikikg and ogbl-biokg. Best results are in bold. All the results except PairRE are from [12]. \dagger requires a GPU with 48GB memory. PairRE runs on a GPU with 16GB memory.

-	FB15k					FB15k-237				
Model	MR	MRR	Hit@10	Hit@3	Hit@1	MR	MRR	Hit@10	Hit@3	Hit@1
TransE \dagger	-	0.463	0.749	0.578	0.297	357	0.294	0.465	-	-
DistMult \diamond	42	0.798	0.893	-	-	254	0.241	0.419	0.263	0.155
HolE	-	0.524	0.739	0.759	0.599	-	-	-	-	-
ConvE	51	0.657	0.831	0.723	0.558	244	0.325	0.501	0.356	0.237
ComplEx	-	0.692	0.840	0.759	0.599	339	0.247	0.428	0.275	0.158
Simple	-	0.727	0.838	0.773	0.660	-	-	-	-	-
RotatE	40	0.797	0.884	0.830	0.746	177	0.338	0.533	0.375	0.241
Seek	-	0.825	0.886	0.841	0.792	-	-	-	-	-
OTE	-	-	-	-	-	-	0.351	0.537	0.388	0.258
GC-OTE	-	-	-	-	-	-	0.361	0.550	0.396	0.267
PairRE	38	0.811	0.895	0.843	0.762	160	0.351	0.544	0.387	0.256

Table 4: Link prediction results on FB15k and FB15k-237. Results of \dagger are taken from [21]; Results of \diamond are taken from [15]. Other results are taken from the corresponding papers. GC-OTE model adds graph context to OTE model [27].

- **ogbl-wikikg** [12] is extracted from Wikidata knowledge base [30]. One of the main challenges for this dataset is **complex relations**.
- **ogbl-biokg** [12] contains data from a large number of biomedical data repositories. It describes relationships among diseases, proteins, drugs, side effects and protein functions. One of the main challenges for this dataset is **symmetry** relations.
- **FB15k** [3] contains triples from Freebase. The triples in this dataset mainly describe movies, actors, awards, sports, and sport teams. One of the main relation patterns is **inverse**.
- **FB15k-237** [28] is a subset of FB15k, with inverse relations removed. The main relation patterns antisymmetry and **composition**.
- **DB100k** [7] is a subset of DBpedia. The main relation patterns are composition and **subrelation**.
- **Sports** [31] is a subset of NELL [20]. The main relation patterns are antisymmetry and **subrelation**.

Evaluation protocol. Following the state-of-the-art methods, we measure the quality of the ranking of each test triple among all possible head entity and tail entity substitutions: (h', r, t) and (h, r, t') , $\forall h', \forall t' \in \mathcal{E}$. Three evaluation metrics, including Mean Rank(MR), Mean Reciprocal Rank (MRR) and Hit ratio with cut-off values $n = 1, 3, 10$, are utilized. MR measures the average rank of all correct entities. MRR is the average inverse rank for correct entities with higher value representing better performance. Hit@ n measures the percentage of correct entities in the top n predictions. The rankings of triples are computed after removing all the other observed triples that appear in either training, validation or test set. This means only filtered results [3] are reported.

Implementation. The benchmarks, ogbl-wikikg and ogbl-biokg, come from Open Graph Benchmark [12]. We utilize the official implementation of organizers for these two datasets. Only the hyper-parameter γ and embedding dimension are tuned. The other settings are kept the same with baselines. For the rest experiments, we implement our models based on the implementation of RotatE [26]. All hyper-parameters except γ and embedding dimension are kept the same with RotatE.

Subrelation
(h, CoachesTeam, t) \rightarrow (h, PersonBelongsToOrganization, t)
(h, AthleteSportsTeam, t) \rightarrow (h, AthletePlaysForTeam, t)

Table 5: The added subrelation rules for Sports dataset.

