Parameters Conjugation for Knowledge Graph Embeddings in Complex Space

Xincan Feng[†‡], Zhi Qu[†], Yuchang Cheng[‡], Taro Watanabe[†], Nobuhiro Yugami[‡]

[†]Natural Language Processing Laboratory, Nara Institute of Science and Technology

[‡]Multilingual Knowledge Computing Laboratory, Fujitsu Ltd.

{feng.xincan.fy2, qu.zhi.pv5, taro}@is.naist.jp

{cheng.yuchang, yugami}@fujitsu.com

Abstract

Knowledge Graph Embedding (KGE) have been a popular approach to do link prediction task for Knowledge Graph Completion (KGC). Researchers are designing increasingly complicated models to improve model expressiveness. We need to simplify the models without losing performance. Besides, many Knowledge Graph (KG) applications like (https://aclanthology.org/2021.acl-demo.14.pdf) regard visual appeal as important, where appropriate visuals can better convey the points of the data and facilitate user interaction. Our research proposed parameters conjugation a dimension-level restriction method, to reduce computation to at most 50% and help obtain better visualization while receiving comparable performance in most cases in the complex number represented KGE models. We verified our method's generalizability in two best KGE models: $5^\star$E (Nayyeri et al. 2021) (**?**) and ComplEx (Trouillon et al. 2016).

1 Introduction

Knowledge Graph (KG) has become trustable information source in applications including question answering, data integration, recommender systems (Ji et al. 2020) and automated personal agents (Trouillon et al. 2016). Knowledge Graph (KG) is the directed graphical Knowledge Base (KB) representation in the form of (head, relation, tail) triples. The heads and tails in KG triples are also called the entities, and the relations are correspondingly called the links that relate the entities. In the graph representation of KG, entities are usually represented as nodes, and links are represented as edges. From a mathematical point of view, entities and links are often represented as vectors and transformation functions, respectively, in Knowledge Graph Embedding (KGE) models.

KGE is a prominent approach used for Knowledge Graph Completion (KGC) by predicting missing links (Nayyeri et al. 2021). We do KGC because, in addition to the existing triples, KGE models can learn missing links that indicate implicit relations among heads and tails in the KG. Link Prediction task is one of the main problems in Statistical Relational Learning (SRL, Getoor and Taskar, 2007). KGC extends KG's applicability by saving repetitive human labor for constructing KBs.

KGE models usually define a specific representation, and use a transformation function to map entities (nodes) through links (edges) into a corresponding vector space, and then calculate the plausibility of triples via a corresponding score function (Nayyeri et al. 2021). Thus, the main directions in improving the KGE models are: representations, vector spaces, and transformation functions. We generically concluded the following about above three directions:

- Compared with real number representation, the composition of complex number can handle a larger variety of relations, among them symmetric and antisymmetric relations (Trouillon et al. 2016).

- Compared with Euclidean space, the Hyperbolic space models can save more structures using fixed or trainable curvatures in different positions for hierarchical relations (Chami et al. 2020).

- Compared with naive addition or multiplication transformation functions, more specific designed functions, including projective geometric functions, support multiple simultaneous transformations, and are more capable to represent multiple structures in multirelational KGs (Nayyeri et al. 2021).

The relationship among transformation functions, representation methods and dimensions are that: The transformation functions distinguish the

extent to which the KGE model is able to learn complicated patterns formed by combinations of entities and links (Nayyeri et al. 2021). The representation methods of different vector spaces are where the transformation functions built with. Dimensions are the base of the representation methods and can thus nail the transformation functions' capabilities. From above, we noticed two points:

- The increased performance from real number to complex number representation, which enables the imaginary and real part dimensions to be relational, indicates the benifit of **hidden relation** among parameter dimensions.

- The increased performance from Euclidean space to Hyperbolic space, which enables the distances or angles of **different positions** to vary in different degrees, indicates different positions in the parameter dimensions could have different degrees of freedom.

Here was born our intuition that the dependent parameter dimensions are maybe capable enough compared with independent dimensions to do KGE. Since, although the KGE models have kept on discovering more complex transformations, the relationship between the transformation complexity and the model expressiveness is sophisticated. Using conjugation on complex number is a natural thought.

Our contribution is that we proposed the parameters conjugation method, i.e. a dimension-level restriction method used in the complex number represented KGE models. Specifically, our method formulates conjugated parameters in appropriate dimensions of the transformation functions, so that to reduce parameter size and computation. Although the origin models can be different, using our method will more or less reduce the computation. Experiments demonstrated that our conjugated models usually receive comparable performance as the origin models under the same hyperparameters settings. We verified our method's generalizability in two best performing KGE models: $5^\star$E (Nayyeri et al. 2021) and ComplEx (Trouillon et al. 2016). Besides, because our method creates stronger geometric restriction on nodes and edges, it also obtains better visualization of the KGs than the origin models by making the generated KGs clearer.

The remainder of the paper is organized as follows: We first formulated our parameters conjugation method in two baseline models, call our conjugate models complexC and $5^\star$Ce. We then describe experiments on large scale public benchmark datasets, in which we empirically show that our method not only reduces computation, but also brings better visualization for KG applications.

To compare our method's limitations with original approaches using only half of the parameters, we also did auxiliary experiments to see how the original models would perform with only half of the parameters in the regularization term. This should help practitioners when applying our method to more datasets.

2 Related work

KGE models can be classified in many ways. Our method involves both the representation and the dimensional position of the model. We describe the classification according to the representation methods, and the classification according to the vector spaces, that has inspired our method, and the related work under these two classifications.

KGE models that are classified according to their representation methods:

Real Embeddings Translation approaches include TransE (Bordes et al. 2013) and its variants (Ji et al. 2015, Lin et al. 2015). Although these models are fairly simple and have few parameters, they fail to encode important logical properties (e.g., translations can't encode symmetry).

Complex Embeddings ComplEx (Trouillon et al. 2016) gives a clear comparison with respect to existing approaches using only real numbers by presenting an equivalent reformulation of their model that involves only real embeddings. RotatE (Sun et al. 2019) does require additional optimization component to effectively draw negative samples for training. $5^\star$E (Nayyeri et al. 2021) uses Möbius transformation as the transformation function, which is the composition of four functions that can represent five transformations.

KGE models that are classified according to their vector spaces:

Euclidean Embeddings Tensor factorization methods such as RESCAL (Nickel et al. 2011) and DistMult (Yang et al. 2015) are designed based on element-wise multiplication of transformed head and tail. In this case, the plausibility of triples is measured based on the angle of transformed head and tail.

Non-Euclidean Embeddings MuRP (Balaze-

vic et al. 2019) minimizes hyperbolic distances between a re-scaled version of the head entity embedding and a translation of the tail entity embedding. It uses hyperbolic embeddings with fewer dimensions than its Euclidean analogues. ATTH (Chami et al. 2020) leverages trainable hyperbolic curvatures per relationship to simultaneously capture logical patterns and hierarchies. However, it also needs optimization methods in tangent space (i.e., Euclidean).

3 Method

Our method turns half of the transformation function parameters into the complex conjugate; the position selection of the conjugated parameters depends on the original model. Complex embeddings set relational restriction for entities and links representation. Our method further adds relational restriction within the transformation function parameters.

3.1 Method Formulation

$a, b, c, d \in \mathbb{C}$ are the parameters in the relation embedding matrices, also the transformation functions, $x \in \mathbb{C}$ is the entity embedding, $Re(x)$ is the real part of the complex number x, $\overline{x}$ is the complex conjugate of x.

ComplEx (Trouillon et al. 2016) is the first and also one of the best performing complex embedding models in Euclidean space. They discussed the problem of finding the best ratio between expressiveness and parameter space size as the keystone of KGE models. They demonstrated that complex vectors can effectively capture antisymmetric relations while retaining the efficiency benefits of the dot product, that is linearity in both space and time complexity. In this model, relations are represented as real part of low-rank normal matrices, and relation embeddings naturally act as weights on each latent dimension. Geometrically, each relation embedding is an anisotropic scaling of the basis defined by the entity embeddings, followed by a projection onto the real subspace.