5.2 Main results

Comparisons for ogbl-wikikg and ogbl-biokg are shown in Table 3. On these two large scale datasets, PairRE achieves state-of-the-art performances. For ogbl-wikikg dataset, PairRE performs best on both limited embedding dimension and increased embedding dimension. With the same parameter number of ComplEx (dimension 100), PairRE improves Test MRR close to 10%. With increased dimension, all models are able to achieve higher MRR on validation and test sets. Due to the limitation of hardware, we only increase embedding dimension to 200 for PairRE. PairRE also outperforms all baselines and improves Test MRR almost 7.5%. **Based on performances of baselines, the performance of PairRE may be improved further if embedding dimension is increased to 600.** Under the same experiment setting and the same number of parameters, PairRE also outperforms all baselines on ogbl-biokg dataset. It improves Test MRR by 0.69%, which proves the superior ability to encode symmetry relations.

Comparisons for FB15k and FB15k-237 datasets are shown in Table 4. Since our model shares the same hyper-parameter settings and implementation with RotatE, comparing with this state-of-the-art model is fair to show the advantage and disadvantage of the proposed model. Besides, the comparisons also include several leading methods, such as TransE [3], DistMult [35], HolE [21], ConvE [6], ComplEx [29], Simple [16], SeeK [34] and OTE [27]. Compared with RotatE, PairRE shows clear improvements on FB15k and FB15k-237 for all evaluation metrics. For MRR metric, the improvements are 1.4% and 1.3% respectively. Compared with the other leading methods, PairRE also shows highly competitive performances. All these comparisons prove the effectiveness of PairRE to encode inverse and composition relations.

5.3 Further experiments on subrelation

We further compare our method with two of the leading methods ComplEx-NNE-AER and Simple⁺, which focus on encoding subrelations. All these two methods add subrelation rules to the semantic matching models. We utilize these rules to add constraints on relation representations for PairRE. Two ways are validated. We first test the performance of weight tying for subrelation rules. The rules ($r_1 \rightarrow r_2$) are added as follows:

$$\begin{aligned} r_2^h &= r_1^h \circ \cos(\theta), \\ r_2^t &= r_1^t \circ \cos(\theta), \end{aligned} \quad (6)$$

where $\theta \in \mathbb{R}^d$.

Sports dataset is utilized in this section. The added rules are shown in Table 5. For Sports dataset, the results have a high variance, as such we average 10 runs and produce 95% confidence

Model	MRR	hit@1
Simple	0.230	0.184
Simple ⁺	0.404	0.349
PairRE	0.468 \pm 0.003	0.416 \pm 0.005
PairRE+Rule	0.475 \pm 0.003	0.432 \pm 0.004

Table 6: Link prediction results on Sports dataset. Other results are taken from [8].

Model	MRR	Hit@10	Hit@3	Hit@1
TransE	0.111	0.270	0.164	0.016
DistMult	0.233	0.448	0.301	0.115
HolE	0.260	0.411	0.309	0.182
ComplEx	0.242	0.440	0.312	0.126
SeeK	0.338	0.467	0.370	0.268
ComplEx-NNE	0.298	0.426	0.330	0.229
ComplEx-NNE-AER	0.306	0.418	0.334	0.244
PairRE	0.412	0.600	0.472	0.309
PairRE+rule	0.419	0.599	0.475	0.321

Table 7: Link prediction results on DB100k. All the results are taken from the corresponding papers.

intervals. The experiments results in Table 6 show effectiveness of the proposed method.

Weight tying on relation representation is a way to incorporate strict, hard rules. The soft rules can also be incorporated into PairRE by approximate entailment constraints on relation representations. In this section, we add the same rules from ComplEx-NNE-AER, which includes subrelation and inverse rules. Compared to ComplEx-NNE-AER, PairRE only needs add constraints on relation representations. We denote by $r_1 \xrightarrow{\lambda} r_2$ the approximate entailment between relations r_1 and r_2 , with confidence level λ . The objective for training is then changed to:

$$\begin{aligned} L_{rule} = L + \mu \sum_{\tau_{subrelation}} \lambda 1^T (r_1^h \circ r_2^t - r_1^t \circ r_2^h)^2 \\ + \mu \sum_{\tau_{inverse}} \lambda 1^T (r_1^h \circ r_2^h - r_1^t \circ r_2^t)^2, \end{aligned} \quad (7)$$

where L is calculated from Equation 4, μ is loss weight for added constraints, $()^2$ means an entry-wise operation, $\tau_{subrelation}$ and $\tau_{inverse}$ are the sets of subrelation rules and inverse rules respectively. Following [7], we take the corresponding two relations from subrelation rules as equivalence. Because $\tau_{subrelation}$ contains both rule $r_1 \rightarrow r_2$ and rule $r_2 \rightarrow r_1$. Although some relations are taken as equivalence, our model will not force their embeddings to same values. This is advantageous compared to ComplEx-NNE-AER as our model has the ability to capture both the similarities and differences of subrelation pairs.

We validate our method on DB100k dataset. The results are shown in Table 7. We can see PairRE outperforms the recent state-of-the-art SeeK and ComplEx based models with large margins on all evaluation metrics. With added constraints, the performance of PairRE is improved further. The improvements for the added rules are 0.7%, 1.2% for MRR and Hit@1 metrics respectively.

Model	FB15k(Hits@10)				ogbl-wikikg(Hits@10)			
	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
KGE2E_KL[11]	0.925	0.813	0.802	0.715	-	-	-	-
TransE	0.887	0.822	0.766	0.895	0.362	0.064	0.400	0.220
ComplEx	0.939	0.896	0.822	0.902	0.501	0.345	0.505	0.517
RotatE	0.923	0.840	0.782	0.908	0.398	0.144	0.432	0.262
PairRE	0.785	0.899	0.872	0.940	0.582	0.282	0.598	0.602

Table 8: Experimental results on FB15k and ogbl-wikikg by relation category. Best results are in bold. Results on FB15k are taken from RotatE [26]. The embedding dimensions for models on ogbl-wikikg are same to the experiments in Table 3, which is 100 for real space models and 50 for complex value based models.