By using our method, the original model ComplEx is adapted to the conjugate model complexC, where relations are represented as the real part of low-rank conjugate matrices. Specifically, we set the original normal square relation embedding matrices $\begin{bmatrix} a \end{bmatrix}$ to be half the normal parameters and the other half their conjugate param-

eters. I.e. $\begin{bmatrix} a & \overline{a} \end{bmatrix}$. In this model, since each parameter position is of the same importance, the positions of the conjugate parameters can be set randomly.

$5^{\star}\text{E}$ (Nayyeri et al. 2021) is a novel complex embedding model in Non-Euclidean space. They tackled the problem of multiple sub-graph structures in the neighborhood (e.g. combinations of path and loop structures). Unlike the ComplEx model, they replaced the dot product with the Möbius function to project the entities into Non-Euclidean space, so that to support multiple simultaneous transformations. The Möbius function has several favorable theoretical properties and this model subsumes ComplEx in that it embeds entities in much lower ranks with better performance. (But the authors haven't published the hyperparameters, and we failed to reproduce the same performance.) However, $5^{\star}\text{E}$ is inferior to ComplEx because the calculation is more sophisticated and hence costs more time.

Our method transforms the original model $5^{\star}\text{E}$ into the conjugate model $5^{\star}\text{Ce}$, where relations are represented as the real part of low-rank conjugate matrices. Specifically, we set the original normal square relation embedding matrices $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ to be half the normal parameters and the other half their conjugate parameters. I.e. $\begin{bmatrix} a & b \\ \overline{b} & \overline{a} \end{bmatrix}$, or $\begin{bmatrix} a & b \\ -\overline{b} & \overline{a} \end{bmatrix}$. In this model, different positions of the parameters play different roles, and the best conjugate positions are the principal and secondary diagonal positions. Note that experiments showed that, the parameters conjugation or negative parameters conjugation method achieved the same accuracy. Negative parameters conjugation on this model is equivalent to restricting the original Möbius function to the unitary Möbius transformation.

3.2 Transformation Analysis

ComplEx: $a \in \mathbb{C}$

$$x \to \begin{bmatrix} a \end{bmatrix} x \to ax$$

complexC: $a, \overline{a} \in \mathbb{C}$

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \to \begin{bmatrix} a & \overline{a} \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \to \begin{bmatrix} ax_1 & \overline{a}x_2 \end{bmatrix}$$

$5^{\star}\text{E}$: $a, b, c, d \in \mathbb{C}$

$$x \to \begin{bmatrix} x \\ 1 \end{bmatrix} \to \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \to \frac{ax + b}{cx + d}$$

Möbius transformation ϑ is capable of performing five kinds of transformations simultaneously because it combines five subsequent transformations: $\vartheta = \vartheta_4 \circ \vartheta_3 \circ \vartheta_2 \circ \vartheta_1$. Among them $\vartheta_1 = x + \frac{d}{c}$ (translation by $\frac{d}{c}$), $\vartheta_2 = \frac{1}{x}$ (inversion and reflection w.r.t. real axis), $\vartheta_3 = \frac{bc-ad}{c^2}x$ (homothety and rotation), and $\vartheta_4 = x + \frac{a}{c}$ (translation by $\frac{a}{c}$).

$5^\star$Ce: $c = \bar{b}, d = \bar{a}$ (Note that $c = -\bar{b}, d = \bar{a}$ works fine as well, and this forms a unitary Möbius transformation.)

$$ x \rightarrow \begin{bmatrix} x \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} a & b \\ \bar{b} & \bar{a} \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \rightarrow \frac{ax+b}{\bar{b}x+\bar{a}} $$

The five subsequent transformations turn into: $c = \bar{b}, d = \bar{a}$ $\vartheta_1 = x + \frac{\bar{a}}{\bar{b}}$ (translation by $\frac{\bar{a}}{\bar{b}}$), $\vartheta_2 = \frac{1}{x}$ (inversion and reflection w.r.t. real axis), $\vartheta_3 = \frac{b\bar{b}-a\bar{a}}{\bar{b}^2}x$ (homothety and rotation), and $\vartheta_4 = x + \frac{a}{\bar{b}}$ (translation by $\frac{a}{\bar{b}}$).

We conclude from above that:

(i) The transformed entities in our conjugate models basicly retain the expressiveness for different relation patterns compared to their original ones.

(ii) The vector representations of transformed entities have more substantial geometric constraints (transformed entities are illustrated in Figure **??**).

4 Experiments

4.1 Experimental Setup

Experiments Purposes To evaluate the performance of our method, and to explore potential problems and the reasons, we conducted the following comparison experiments. (i) We experimented on five datasets to explore how the model performs on different datasets. (ii) We experimented using different hyperparameter settings to explore whether hyperparameters influence our method. (iii) We experimented with different conjugate dimensions to explore what kind of dimensions are more suitable for $5^\star$E. We set $c = -\bar{b}, d = \bar{a}$ in model $5^\star$C$^-$e. And $c = \bar{a}, d = \bar{b}$ in model $5^\star$De. (iv) We experimented where the regularization term is reduced to half of the parameters on the original model ($5^\star$E$_{\text{hreg}}$) to explore whether the flaw in our model is similar to the effect of reduced parameters. We compared all experiments results running on GeForce GTX 1080 Ti, except for the largest database YAGO3-10

which uses Tesla V100S-PCIE-32GB. Our method is implemented in Pytorch[1] and the code[2] is available online.

Metrics We followed the standard evaluation protocal for KGE models as $5^\star$E (Nayyeri et al. 2021). T: the rank set of truth, r_i: the rank position r of the first true entity for the i-th query. We computed two rank-based metrics:

(i) Mean Reciprocal Rank (MRR), which computes the arithmetic mean of reciprocal ranks of all true entities from the ranked list of answers to queries T. and (ii) Hits@N (N = 1, 3, 10), which counts the true entities and calculate their proportion in the truth T in top N sorted predicted answers' list.

$$ \text{MRR} = \frac{1}{T} \sum_{i=1}^{T} \frac{1}{r_i} $$

$$ \text{Hits@}N = \frac{1}{T} \sum_{r \in T, r \leq N} \mathbb{I} $$

Datasets We evaluated our method on five widely used benchmark datasets: FB15K (Bordes et al. 2013) is a subset of Freebase, the contents of which are general facts. WN18 (Bordes et al. 2013) is a subset of Wordnet, a database that features lexical relations between words. FB15K-237 (Toutanova and Chen 2015) is a subset of FB15K. WN18RR (Dettmers et al. 2018) is a subset of WN18. YAGO3-10 (Dettmers et al. 2018) is the largest in common datasets, it mostly describes attributes of persons, and contains entities associated with at least ten different relations.

As was first noted by Toutanova and Chen (2015) that WN18 and FB15k suffer from test leakage through inverse relations: For example, the test set frequently contains triples such as (s, hyponym, o) while the training set contains its inverse (o, hypernym, s). To create a dataset without this property, Toutanova and Chen (2015) introduced FB15K-237, a subset of FB15K where inverse relations are removed. WN18RR was created for the same reason by Dettmers et al. (2018).

4.2 Results

The experiments results are shown in table **??**. To clarify the difference of baseline and our model,

---

[1] https://pytorch.org/
[2] https://github.com/FengXincan/dimension
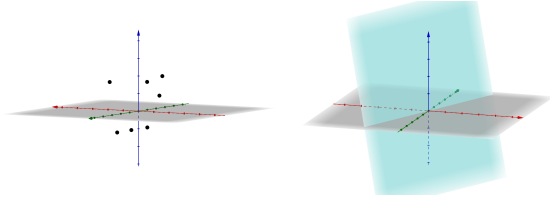
(a) Transformed entities by ComplEx and complexC. The black points in the left graph are transformed entities. In the original model, vector representations of the points are free in every dimension. While in the conjugate model, half of the vector representations of the points are linearly dependent on the other half. The linear relationship is illustrated in the right graph.

(b) Transformed entities by $5^\star$E and $5^\star$C$^-$e. The original model is shown on the left, and the negative conjugate model is shown on the right. Blue traces are the original entity and its projection in the Non-Euclidean space. Green traces are the multiple copies of the blue traces under iterations of the Möbius transformation. Red traces are the inverse of green traces.

Figure 1: Transformed entities illustrated in 3D

we calculated the training time per epoch and conducted two independent-sample t-test ($\alpha = 0.05$) on time and MRR.