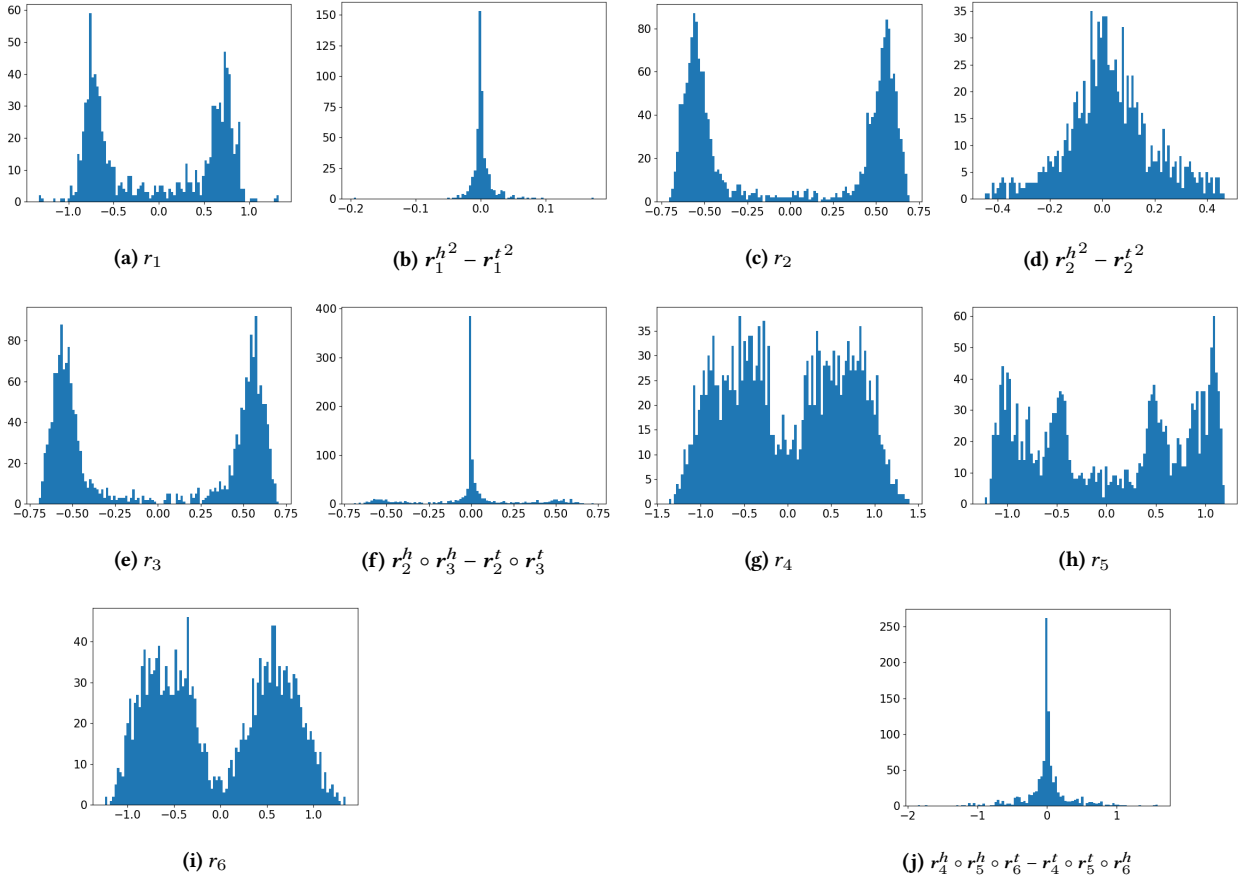


Figure 3: Histograms of relation embeddings for different relation patterns. r_1 is relation *spouse*. r_2 is relation */broadcast/tv_station/owner*. r_3 is relation */broadcast/tv_station_owner/tv_stations*. r_4 is relation */location/administrative_division/capital/location/administrative_division_capital_relationship/capital*. r_5 is relation */location/hud_county_place/place*. r_6 is relation *base/areas/schema/administrative_area/capital*.

5.4 Model analysis

Analysis on complex relations

We analyze the performances of PairRE for complex relations. The results of PairRE on different relation categories on FB15k and ogbl-wikikg are summarized into Table 8. We compare our method to

several leading methods. We can see PairRE performs quite well on N-to-N and N-to-1 relations. It has a significant lead over baselines. We also notice that the performance of 1-to-N relations on ogbl-wikikg dataset is not as strong as the other relation categories. One of the reasons is that only 2.2% of test triples are belong to the 1-to-N relation category.

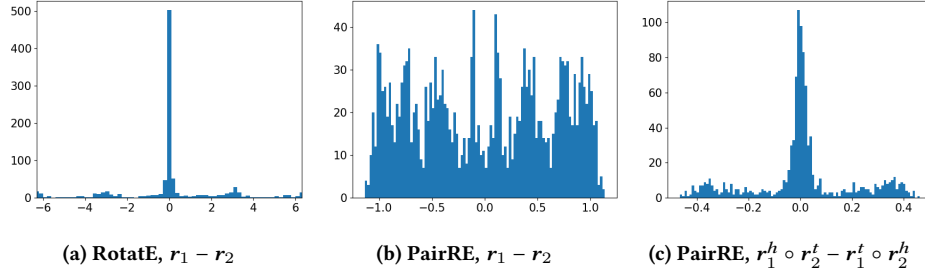


Figure 4: Comparison of RotatE and PairRE for representing subrelation pair : $/medicine/drug/drug_class$ (r_1) and $/medicine/drug_ingredient/active_moiety_of_drug$ (r_2).