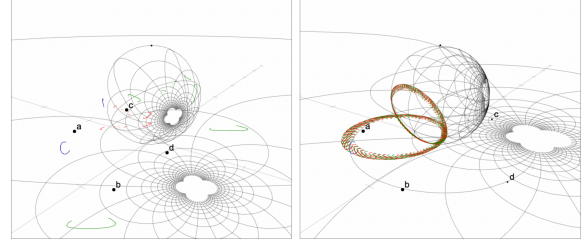Main observations: (i) The parameter size and calculation time were reduced to at most 50% in different situations. Please note that our method mainly improves the parameters' efficacy. The computation is reduced in the regularization process, and the time required is influenced by multiple factors. (ii) Although we didn't find out the best hyperparameter setting for all baseline models (e.g., $5^\star$E), We observe that in all hyperparameter settings we've experimented, the proposed conjugate models perform comparable or even little improved than origin model in big datasets. (iii) On smaller datasets, reducing parameters might cause problem in regularization process, which hurts the performance. (e.g., there is no problem in complexC, but there is in $5^\star$Ce on WN18.)

Other minor observations: (i) $5^\star$E with only half of the parameter regularization (which we call $5^\star$E$_{hreg}$) performs **alike** with $5^\star$Ce. (ii) The performance of $5^\star$Ce comparing with $5^\star$E is **basicly proportional to** $\frac{Rel \times Exa}{Ent}$. The results are comparable in datasets FB15K-237, FB15K, YAGO3-10, but dropped a bit in datasets WN18RR and WN18. (iii) Negative conjugate model $5^\star$C$^-$e performs as well as $5^\star$Ce. (iv) Conjugate model in other dimensions ($5^\star$De) does not perform as well.

We conclude that for the link prediction task, using conjugate parameters in transformations generally does not hurt model expressiveness or at least directly, especially in big datasets. While on smaller datasets, conjugation might be problematic similar to reducing the parameters to 50% in the regularization process.

5 Conclusions

By using complex number in the representation of entities and transformations, researchers have started to utilize the relational dimensions. By using complex conjugate within the transformation representations, researchers can benifit more efficiency from the exquisite mathematics. Out parameter conjugate method generally does not hurt model expressiveness, and at the same time has three advantages: generalizable across complex KGE models; simplify calculations; obtain better visualizations.

6 References

Nayyeri M.; Vahdati S.; Aykul C.; and Lehmann J. $5^\star$ Knowledge Graph Embeddings with Projective Transformations. In *AAAI*. 2021

Ji S.; Pan S.; Cambria E.; Marttinen P.; and Yu P. S. A survey on knowledge graphs: Representation, acquisition and applications. In *arXiv*:2002.00388. 2020

Trouillon T.; Welbl J.; Riedel S.; Gaussier É.; and Bouchard G. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2071-2080. 2016

Chami I.; Wolf A.; Juan D.-C.; Sala F.; Ravi S.; and Ré C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6901-6914. 2020

Vaswani A.; Shazeer N.; Parmar N.; Uszkoreit J.; Jones L.; Gomez A. N.; Kaiser L.; Polosukhin I. Attention Is All You Need. In *arXiv*:1706.03762v5. 2017

Sun Z.-Q.; Deng Z.-H.; Nie J.-Y.; and Tang

	FB15K-237					WN18RR				
Model	time	MRR	H@1	H@3	H@10	time	MRR	H@1	H@3	H@10
ComplEx	42±8	.366±4e-4	.271	.402	.558	139±21	.488±1e-3	.442	.503	.580
complexC	46±11	.363±5e-4	.268	.400	.555	146±45	.475±9e-4	.433	.488	.558
5$\star$E										
5$\star$Ce										
5$\star$C$^-$e										
5$\star$E$_{hreg}$										
5$\star$De										

	YAGO3-10					FB15K				
Model	time	MRR	H@1	H@3	H@10	time	MRR	H@1	H@3	H@10
ComplEx	370±2	.577±1e-3	.502	.622	.712	346±124	.855±1e-3	.823	.874	.910
complexC	371±2	.574±2e-3	.500	.618	.707	293±16	.855±1e-3	.827	.871	.907
5$\star$E										
5$\star$Ce										
5$\star$C$^-$e										
5$\star$E$_{hreg}$										
5$\star$De										