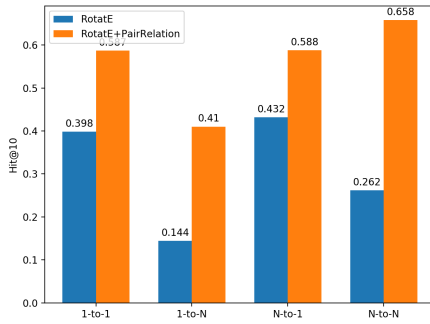


Figure 5: Performance comparison between RotatE and RotatE+PairRelation on ogbl-wikikg dataset. The results of Hit@10 for different relation categories are compared.

In order to further test the performance of paired relation vectors, we change the relation vector in RotatE to paired vectors. In the modified RotatE model, both head and tail entities are rotated with different angles based on the paired relation vectors. This model can also be seen as complex value based PairRE. We name this model as RotatE+PairRelation. The experiment results are shown in Figure 5. With the same embedding dimension (50 in the experiments), RotatE+PairRelation improves performance of RotatE with 18.9%, 26.6%, 15.6% and 39.6% on 1-to-1, 1-to-N, N-to-1 and N-to-N relation categories respectively. These significant improvements prove the superior ability of paired relation vectors to handle complex relations.

Case study on relation patterns

To further verify the learned relation patterns, we visualize some examples. We plot histograms of the learned relation embeddings.

Symmetry/AntiSymmetry. Figure 3a shows a symmetry relation *spouse* from DB100k. The embedding dimension is 500. In PairRE, symmetry relation pattern can be encoded when embedding r satisfies $r^{h^2} = r^{t^2}$. Figure 3b shows most of the paired elements in r^h and r^t have the same absolute value. Figure 3c shows a antisymmetry relation *tv_station_owner*. Different with symmetry relation, most of the paired elements do not have the same absolute value as shown in Figure 3d.

Inverse. Figure 3c and Figure 3e show an example of inverse relations from FB15k. As the histogram in Figure 3f shows these two inverse relations *tv_station_owner* and *tv_station_owner_to_stations* close to satisfy $r_1^h \circ r_2^h = r_1^t \circ r_2^t$.

Composition. Figures 3g, 3h, 3i show an example of composition relation pattern from FB15k, where the third relation r_6 can be seen as the composition of the first relation r_4 and the second relation r_5 . As Figure 3j shows these three relations close to satisfy $r_4^h \circ r_5^h \circ r_6^t = r_4^t \circ r_5^t \circ r_6^h$.

Case study on representing subrelation

To show the superiority of PairRE to represent subrelation, we compare our model with RotatE in Figure 4. We get the embeddings of RotatE from their public implementation and default hyperparameters. Based on AMIE [9], the empirical precision of rule $/medicine/drug/drug_class$ (r_1) \rightarrow $/medicine/drug_ingredient/active_moiety_of_drug$ (r_2) from FB15k is 1.0. As we can see the learned embeddings of RotatE take same values on most dimensions. While PairRE can distinguish these two relations more effectively. The connections of these two relations are also well captured as shown in Figure 4c. We think the ability to distinguish subrelation representations is important as relations can connect and influence a lot of entities.

6 CONCLUSION

To better handle complex relations and tackle more relation patterns for knowledge graph embedding, we proposed PairRE, which represents each relation with paired vectors. With a slight increase in complexity, PairRE can be the first distance based model that well solves the aforementioned two problems. Beyond the symmetry/antisymmetry, inverse and composition relations, PairRE can further encode subrelation with simple constraint on relation representations. Moreover, our model has advantages in representing relation pairs whose connected entities have large overlap. On large scale benchmark ogbl-wikikg, PairRE outperforms all the state-of-the-art baselines with a large margin. Experiments on other well designed benchmarks also demonstrate the effectiveness of the focused key abilities.