	WN18				
Model	time	MRR	H@1	H@3	H@10
ComplEx	57±3	.951±3e-4	.944	.954	.961
complexC	58±5	.950±3e-4	.945	.953	.960
5$\star$E					
5$\star$Ce					
5$\star$C$^-$e					
5$\star$E$_{hreg}$					
5$\star$De					

Table 1: Link prediction results on FB15K-237, WN18RR, YAGO3-10, FB15K and WN18 datasets. Time (s/epoch), MRR and H@n are presented as mean (± standard deviation) of 17 same experiments.

Dataset	Ent	Rel	Exa	$\frac{Rel}{Ent}$	$\frac{Rel \times Exa}{Ent}$
FB15K-237	14541	237	544230	0.01630	8870
WN18RR	40943	11	173670	0.00027	46
YAGO3-10	123188	37	2158080	0.00030	648
FB15K	14951	1345	966284	0.08996	86927
WN18	40943	18	282884	0.00044	124

Table 2: Datasets statistics: Ent: Entities; Rel: Relations; Exa: Examples.

time t-test (h, p) on FB15K-237							
time(h, p)	ComplEx	complexC	$5^{\star}$E	$5^{\star}$Ce	$5^{\star}$C^{-}e	$5^{\star}$E$_{hreg}$	$5^{\star}$De
ComplEx	-	(0, 2e-1)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^{\star}$E	-	-	-				
$5^{\star}$Ce	-	-	-	-			
$5^{\star}$C^{-}e	-	-	-	-	-		
$5^{\star}$E$_{hreg}$	-	-	-	-	-	-	
$5^{\star}$De	-	-	-	-	-	-	-

MRR t-test (h, p) on FB15K-237							
MRR(h, p)	ComplEx	complexC	$5^{\star}$E	$5^{\star}$Ce	$5^{\star}$C^{-}e	$5^{\star}$E$_{hreg}$	$5^{\star}$De
ComplEx	-	(1, 8e-17)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^{\star}$E	-	-	-				
$5^{\star}$Ce	-	-	-	-			
$5^{\star}$C^{-}e	-	-	-	-	-		
$5^{\star}$E$_{hreg}$	-	-	-	-	-	-	
$5^{\star}$De	-	-	-	-	-	-	-

time t-test (h, p) on WN18RR							
time(h, p)	ComplEx	complexC	$5^{\star}$E	$5^{\star}$Ce	$5^{\star}$C^{-}e	$5^{\star}$E$_{hreg}$	$5^{\star}$De
ComplEx	-	(0, 6e-1)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^{\star}$E	-	-	-				
$5^{\star}$Ce	-	-	-	-			
$5^{\star}$C^{-}e	-	-	-	-	-		
$5^{\star}$E$_{hreg}$	-	-	-	-	-	-	
$5^{\star}$De	-	-	-	-	-	-	-

MRR t-test (h, p) on WN18RR							
MRR(h, p)	ComplEx	complexC	$5^{\star}$E	$5^{\star}$Ce	$5^{\star}$C^{-}e	$5^{\star}$E$_{hreg}$	$5^{\star}$De
ComplEx	-	(1, 3e-28)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^{\star}$E	-	-	-				
$5^{\star}$Ce	-	-	-	-			
$5^{\star}$C^{-}e	-	-	-	-	-		
$5^{\star}$E$_{hreg}$	-	-	-	-	-	-	
$5^{\star}$De	-	-	-	-	-	-	-

time t-test (h, p) on YAGO3-10							
time(h, p)	ComplEx	complexC	$5^{\star}$E	$5^{\star}$Ce	$5^{\star}$C^{-}e	$5^{\star}$E$_{hreg}$	$5^{\star}$De
ComplEx	-	(0, 4e-1)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^{\star}$E	-	-	-				
$5^{\star}$Ce	-	-	-	-			
$5^{\star}$C^{-}e	-	-	-	-	-		
$5^{\star}$E$_{hreg}$	-	-	-	-	-	-	
$5^{\star}$De	-	-	-	-	-	-	-

MRR t-test (h, p) on YAGO3-10							
MRR(h, p)	ComplEx	complexC	$5^{\star}$E	$5^{\star}$Ce	$5^{\star}$C^{-}e	$5^{\star}$E$_{hreg}$	$5^{\star}$De
ComplEx	-	(1, 3e-6)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^{\star}$E	-	-	-				
$5^{\star}$Ce	-	-	-	-			
$5^{\star}$C^{-}e	-	-	-	-	-		
$5^{\star}$E$_{hreg}$	-	-	-	-	-	-	
$5^{\star}$De	-	-	-	-	-	-	-

time t-test (h, p) on FB15K							
time(h, p)	ComplEx	complexC	$5^\star$E	$5^\star$Ce	$5^\star$C$^-$e	$5^\star$E$_{hreg}$	$5^\star$De
ComplEx	-	(0, 1e-1)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^\star$E	-	-	-				
$5^\star$Ce	-	-	-	-			
$5^\star$C$^-$e	-	-	-	-	-		
$5^\star$E$_{hreg}$	-	-	-	-	-	-	
$5^\star$De	-	-	-	-	-	-	-
MRR t-test (h, p) on FB15K							
MRR(h, p)	ComplEx	complexC	$5^\star$E	$5^\star$Ce	$5^\star$C$^-$e	$5^\star$E$_{hreg}$	$5^\star$De
ComplEx	-	(0, 8e-2)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^\star$E	-	-	-				
$5^\star$Ce	-	-	-	-			
$5^\star$C$^-$e	-	-	-	-	-		
$5^\star$E$_{hreg}$	-	-	-	-	-	-	
$5^\star$De	-	-	-	-	-	-	-
time t-test (h, p) on WN18							
time(h, p)	ComplEx	complexC	$5^\star$E	$5^\star$Ce	$5^\star$C$^-$e	$5^\star$E$_{hreg}$	$5^\star$De
ComplEx	-	(0, 5e-1)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^\star$E	-	-	-				
$5^\star$Ce	-	-	-	-			
$5^\star$C$^-$e	-	-	-	-	-		
$5^\star$E$_{hreg}$	-	-	-	-	-	-	
$5^\star$De	-	-	-	-	-	-	-
MRR t-test (h, p) on WN18							
MRR(h, p)	ComplEx	complexC	$5^\star$E	$5^\star$Ce	$5^\star$C$^-$e	$5^\star$E$_{hreg}$	$5^\star$De
ComplEx	-	(0, 8e-2)	-	-	-	-	-
complexC	-	-	-	-	-	-	-
$5^\star$E	-	-	-				
$5^\star$Ce	-	-	-	-			
$5^\star$C$^-$e	-	-	-	-	-		
$5^\star$E$_{hreg}$	-	-	-	-	-	-	
$5^\star$De	-	-	-	-	-	-	-

Table 3: t-test T-test (h, p) about time and MRR on FB15K-237, WN18RR, YAGO3-10, FB15K and WN18 datasets. Results are calculated using these 17 same experiments.

J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*. 2019

Bordes A.; Usunier N.; Garcia-Duran A.; Weston J.; and Yakhnenko O. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787-2795. 2013

Ji G.; He S.; Xu L.; Liu K.; and Zhao J. 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. 2015

Lin Y.; Liu Z.; Sun M.; Liu Y.; and Zhu X. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*. 2015

Nickel M.; Tresp V.; and Kriegel H.-P. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, pages 809-816. Omnipress. 2011

Yang B.; Yih W.-t.; He X.; Gao J.; and Deng L. Embedding entities and relations for learning and inference in knowledge bases. In *Conference on Learning Representations (ICLR)*. 2015

Balazevic I.; Allen C.; and Hospedales T. Multi-relational Poincaré graph embeddings. In *Advances in Neural Information Processing Systems*, 4465-4475. 2019

Toutanova K.; and Chen D. 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. 2015

Dettmers T.; Minervini P.; Stenetorp P.; and Riedel S. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference*. 2018

Ruffinelli D.; Broscheit S.; and Gemulla R. You can teach an old dog new tricks! On training knowledge graph embeddings. In *The International Conference on Learning Representations*. 2020