REFERENCES

- [1] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013*

- Conference on Empirical Methods in Natural Language Processing. 1533–1544.
- [2] 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*. ACM, 1247–1250.
- [3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
- [4] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*.
- [5] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 165–180.
- [6] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [7] Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. *arXiv preprint arXiv:1805.02408* (2018).
- [8] Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. 2019. Improved knowledge graph embedding using background taxonomic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3526–3533.
- [9] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*. 413–422.
- [10] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [11] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. 623–632.
- [12] 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. *arXiv preprint arXiv:2005.00687* (2020).
- [13] Guoliang Ji, Shizhu He, Lihong 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)*. 687–696.
- [14] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In *AAAI*, Vol. 16. 985–991.
- [15] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744* (2017).
- [16] Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*. 4284–4295.
- [17] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic web* 6, 2 (2015), 167–195.
- [18] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.
- [19] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.
- [20] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhanava Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Commun. ACM* 61, 5 (2018), 103–115.
- [21] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*.
- [22] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*, Vol. 11. 809–816.
- [23] Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. In *Advances in Neural Information Processing Systems*. 7710–7720.
- [24] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.
- [25] 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*. ACM, 697–706.
- [26] 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* (2019).
- [27] Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2019. Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding. *arXiv preprint arXiv:1911.04910* (2019).
- [28] 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*. 57–66.
- [29] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*. 2071–2080.
- [30] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
- [31] Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- [32] Yanjie Wang, Rainer Gemulla, and Hui Li. 2018. On Multi-Relational Link Prediction With Bilinear Models.
- [33] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.
- [34] Wentao Xu, Shun Zheng, Liang He, Bin Shao, Jian Yin, and Tie-Yan Liu. 2020. SEEK: Segmented Embedding of Knowledge Graphs. *arXiv preprint arXiv:2005.00856* (2020).
- [35] 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* (2014).
- [36] 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*. ACM, 353–362.
- [37] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embedding. *arXiv preprint arXiv:1904.10281* (2019).
- [38] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y Chang, and Xiaoyan Zhu. 2012. Entity disambiguation with freebase. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 82–89.

A PROOF OF PROPOSITIONS

Let r_1, r_2, r_3 denote relations and e_1, e_2, e_3 denote entities. The entities are sampled from the entity set \mathcal{E} randomly.

A.1 Proof of Proposition 1

PROOF. If $(e_1, r_1, e_2) \in \mathcal{T}$ and $(e_2, r_1, e_1) \in \mathcal{T}$, we have

$$\begin{aligned} e_1 \circ r_1^h &= e_2 \circ r_1^t \wedge e_2 \circ r_1^h = e_1 \circ r_1^t \\ &\Rightarrow r_1^{h2} = r_1^{t2} \end{aligned} \quad (8)$$

if $(e_1, r_1, e_2) \in \mathcal{T}$ and $(e_2, r_1, e_1) \notin \mathcal{T}$, we have

$$\begin{aligned} e_1 \circ r_1^h &= e_2 \circ r_1^t \wedge e_2 \circ r_1^h \neq e_1 \circ r_1^t \\ &\Rightarrow r_1^{h2} \neq r_1^{t2} \end{aligned} \quad (9)$$

□

A.2 Proof of Proposition 2

PROOF. If $(e_1, r_1, e_2) \in \mathcal{T}$ and $(e_2, r_2, e_1) \in \mathcal{T}$, we have

$$\begin{aligned} e_1 \circ r_1^h &= e_2 \circ r_1^t \wedge e_2 \circ r_2^h = e_1 \circ r_2^t \\ &\Rightarrow r_1^h \circ r_2^h = r_1^t \circ r_2^t \end{aligned} \quad (10)$$

□

A.3 Proof of Proposition 3

PROOF. If $(e_1, r_1, e_2) \in \mathcal{T}$, $(e_2, r_2, e_3) \in \mathcal{T}$ and $(e_1, r_3, e_3) \in \mathcal{T}$, we have

$$\begin{aligned} e_1 \circ r_1^h &= e_2 \circ r_1^t \wedge e_2 \circ r_2^h = e_3 \circ r_2^t \wedge \\ &e_1 \circ r_3^h = e_3 \circ r_3^t \\ &\Rightarrow r_1^t \circ r_2^t \circ r_3^h = r_1^h \circ r_2^h \circ r_3^t \end{aligned} \quad (11)$$

□